Analyzing WhisperGate and BlackCat Malware: Methodology and Threat Perspective

Mathew Nicho¹, Rajesh Yadav², Digvijay Singh³

Research and Innovation Centre, Rabdan Academy Abu Dhabi, United Arab Emirates¹ Department of Computer Science and Engineering, GITAM University Visakhapatnam, India² Department of Computer Science and Engineering, BML Munjal University Gurgaon, India³

Abstract—The increasing use of powerful evasive ransomware malware in cyber warfare and targeted attacks is a persistent and growing challenge for nations, corporations, and small and medium-sized enterprises. This threat is evidenced by the emergence of the WhisperGate malware in cyber warfare, which targets organizations in Ukraine to render targeted devices inoperable, and the BlackCat malware, which targets large organizations by encrypting files. This paper outlines a practical approach to malware analysis using WhisperGate and BlackCat malware as samples. It subjects them to heuristic-based analysis techniques, including a combination of static, dynamic, hybrid, and memory analysis. Specifically, 12 tools and techniques were selected and deployed to reveal the malware's innovative stealth and evasion capabilities. This methodology shows what techniques can be applied to analyze critical malware and differentiate samples that are variations of known threats. The paper presents currently available tools and their underlying approaches to performing automated dynamic analysis on potentially malicious software. The study thus demonstrates a practical approach to carrying out malware analysis to understand cybercriminals' behavior, techniques, and tactics.

Keywords—Malware analysis; WhisperGate; BlackCat; malware sample; ransomware

I. INTRODUCTION

The geopolitical events in Ukraine at the start of 2022 were preceded by the devastating cyber warfare operation highlighted by WhisperGate malware (A malware that corrupts a system's master boot record, displays a fake ransomware note, and encrypts files based on certain file extensions). WhisperGate is considered dangerous because it can launch cyber-attacks and compromise sensitive information against hardened targets. Since it was deployed in cyber warfare against Ukraine, it could exploit unknown vulnerabilities in a target's security systems and cause significant harm. The destructive capabilities of WhisperGate make it a threat to individual, organizational, and national security. At the end of 2021, a sophisticated malware called BlackCat also known as "AlphaV," emerged, targeting U.S. organizations and their affiliates in Europe, the Philippines, and other locations. While WhisperGate masquerades as ransomware targeting nationstates (in this case, Ukraine), BlackCat has emerged as deadly ransomware targeting U.S. and European retail, construction, and transportation organizations. BlackCat appeared as an innovative ransomware-as-a-service (RaaS) group leveraging the Rust programming language and offering affiliates 80% to 90% of ransom payments [1]. Affiliates included Germany's tank storage and terminal firm Oiltanking and energy firm Mabatnaft, Belgian energy firm Sea-Invest, and Dutch oil and gas firm Evos. These attacks underlined the growing vulnerability of critical infrastructure companies to malicious hackers [2]. Both pieces of malware were challenging to defend against due to their elusive and evasive nature, which intrigued cybersecurity analysts worldwide. Since the top three cyberattacks that organizations are most concerned about are ransomware, social engineering, and malicious insider activities [3], WhisperGate and BlackCat were ideal candidates for our practical malware analysis approach due to their stealth and evasive capabilities and the destructive consequences they can cause.

Multiple malware classes, such as worms, viruses, spyware, Trojan horses, rootkits, ransomware, keyloggers, and adware, are designed with specific functionalities namely data exfiltration, data encryption, and data destruction. Despite the widespread use of antimalware software, the number of malware infections continues to grow. Malware, especially zero-day malware, can evade antimalware solutions and even infect them with its built-in defensive mechanisms. Along with WhisperGate, malware deployed against Ukraine included HermeticWiper, IsaacWiper, HermeticWizard, and CaddyWiper. Once inside the initial network, it leverages that access to compromise user and administrator accounts in the active directory of Windows' server and configures malicious group policy objects through Windows' task scheduler [4].

This paper provides a practical approach to performing malware analysis using integrated tools and techniques to assess WhisperGate and BlackCat. The Microsoft Threat Intelligence Centre disclosed that WhisperGate, categorized as a wiper, targeted several organizations in Ukraine and was tracked as DEV-0586 with a design similar to ransomware but lacking a recovery mechanism [5]. BlackCat, which belongs to a sophisticated ransomware as a service (RaaS) family, extorts money from targeted institutions instead. The structure of the remainder of this paper is as follows: Section II discusses extant literature on ransomware in general, the identified ransomware, and an evaluation of malware analysis techniques. Section III outlines the methodology, and Section IV focuses on the experimental analysis. Section V discusses the results, and Section VI concludes the research with suggestions on future research.

II. LITERATURE REVIEW

The section provides an overview of ransomware, discusses two specific types of malware (WhisperGate and BlackCat), and then evaluates malware analysis methodologies to focus on the appropriate technique(s).

A. Ransomware

Ransomware is considered one of the most threatening types of malware. Its attacks increased by 151% in 2012, averaging 270 cyberattacks per organization, with each successful breach resulting in a cost of \$3.6 million for the affected company [3]. Cyber-attacks predominantly occur through ransomware, social engineering, and malicious insider activity [3]. In particular, ransomware leverages social engineering methods to gain unauthorized access to the victim's network. Once an infection is spread, the user is extorted and asked for a monetary payment against the locked access [6], but there is no guarantee that they will regain access to their locked files after paying the ransom. Threat actors often receive the payment but still retain the data. These cybercriminals often request payment in cryptocurrency, as it is untraceable and allows them to evade responsibility [7]. Malwares leverage the Trojan by disguising themselves as legitimate software and download the malicious components, which negatively impact the system and tend to infect files and target other systems [8]. While commercial solutions are available, these are not 100% secure, because hackers use more sophisticated techniques to follow the evolution and bypass the protection techniques [9]. WhisperGate is classified as a wiper, i.e., it disguises itself as ransomware but instead aims to cause mass destruction by wiping out hard drives at targeted organizations [10].

Removing the ransomware or restoring the infected devices is ineffective, as the ransomware uses asymmetric cryptography [11], which makes it robust. The encryption makes it so that the victim is unable to access the data without first decrypting it using a key [12]. Threat actors usually ask for a ransom in exchange for the decryption key and target organizations that handle large amounts of sensitive data. The victim is faced with inaccessibility and damage to their data and often pays the ransom demand. Since most of the victims are threatened with their data and sensitive information being exposed [13].

Among the five types of ransomware—locker, crypto leakware, scareware, and pseudo-ransomware [14], WhisperGate comes under the pseudo-ransomware category, while BlackCat comes under leakware category. Also known as doxware, leakware presents a high-risk level because it is well-targeted to institutions such as banks or those that work with confidential and critical data. This ransomware does not destroy the data but threatens to release them into the public domain. Furthermore, since the context can damage the institution's image, an even greater emphasis can be placed on the quick payment of a ransom. Accordingly, BlackCat operates as a RaaS option that permits earning a percentage of the ransom payment to all the persons who have low technical knowledge about how to create ransomware but are members of this network. It is only necessary that those members spread the ransomware as far as possible while the RaaS vendor can focus on how to make this malicious software cause even more damage.

B. WhisperGate

Unlike traditional ransomware campaigns where the motive is clear, the BlackCat campaign is believed to be pseudo, with its intention being to cause the destruction of infected systems, as evidenced by the Stage 4 wiper that overwrites data on the victim's system, making decryption impossible [15]. The malware that was explicitly launched against various Ukrainian organizations in geopolitically motivated attacks was first analyzed by the Microsoft Threat Intelligence Center and detected on January 13, 2022 [16]. The BlackCat ransomware campaign targeted Ukraine in 2021 prior to its physical invasion, but it was detected and neutralized before causing any severe damage [17]. Russian cyber operations have targeted Ukraine with destabilization efforts for years through attacks on critical infrastructure, influence operations, website defacement, and attacks against banks and military networks [16]. WhisperGate, while masquerading as ransomware, corrupts a system's master boot record, displays a fake ransomware note, and then encrypts files based on specific file extensions. While a ransomware message is displayed during the attack, the targeted data is destroyed and is not recoverable even if a ransom is paid [18]. The multi-stage infection chain downloads a payload that wipes the master boot record (MBR). Then, it downloads a malicious Dynamic-link library (DLL) file hosted on a discord server (a platform where people can interact with each other in real time), which drops and executes another wiper payload that destroys files on the infected machines [19]. The malware, which is designed to look like ransomware, is intended to render the targeted devices inoperable rather than to obtain a ransom, as it does not have an inbuilt recovery code. Using social engineering methods in an Advanced Persistent Threat (APT) campaign, the attackers might have used stolen credentials and likely had access to the victim's network for months before the attack [19]. The malware can also extend to extranet networks. The recommendations from the US Cybersecurity and Infrastructure Security Agency that organizations with ties to Ukraine should carefully consider how to isolate and monitor those connections to protect themselves from potential collateral damage are echoed.

Following the detection of WhisperGate, HermeticWiper, another similar malware masquerading as ransomware used against organizations in Ukraine, was discovered on February 23, 2022. The malware targets Windows devices, manipulating the master boot record and resulting in subsequent boot failure [18]. Since both these pieces of malware (WhisperGate and HermeticWiper) are similar, the WhisperGate malware was selected as an example to study. The diamond model of intrusion analysis (DMIA) illustrates (Fig. 1) the four dimensions of the malware attack: the adversary profile, the affected infrastructure, the deployed capabilities, and the target. Specifically, the adversary deploys a capability over a specific infrastructure against a victim [20].

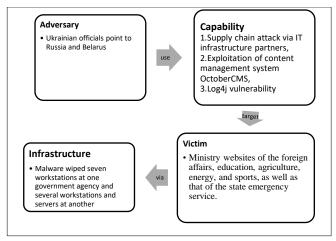


Fig. 1. DMIA model of WhisperGate malware attack.

C. BlackCat

BlackCat is a sophisticated and innovative ransomware family that surfaced in mid-November 2021. It operates as a RaaS business model, and it gained notoriety for soliciting affiliates in known cybercrime forums and offering them to leverage the ransomware and keep 80%–90% of the ransom payment [21]. BlackCat made headlines as one of the first ransomware families written in the Rust programming language, which is used to evade detection by conventional security solutions that may struggle to analyze and parse binaries written in Rust [5].

The rise of cybercrimes has been fueled by the anonymity and non-reversibility of cryptocurrencies, particularly Bitcoin, which makes ransomware payments simple for victims and risk-free for ransomware operators. The trend towards using cryptocurrencies such as Monero, which offers improved security, privacy, and anonymity, is growing, as Monero transactions cannot be traced back to a specific user or address, and the transaction history is kept private. Nonetheless, Bitcoin remains the most popular payment method for ransomware [14]. Among the 31 Ransomware listed by Unit 42, BlackCat has only the seventh largest number of victims listed on their leak site. However, while Lockbit 2.0 ransomware has a list of 50 victims over a period of six months, BlackCat has had an impressive record of 12 victims in just one month since its emergence tanner [21], which makes it a suitable candidate for analysis. In some cases, BlackCat operators use triple extortion by threatening to perform a Distributed Denial of Service (DDoS) attack on the victims' infrastructure: if the ransom is not paid, leak the information along with the data encryption [21]. The DMIA model illustrates the attack process (Fig. 2).

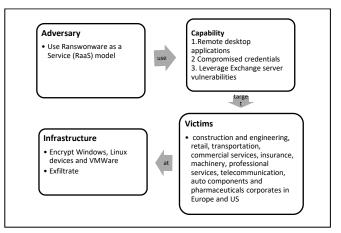


Fig. 2. DMIA model of a BlackCat malware attack.

D. Malware Analysis

Malware analysis applies program analysis and network analysis techniques to understand the behavior and evolution of malicious samples over time [22] and estimate the level of threat and harm a file can cause. Additionally, this kind of analysis helps identify a malicious file's purpose, origin, process execution, file monitoring, and hidden indicators [23].

Malware analysis from a heuristic-based detection (also known as anomaly or behavior based) consists of four types, namely, static, dynamic, hybrid and memory analysis [24]. Being heuristic-based, the proposed research focus on the static, dynamic, hybrid and memory analysis. First, static analysis was used to examine malware samples without the file's execution to extract necessary information from a suspicious file, which assisted us in classifying and identifying its execution. This information is usually gathered using static analysis tools, which examine the sample code more effectively [25]. Static analysis assists in the discovery of the binary code, which contains very useful information about the malicious behavior of a program in the form of op-code sequences, functions, and parameters. However, this method alone may not suffice for a zero-day malware (WhisperGate and BlackCat) before its discovery because new pieces of malware are created daily. The signature-based detection approach followed by the static analysis method requires frequent updates of the virus signature database, which is the method's main disadvantage [26].

Dynamic analysis is deployed, since hackers use various techniques, such as code obfuscation, dynamic code loading, encryption, and packing, to evade static analysis (including signature-based antivirus tools). Furthermore, dynamic analysis can help understand the analyzed file, thus improving detection capabilities [27]. In dynamic malware analysis, the suspicious files are executed and monitored in a controlled environment [24]. Dynamic analysis includes function call monitoring, network simulation, and registry and file changes. Interactive behavioral properties are observed and analyzed after the simulation of malware. When malware is executed in a dynamic environment, it changes its behaviors. Therefore, static features can be extracted easily and correctly. Hence, the extraction of static features in a dynamic environment detects malware efficiently. The accuracy of dynamic malware analysis alone may instead not be efficient due to the malware's intelligent behaviors [28].

While both static and dynamic analysis techniques are effective on their own, in specific situations, an integrated technique combining the relative merits of each is more efficient. Hybrid analysis that combines both static and dynamic malware analysis is thus generally preferred [29]. Memory analysis, which is used in both malware analysis and malware forensics, involves both the acquisition and analysis phase, thus providing a more comprehensive view of the malware than static and dynamic analyses and an excellent way to analyze memory by preserving a system's contents [29]. Since malware can hide its code in the computer system effectively, it must execute its code in the memory to perform its tasks [24]. Therefore, based on the evaluation of static, dynamic, hybrid, and memory analysis, and having reviewed the efficiency and effectiveness of each of these approaches, the proposed research thus focusses on all four heuristic methods.

E. Methodologies Deployed in Malware Analysis

A survey of extant research presented relevant methods used for malware analysis, namely Eureka, disassembled code analyzer for malware (DCAM), malware analysis reverse engineering (MARE), and systematic approach to malware analysis (SAMA). Eureka, a framework allowing a static analysis of malware binaries, highlights the need to produce unpacked code. It provides Windows application programming interface (API) resolution to identify the system calls in the unpacked code [30]. Or-Meir et al. conducted an overview of existing dynamic analysis methods and provided a malware classification based on each category's behavior, mapping layouts, techniques, and flow comprising memory forensics using volatile tools [27]. Almashhadani et al. used the Lock family of crypto-ransomware as their case study for their comprehensive behavioral analysis (BA) of crypto-ransomware [31]. Their work assisted us in the malware analysis of BlackCat, as the latter showed similarities with cryptoransomware. Ren et al. provided a three-level ransomware detection and prevention mechanism using virtual machines on Petya and NotPetya ransomware [32]. Similar to WhisperGate in terms of its behavior, NotPetya falls under the category of a wiper disguised as ransomware. Hence, its analysis assisted us in the analysis on WhisperGate.

DCAM is a static malware detection technique using code disassembly to recognize malware variants based on a common core signature with promising results on a set of malware [33]. MARE introduced a four-stage approach covering a structured analysis process that focuses on producing an objective outcome to detect malware followed by isolation and extraction phases, as shown by [34], who introduced the malware behavioral technique, malware reverse engineering, and code analysis. The author in [35] proposed an automated analysis framework to analyze executable behaviors through a synergic combination of malware detection techniques, including using a virtual machine over a sandbox to enhance invisibility. SAMA provides detailed information on the working of malware, and its applicability over any type of malware makes it robust. It follows a four-stage approach, namely, an modified version of MARE, as shown by [36]. The authors pointed to the execution order provided by MARE and noted that code analysis must be executed along with behavioral analysis.

SAMA is a complete methodology for performing malware analysis, and malware analysts have used it to analyze the following malware threats: Stuxnet, Dark Comet, Poison Ivy, Locky, Careto, and Sofacy Carberp, including Flame and Red October, as shown by [36]. However, the authors did not explain how they used their tools for each step. Furthermore, through the use of SAMA, the authors have only partially discussed memory analysis. Additionally, the stage of packaging obfuscation is executed after the initial five steps of classification in SAMA. However, it would be more impactful to include obfuscation checking before any other step because the analysis could lead to incorrect results. Finally, a hybrid technique should be included as part of the methodology to perform an in-depth analysis, which is missing in SAMA. Accordingly, hybrid analysis was performed to obtain relevant information and fast results to assess WhisperGate and BlackCat malware. The proposed approach is illustrated against those presented in the literature in Table I. Specifically, the lab setup process and static, dynamic, code, and memory analysis is compared.

 TABLE I.
 Comparative Analysis of Analysis Methodologies

Research Methods	Lab Set-up	Static	Dynamic	Hybrid	Memory
DCAM	No	Yes	No	No	No
MARE	No	No	Partial	No	No
Vidyarthi et al.	Partial	Yes	Yes	No	No
SAMA	Yes	Yes	Yes	No	Partial
Proposed approach	Yes	Yes	Yes	Yes	Yes

III. METHODOLOGY

Since the main objective of the paper is to illustrate a practical approach to carrying out malware analysis, the two sample pieces of malware were analyzed using the following integrated tools and the four analyses. The implemented methodology is shown in Fig. 3 and can be used as a guideline for future comprehensive malware analyses.

A. Lab Setup

Flare VM is an open-source collection of software installation scripts for Windows systems to easily setup and maintains a reverse engineering environment on a virtual machine. It was installed on Virtual Box hypervisor to analyze the encrypted malicious file downloaded from Malware Bazaar—a project from abuse.ch). Then, it was downloaded and installed on Windows 10 VM. A system snapshot was taken before each analysis to preserve the integrity of the

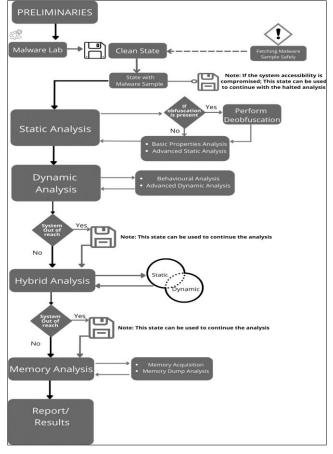
results and for future-reference analysis. The following sub sections discuss the rationale for choosing the four malware analysis techniques.

B. Applying Static Analysis

Static analysis examines the malicious sample by gathering maximum information without executing it by following a three-step phases approach: de-obfuscation, basic properties analysis (BPA), and advanced static analysis (ASA) (Fig. 4).

The cyber attacker uses obfuscation to intentionally disguise some attributes of the malware specimen by packing it. Hence, before proceeding with the analysis, the initial step should be to perform an obfuscation check and bring the sample to its unpacked state (BPA) In this respect, the file type and signature identification are crucial steps of static analysis to obtain useful information, such as the target operating system (OS) and the architecture of the suspicious file. Among Windows' basic executable files, the presence of portable executables in the form of .exe or .dll provides a glimpse of hexadecimal values and notes that are present in a file.

Malware hashing involves the generation of cryptographic hashes for a malicious file. Hashing algorithms are commonly used to generate hash values of the malicious files are MD5, SHA-256, and SHA-1. This step provides unique values that act as fingerprints for the malware samples. VirusTotal website allows for the flexibility to either upload a file or a URL or simply search the hash value of the sample, and it offers APIbased support for detection and recognition by supplying the details of the previous records created by other researchers. String analysis extracts legible letters and words from the malware and focuses on critical information that can be fetched from strings, such as file names, IP addresses, registry keys, and URLs. However, an attacker may include fake strings to divert an analyst from disrupting their task, as strings provide an overview of what malware can do.



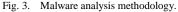




Fig. 4. The three-step phase of the static analysis.

The goal of advanced static analysis of the static code is to understand the malware code's design. This kind of analysis includes the analysis of the machine code by disassembling a file. Performing the static code analysis after BA is appropriate because it requires analyzing the processes and behavior of malware by comparing the two states, namely pre- and postexecution. Conversely, if static code analysis is performed before behavioral analysis, it might reduce the accuracy.

C. Applying Dynamic Analysis

Dynamic analysis implies the execution of the malware sample in a contained and safe environment (sandboxed) to understand the malware's functionality, which includes changes in registry and the files created by it. The objective is to cover two important phases of dynamic analysis, namely behavioral analysis (BA) and advanced dynamic analysis (ADA) (Fig. 5).



Fig. 5. The two step phase of dynamic malware analysis.

The objective of BA is to understand the suspicious behavior of malware through the interaction with the sample to gather maximum information. BA helps understand the changes in registry, network, and files. The methods in this step include domain name system (DNS) handling, snapshot comparison, process analysis, and registry analysis. DNS handling involves setting up a fake server to generate responses for the requests created by the suspicious file. When the malware is executed, it creates a DNS request so that it can perform the required malicious behavior. These requests are resolved by creating a fake server that fools the malware and generates the response. The snapshot comparison of Window's registry keys focuses on obtaining information about the changes in the number of registry keys and their values before and after the execution of the suspicious file. A registry key is an organizational unit that serves the purpose of an internal database and is used by the computer to store information related to configuration. In the process analysis stage, the malicious application is executed to elicit information regarding its behavior, namely the activities of the application and the details of threads, memory, handles, and the child processes created by it. Analysis of real-time registry, file system, and thread activity of the malicious file involves advanced monitoring of the applications using thread stacks, sessions created, and their activities. It also helps obtain information on the path the processes have traversed in the system, including the changes made. The objective of the ADA stage is to perform advanced analysis on the code by debugging the dynamic code by executing a file.

D. Applying Hybrid Analysis

Since hybrid malware analysis assist to obtain the benefits of both static and dynamic malware analysis [24], this increases their ability to detect malicious software correctly. Furthermore, this analysis technique has all the strengths of static and hybrid analyses while overcoming the shortcoming they have when they are performed independently.

E. Applying Memory Analysis

The main objective of performing this stage is to gain information by monitoring memory changes. An analyst examines the memory dump to gain additional information on process execution and performs a restoration step to make a clean state for further analysis. Memory analysis is integrated with hybrid analysis when an analyst applies basic static analysis to the information gathered during interactive BA, namely the execution of malicious code to generate memory changes followed by dynamic analysis. This phase will be reverted to perform basic static analysis on that memory dump. The table below lists the tools used in malware analysis.

TABLE II. TOOLS AND TECHNIQUES USED IN MALWARE ANALYSIS

No.	Tools	Static	Dynamic	Hybrid	Memory
1	ExinfoPE	Y	Х	Х	Х
2	Hex Editor	Y	Х	Х	Х
3	PeStudio	Y	Х	Х	Х
4	Virustotal	Y	Х	Х	Х
5	Ghidra	Y	Х	Y	Х
6	ApateDNS	Х	Y	Х	Х
7	Regshot	Х	Y	Х	Х
8	Process Monitor	Х	Y	Х	Х
9	IDA	Х	Х	Y	Х
10	Procmon	Х	Х	Y	Х
11	AccessData FDK	Х	Х	Х	Y
12	Volatility	Х	Х	Х	Y

IV. RESULTS AND ANALYSIS

This section presents a practical approach to analyzing malware using open-source and powerful tools (Table II). The WhisperGate malware is analyzed first, followed by the analysis of BlackCat malware. The preliminary lab setup was meticulously followed to ensure the status of the malware before and after each process (i.e., static, dynamic, hybrid and memory analysis). The following subsections highlight the experimental results obtained through the judicious use of appropriate tools.

A. Experimental Findings - WhisperGate

1) Static analysis: The static analysis followed a three-step approach, namely deobfuscation, BPA, and ASA. Exeinfo PE is a software that can be used to view executable file properties. When using the tool Exeinfo PE to deobfuscate and identify whether the malware was obfuscated (Fig. 6), it was found that the file was unpacked.



Fig. 6. Deobfuscation of WhisperGate using Exinfo PE.

In the File and Signature Identification step of BPA, the tools Hex Editor (Hxd) were used to obtain detailed information on the signature (Fig. 7 and 8), and PeStudio was used to identify the file type of the malware (Fig. 9). HxD is a tool that can inspect, compare, and verify files, disks, disk images, memory, and log files; patch errors, and repair disk structures. Used for malware detection, PeStudio analyzes the executable files and provides information about the file's properties, characteristics, and potential risks.

W HxD - [C:\U									1			ige1.	exe]				
📄 rite Edit .				15	· .			-		ows		i)		\sim	hex		\checkmark
🕼 Stage1.exe																	
Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	OF	Decoded text
00000000	4D	5A	90	00	03	00	00	00	04	00	00	00	FF	FF	00	00	MZÿÿ.
00000010	B8	00	00	00	00	00	00	00	40	00	00	00	00	00	00	00	,
00000020	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0000030	00	00	00	00	00	00	00	00	00	00	00	00	80	00	00	00	€
00000040	0E	1F	BA	0E	00	В4	09	CD	21	B8	01	4C	CD	21	54	68	º'.Í!,.LÍ!1
00000050	69	73	20	70	72	6F	67	72	61	6D	20	63	61	6E	6E	6F	is program can
00000060	74	20	62	65	20	72	75	6E	20	69	6E	20	44	4F	53	20	t be run in DOS
00000070	6D	6F	64	65	2E	0D	0D	0A	24	00	00	00	00	00	00	00	mode\$
00000080	50	45	00	00	4C	01	08	00	5F	00	DC:	61	00	00	00	00	PE. T. ^ Üa

Fig. 7. File and signature identification of WhisperGate using Hxd.

The figure shows that the first two bytes contain 4D 5A, and the decoded text is MZ (which stands for Mark Zbikowski, a leading developer of DOS). Both these values are a crucial factor, which tells us that it is a portable executable. Another file signature that can be observed from the tool is the note that tells that "This program cannot be run in DOS mode." The decoded text states that "Your hard drive has been corrupted... In case you want to recover all hard drives of your organization, You, should pay us \$10k Dollars via bitcoin wallet ** and send message via tox ID ** with your organization name. We will contact you to give further instructions" (Fig. 8).

₩ HxD - [C:\U						1						ige1.	exe]				- L
🗋 🚵 🕶 🕅				-	· .		•				r (ANS	il)		\sim	hex		\sim
Stage1.exe																	
Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0В	oc	OD	0E	OF	Decoded text
00003270	00	00	66	C7	06	7E	7C	00	00	00	00	EB	C4	66	81	06	fÇ.~ ëÄf
00003280	7A	7C	C7	00	00	00	66	81	16	7E	7C	00	00	00	00	F8	z Çf~
00003290	EB	AF	10	00	01	00	00	00	00	00	01	00	00	00	00	00	ē
000032A0	00	00	41	41	41	41	41	00	59	6F	75	72	20	68	61	72	AAAAA.Your ha
000032B0	64	20	64	72	69	76	65	20	68	61	73	20	62	65	65	6E	d drive has bee
000032C0	20	63	6F	72	72	75	70	74	65	64	2E	0D	AO	49	6E	20	corruptedIn
000032D0	63	61	73	65	20	79	6F	75	20	77	61	6E	74	20	74	6F	case you want t
000032E0	20	72	65	63	6F	76	65	72	20	61	6C	6C	20	68	61	72	recover all ha
000032F0	64	20	64	72	69	76	65	73	0D	0A	6F	66	20	79	6F	75	d drivesof yo
00003300	72	20	6F	72	67	61	6E	69	7A	61	74	69	6F	6E	2C	0D	r organization
00003310	0A	59	6F	75	20	73	68	6F	75	6C	64	20	70	61	79	20	.You should pay
00003320	75	73	20	20	24	31	30	6B	20	76	69	61	20	62	69	74	us \$10k via b
00003330	63	6F	69	6E	20	77	61	6C	6C	65	74	0D	0A	31	41	56	coin wallet12
00003340	4E	4D	36	38	67	6A	36	50	47	50	46	63	4A	75	66	74	NM68gj6PGPFcJu
00003350	4B	41	54	61	34	57	4C	6E	7A	67	38	66	70	66	76	20	KATa4WLnzg8fpf
00003360	61	6E	64	20	73	65	6E	64	20	6D	65	73	73	61	67	65	and send messa
00003370	20	76	69	61	0D	0A	74	6F	78	20	49	44	20	38	42	45	viatox ID 8
00003380	44	43	34	31	31	30	31	32	41	33	33	42	41	33	34	46	DC411012A33BA34
00003390	34	39	31	33	30	44	30	46	31	38	36	39	39	33	43	36	49130D0F1869930
000033A0	41	33	32	44	41	44	38	39	37	36	46	36	41	35	44	38	A32DAD8976F6A51
000033B0	32	43	31	45	44	32	33	30	35	34	43	30	35	37	45	43	2C1ED23054C057
000033C0	45	44	35	34	39	36	46	36	35	OD	AO	77	69	74	68	20	ED5496F65wit)
000033D0	79	6F	75	72	20	6F	72	67	61	6E	69	7A	61	74	69	6F	your organizat:
000033E0	6E	20	6E	61	6D	65	2E	OD	A0	57	65	20	77	69	6C	6C	n nameWe wil
000033F0	20	63	6F	6E	74	61	63	74	20	79	6F	75	20	74	6F	20	contact you to
00003400	67	69	76	65	20	66	75	72	74	68	65	72	20	69	6E	73	give further in
00003410	74	72	75	63	74	69	6F	6E	73	2E	00	00	00	00	55	AA	tructions
00003420	EB	00	8C	C8	8E	D8	BE	88	7C	E8	00	00	50	FC	8A	04	ë.ŒÈŽØ%^∣èPü
00003430	3C	00	74	06	E8	05	00	46	EB	F4	EB	05	B4	0E	CD	10	<.t.èFëôë.'.
00003440	C3	8C	C8	8E	D8	A3	78	7C	66	C7	06	76	7C	82	7C	00	ÌȎأx fÇ.v ,
00003450	00	B4	43	BO	00	8A	16	87	7C	80	C2	80	BE	72	7C	CD	′C°.Š.‡ €Â€%r
00003460	13	72	02	73	18	FE	06	87	70		C7		7A	7C	01	00	.r.s.p.‡ fÇ.z
00003470	00	00	66	C7	06	7E	7C	00	00	00	00	EB	C4	66	81	06	fC.~ ëÄf
00003480	7A	7C	C7	00	00	00	66	81	16	7E	7C	00	00	00	00	F8	z Cf~
00003490	EB	AF	10	00	01	00	00	00	00	00	01	00	00	00	00	00	ë ⁻
000034A0	00	0.0	41	41	41	41	41	0.0						68	61	72	

Fig. 8. Decoded text message of WhisperGate using Hxd.

After the signature identification, PeStudio was used to identify the correct type of file (Fig. 9). Based on the results, the file is built on a 32-bit CPU architecture with a file size of 27648 bytes.

1日×日 8		
ClusersVieuser\desktop\whispergate\whisp desktop\whispergate\whisp desktop\whispergate\whisp desktop\whispergate\whisp desktop\whispergate\whisp	md5	value 505C99A08A7D927346CA2DAFA7973FC1
for shared (16 bytes) if do shared (16 bytes)	sha1 sha250 md5-without-overlay sha1-without-overlay sha2-without-overlay first-bytes-ben first-bytes-ben site-without-overlay entropy	1019160002C12C2404A050023005000000010010101 102600000000000000000000000000000000000
	certificate-stamp	n/a

Fig. 9. File type identification of WhisperGate using PeStudio.

Upon hashing, PeStudio generated the hash values:

MD5 : 5D5C99A08A7D927346CA2DAFA7973FC1

SHA-1:

189166D382C73C242BA45889D57980548D4BA37E

SHA-256:

A196C6B8FFCB97FFB276D04F354696E2391311DB3841AE 16C8C9F56F36A38E92

The API-based identification stage involves using the webbased tool VirusTotal. VirusTotal is an online portal where users can upload suspicious files. It uses antivirus engines and website scanners to detect various types of malware and malicious content. The identification and classification of the malware show a community score of 54 on a scale of 69, signifying malware detection by 54 security vendors out of 69. Details such as the history of the application, the compilation stamp, and the information of the target identified through the API are shown below in Fig. 10.

History ①	
Creation Time	2022-01-10 10:37:18 UTC
First Submission	2022-01-16 20:30:19 UTC
Last Submission	2022-01-26 12:15:01 UTC
Last Analysis	2022-03-30 04:31:25 UTC
Names (i)	
a196c6b8ffcb97ffb2	276d04f354696e2391311db3841ae16c8c9f56f36a38e92.exe
172502_99@172502	
m82hr78cg.dll	
unknown	
anknown	
Portable Executab	le Info ①
Header	
Target Machine	Intel 386 or later processors and compatible processors
Compilation Timest	amp 2022-01-10 10:37:18 UTC
Entry Point	4864
Contained Sections	8

Fig. 10. API identification of WhisperGate from VirusTotal.

In the last step of BPA, namely string analysis, PeStudio was used to analyze the strings and retrieve useful information, as shown in Fig. 11.

oncoding (2)	size(bytes)	file offset	blockist (4)	hint (23)	group (7)	volue (161)
ALCI	4	0.0310103				· · ·
ALCI	384	0.03300284				Your hard drive has been complexity/pin case you want to recover all hard drively(not you
ALCI	384	0.0330484				Your hard drive has been complexity/pin case you want to recover all hard driver/pinot you
A10.0	384	0.03300644				Your hard drive has been complexity/pin case you want to recover all hard drively (not you
810.0	354	0.0330354				Your hard drive has been complexity on care you want to recover all hard driverly not you
4104	384	0.03300.484				Your hard drive has been complexity/win case you want to recover all hard driver/y/not you
810.0	354	CONTRACTO				Your hard drive has been corrupted. (r) of n care you want to recover all hard drived yipot yo.
101	384	0.0330194				Your hard drive has been complexity(win care you want to recover all hard driver) ((not you
A101	384	0.03304044				Your hard drive has been complexity/win case you want to recover all hard driver/y/not you
100	354	0503334264				Your hard drive has been complexity on case you want to recover all hard driver young you
100	384	0.03304484				Your hard drive has been complexity(yon care you want to recover all hard driver) (yout you
810.0	354	0.03334644				Your hard drive has been complexit/yinin case you want to recover all hard driver/yinot you
4104	384	0.03304384				Your hard drive has been complexity(win care you want to recover all hard driver) (your you
A12.1	384	CHORDWARA				Your hard drive has been complexit/you'n case you want to recover all hard driver/yout you
MCI.	354	0.03304084				Your hard drive has been complexity/win case you want to recover all hard driver/yipot you
100	384	0.03304194				Your hard drive has been complexity(usin case you want to recover all hard driver) (yout you
100	354	0.03355094				Your hard drive has been complexity/join case you want to recover all hard driver/yipot yo
A171	4					Y*1

Fig. 11. String identification of WhisperGate using PeStudio.

A total of 161 strings were identified, four of which were blacklisted, and 16 carried the note "Your hard drive is corrupted," indicating that malicious activity could be carried out using the sample. Advanced static analysis involves the single task of manually reversing the static code. The tool Ghidra (a free and open source reverse engineering tool) was used to disassemble the code and further examine the functions, which provided relevant information regarding the nature of the file, as shown in Fig. 12.

*(undefined4 *)(<pre>sparam_1 + iVarl) = 0;</pre>
*(undefined4 *)(<pre>stack0x0000000 + iVar1) = 0;</pre>
*(undefined4 *)(<pre>(int) &p_Stack4 + iVarl) = 3;</pre>
* (undefined4 *) (<pre>sstackOxfffffff8 + iVarl) = 0;</pre>
* (undefined4 *) (<pre>stackOxfffffff4 + iVarl) = 3;</pre>
* (undefined4 *) (sstackOxfffffff0 + iVarl) = Ox10000000;
* (wchar_t **) (as	<pre>tack0xffffffec + iVarl) = L"\\\\.\\PhysicalDrive0";</pre>
*(undefined4 *)((int)&uStackY24 + iVarl) = 0x403bcf;
pvVar3 = CreateF	<pre>ileW(*(LPCWSTR *)(sstackOxfffffffc + iVarl),*(DWORD *)(sstackOxfffffff + iVarl) *(DWORD *)(sstackOxffffffff + iVarl),</pre>
	<pre>*(LPSECURITY ATTRIBUTES *)(sstack0xfffffff8 + iVarl),</pre>
	*(DWORD *)((int)sp Stack4 + iVarl),*(DWORD *)(sstack0x00000000 + iVarl),
	*(HANDLE *)(sparam_1 + iVarl));
* (HANDLE *) (asta	<pre>ck0xffffffec + iVarl) = pvVar3;</pre>
* (undefined4 *) (<pre>(int)sp_Stack4 + iVarl) = 0;</pre>
*(undefined4 *)(<pre>stackOxfffffff8 + iVarl) = 0;</pre>
*(undefined4 *)(<pre>sstackOxfffffff4 + iVarl) = 0x200;</pre>
*(undefined4 **)	(sstackOmffffffff + iVarl) = local_2020;
*(undefined4 *)((int)suStackY24 + iVarl) = 0x403bfe;
WriteFile(*(HAND)	<pre>LE *) (&stack0xffffffec + iVarl), * (LPCVOID *) (&stack0xffffffff + iVarl),</pre>
* (DWOR)	<pre>D *)(sstack0xffffffff + iVarl),*(LPDWORD *)(sstack0xffffffff + iVarl),</pre>
* (LPOVI	ERLAPPED *) ((int) sp_Stack4 + iVarl));
* (HANDLE *) (ssta	skOxffffffec + iVarl) = pvVar3;
*(undefined4 *)((int)suStackY24 + iVarl) = 0x403c09;
BVar4 = CloseHand	<pre>dle(*(HANDLE *) (&stackOxffffffec + iVarl));</pre>
* (BOOL *) (sstack	<pre>9xffffffec + iVarl) = BVar4;</pre>
return 0;	

Fig. 12. Code reverse of WhisperGate using ghidra.

The images taken from Microsoft documentation (Fig. 13 and 14) shows that the functions, CreateFileA (which opens a physical disk drive or a volume) and WriteFile (which writes data to the specified file or input/output (I/O) device) fall under the category of Data Access and Storage. Hence, it is possible to relate the general syntax with the disassembled code provided by Ghidra (Fig. 12), which contains the same two functions as the one from Microsoft documentation.

thream, directory, physical disk, volume, console buffer, tape drive, communications resource, m and pipe. The function returns a handle that can be used to access the file or device for various is of I/O depending on the file or device and the flags and attributes specified. To perform this operation as a transacted operation, which results in a handle that can be used for ansacted I/O, use the CreateFileTransacted function. Syntax C+- The transacted specified, [in] DAORD dubesinedAccess, [in] DAORD dubesinedAccess, [in] DAORD dubesinedAccess, [in] DAORD ducreation specified, [in, optional] LPSEURITY_ATTRIBUTES 1pSecurityAttributes, [in] DAORD ducreation specified, [in] DAORD ducreation specified specified, [in] DAORD ducreation specified specified, [in] DAORD ducreation specified specified, [in] DAORD ducreation specified specified specified, [in] DAORD ducreation specified spe	on (fileapi.h)	lea lunct	eater
transacted I/O, use the CreateFileTransacted function. Syntax C+* HWADLE CreateFileA([in] LPCSTR lpFileName, [in] DMORD dwDesiredAccess, [in] DMORD dwDesiredAccess, [in] DMORD dwCreationDisposition, [in] DMORD dwCreationDisposition,	2	29 minutes to read	• 02/16/2022 •
Syntax C++ PMADLE createFileA([in] LPCSTR lpFileName, [in] DMORD dwDesiredAccess, [in] DMORD dwShareMode, [in, optional] LPSECURITY_ATTRIBUTES lpSecurityAttributes, [in] DMORD dwCreatioDisposition,	e buffer, tape drive, communications resource, mails in be used to access the file or device for various typ	nysical disk, volume, cons on returns a handle that	m, directory, ph pipe. The function
transacted I/O, use the CreateFileTransacted function. Syntax C+* HWADLE CreateFileA([in] LPCSTR lpFileName, [in] DMORD dwDesiredAccess, [in] DMORD dwDesiredAccess, [in] DMORD dwCreationDisposition, [in] DMORD dwCreationDisposition,	ation, which results in a handle that can be used for	ration as a transacted op	erform this oper
Syntax C++ PANDLE createFileA([in] LPCSTR lpFileName, [in] DMORD dubesiredAccess, [in] DMORD dushareMode, [in, optional] LPSECURITY_ATTREBUTES lpSecurityAttributes, [in] DMORD ducreationDisposition,	nction	he CreateFileTransacted	acted I/O_use t
HANDLE CreateFileA([in] LPCSTR lpFileName, [in] DMORD dwDeSiredAccess, [in] DMORD dwShareNode, [in] DESCURITY_ATTRIBUTES lpSecurityAttributes, [in] DMORD dwCreationDSposition,			
[in] LPCSTR lpFileHame, [in] DMORD dwDesiredAccess, [in] DMORD dwShareHode, [in, optiona] LPSECURITY_ATTRIBUTES lpSecurityAttributes, [in] DMORD dwCreationDisposition,			
[in] LPGSTR lpfileHame, [in] DMORD dwDesiredAccess, [in] DMORD dwShareHode, [in, optional] LPSECURITY_ATTRIBUTES lpSecurityAttributes, [in] DMORD dwCreationDisposition,	₽ cc		
[in] DAORD dwShareMode, [in, optional] LPSECURITY_ATTRIBUTES lpSecurityAttributes, [in] DAORD dwCreationBispsition,	īb ce	рД.(•
[in, optional] LPSECURITY_ATTRIBUTES lpSecurityAttributes, [in] DWORD dwCreationDisposition,		7/2020/2020	• NDLE CreateFile
[in] DWORD dwCreationDisposition,	pFileName,	LPCSTR	• NDLE CreateFile [in]
	pFileName, ADesiredAccess,	LPCSTR DWORD	* NDLE CreateFile [in] [in]
[in] DWORD dwFlagsAndAttributes.	pFileName, wDesiredAccess, wShareMode,	LPCSTR DWORD DWORD	* NDLE CreateFile [in] [in]
	pFileName, uDesiredAccess, uShareMode, SecurityAttributes,	LPCSTR DWORD DWORD LPSECURITY_ATTRIBUTES	* NDLE CreateFild [in] [in] [in, optional]
[in, optional] HANDLE hTemplateFile);	pFileName, MDesiredAccess, skhareNdoLe, ssecurityAttributes, arreationDisposition, #lagsandattributes,	LPCSTR DWORD DWORD LPSECURITY_ATTRIBUTES DWORD DWORD	• WDLE CreateFile [in] [in] [in, optional] [in] [in]

Fig. 13. The CreateFileA function that relates to the result in Fig. 12.

Article • 02/16/2022 • 10 mir	nutes to read		23 4
Writes data to the specifie	ed file or input	/output (I/O) device.	
his function is designed	for both synch	ronous and asynchronous operation. Fo	r a similar function
lesigned solely for asynch	hronous opera	tion, see WriteFileEx.	
Syntax			
Syntax			
Syntax			Па Сору
,			🔁 Сору
C++	HANDLE	hFile,	ြ Сору
C++ BOOL WriteFile(HANDLE	hFile, 198uffer,	12 Сору
C++ BOOL WriteFile([in]			ြ Сору
C++ BOOL WriteFile([in] [in]	LPCVOID	lpBuffer,	В Сору

Fig. 14. The WriteFile function that relates to the result in Fig. 12.

From the disassembled code, useful information was extracted from the two functions. Their synchronization hints that a file is being opened and overwritten to execute a task. CreateFile accepts the parameter "Physical Drive," which is the name of the file being opened. The access mask used is "0xffffff0." WriteFile provides important details through the handle buffer. The handle returned by CreateFile is used by this function, while the buffer pvVar3 presents the variable Local 2020.

2) Dynamic analysis: The dynamic analysis followed two phases, namely BA and ADA. In BA, the malware was executed to interact with the sample to determine its behavior and intended purpose (see Fig. 5 for the four steps). In the DNS processing step, the ApateDNS tool (that aid analysts in DNS identification) was used to spoof DNS responses to DNA requests generated by the malware, as shown in Fig. 15.

Capture Win	dow DNS Hex View		
Time	Domain Requested	DNS Returner	1
01:00:36	crl3.digicert.com	FOUND	
01:00:37	cr14 digicert.com	FOUND	
01:00:38	client wns windows.com	FOUND	
01:00:38	ocsp.digicert.com	FOUND	
01:00:39	cr13.digicert.com	FOUND	
01:00:40	crl4.digicert.com	FOUND	
01:00:57	157.33.169.83.in-addr.arpa	FOUND	
01:01:24	235.113.112.212.in-addr.arpa	FOUND	
01:01:25	255.1.168.192 in-addr.arpa	FOUND	
01:01:34	1.1.168.192.in-addr.arpa	FOUND	
+] Sendin	t to 127.0.0.1 on Intel(R) PB y valid Disk response of first started at 00:55:53 successf	t request.	
	ply IP (Default: Current Gatway/DNS	5: 127.0.0.1	
	XMAIN's	5): 127.0.0.1 Start :	Server

Fig. 15. DNS spoofing of WhisperGate using the tool ApateDNS

During the execution of the sample, the tool successfully captured a list of DNS requests along with the timestamp, indicating that the malware was attempting to connect to different IP addresses for malicious purposes. In the snapshot comparison step, the Regshot tool was used to take sequential snapshots for the pre- and post-execution states to monitor the changes to the registry and files. RegShot that is a tool for controlling changes in the Windows registry can compare the state of registry entries "before" and "after" system changes. The pre- and post-execution snapshots are shown in Fig. 16. The images indicate information regarding the keys, values, and related attributes in the snapshot.

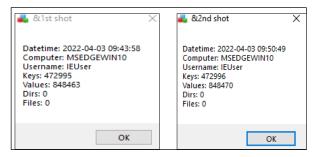


Fig. 16. The pre- and post-execution snapshots using regshot.

The Process Monitor tool was used in the Process Analysis step. Upon invoking the process name filter, the tool generated a list of sub-processes (Fig. 17). The Process Monitor is a troubleshooting and malware hunting monitoring tool for Windows that shows real-time file system, Registry and process/thread activity.

Proc	ess Monitor - Sysi	internals: www.sysim	femals.com		- 0	0.0
ie D	dt Event Filte	r Teels Option	s Hele			
		the later and of				
⇒ 🗄			B & ₽ ₽ 7 📑 =			
ne_	Process Name	PID Operation	Path	Result	Detail	
	Stage1.exe	6534 PegOper			Desired Access: Read	
	Stage1.eve	6534 RegOper			Desired Access. Read	
	Stage1.exe		Key HKUN Systen/CurentControlSet/Set		index 0. Name: Wolfither Instance	
474.	Stage 1 are	8524 PegOper	Key HKLH System Current ControlSet Se	mt. SUCCESS	Dested Access: Read	
47.4	Stage1.ext		Value HRLM\System/CurrentControlSet\Se		Type REG_DWORD. Length 4, Data: 0	
£74_	Stage1.exe	6534 🔤 RegQuer	Value HRLM-System/CurrentControlSet/Set	rvi., SUCCESS	Type: RBG_SZ, Length: 14, Data: 328010	
	Stage 1 ese		Key HILM System CurrentControlSet Set			
474	Stage1 ere	6524 RegEnue	Key HRLM System Current ControlSet (Se	IN NO MORE ENTRI.	. Index 1. Langth 80	
	 Stage1 exe 	6524 RepClose				
	Slage1 ese		Key HKLM-System/CurrentControlSet/Set		Desired Access: Read	
	Stage1 and	6524 RepOper			Destred Access: Read	
474	 Stage1 exe 		Key HKLM System CurrentControlSet Set		Index: 0. Name: Process Montor 24 Instance	
C474_	Stage1 eve	6024 MegOper	Key HKLM System CurrentControlSet Se Value HKLM System CurrentControlSet Se	FW SUCCESS	Desired Access: Read Type: REG_DWDRD_Length 4 Data 0	
	 Stage Leve Stage Leve 		Value HKLM System Current ControlSet Se Value HKLM System Current ControlSet Se		Tose REG. SZ. Length 14, Date: 385200	
24/4_	 Stage Leve Stage Leve 	6524 MepGue	Water HRLM System CurrentControlSet Se Key HRLM System CurrentControlSet Se	PH. SUCCESS	Type: Http://www.http://www.storum.com/	
	 Stage1 eve 	6524 PeoEnus			Index 3 County 30	
2424	Stage1 ese	6524 RepOrte	Key HRLM Soten CorertControlSet Se	SUPPER	nes congras	
47.4	a Stage Lease	6524 CreateFil	 Onvice Vierbile 2: DRI 	5UCCES5	Desired Access Al Access, Disorders Gen Option, Systemate IO Service Net, Net-Desity File, Metades 114, SharWold, Read Wite, Netador/Ser 114	. Ope
47.4	Slage1 eve	0234 WAteFile		SUCCESS	Offert & Length S12, LO Ragi: Non-cached, Priority Normal	
	Stage1 exe	6524 In Cosellie		SUCCESS		
6474	Stage1.ess	6504 c? Thread E		SUCCESS	Tread ID: 6232, User Time: 0.0156250, Kennel Time: 0.0000000	
	Stage1.exe	6524 cf Tread E		SUCCESS	Thread ID: 6216. User Time: 0.0000000. Kernel Time: 0.0000000	
2474	 Stage1 eve 	6534 dProcess		SUCCESS	Ext Status: 0. User Time: 0.0156250 seconds. Kernel Time: 0.000000 seconds. Physite Bytes: 684.032, Peak Physite Bytes: 684.032, Working Set: 3.100.872, Peak	ak W
	Stage1 ese		Key HKLN-System CurrentControlSet Se		Deated Access: All Access	
2474.	 Stage1.ese 		Value HRLM System Current Cantrol Set Set		D Langth: 40	
	 Stage1 eve 	6824 RepClose				
	Stage1 eve	6524 in Coselle		SUCCESS		
	Stage 1 eve	6524 IN CoseFile 6524 IN PeoDose				
	Stage1 exe	tous PegCke	Key HRLM-System CurrentControlSet Co Key HRLM-SOFTIKARE-Microsoft-Wind	W. MALESS		
	Stage Leve	soure PegCale	Hey HILM Softmarke Mooself Wild	EW. DALESS		
Cece.	 Stage1.exe 	tota Piebros	May HADR Gates Constantion of	m. 500.655		
	38 of 258 578 mm	and an array of	Racked by vistual memory			

Fig. 17. Process monitoring of WhisperGate using process monitor.

The tool successfully displayed 258,678 events triggered by the malware sample. The main processes highlighted are, again, CreateFile and WriteFile (Fig. 17), and they provide useful information regarding the execution of malware. In particular, CreateFile offers the desired access for the malware to open a file, and WriteFile allows it to overwrite.

In the Registry and File Analysis step, the output generated by Regshot was used to compare the changes made in the registry values and the modifications in the system files, as shown in Fig. 18.

📥 C&ompare	×
Keys deleted: 11 Keys added: 12 Values added: 21 Values added: 28 Values modified: 35 Folders added: 0 Folders added: 0 Files deleted: 0 Files added: 0 Files [attributes?] modified: 0 Total changes: 107	
ОК	

Fig. 18. Registry and file analysis of WhisperGate using regshot.

A total of 107 changes were observed, and the output file displays useful information, such as the addition and deletion of files and registry changes.

In the final ADA step, the IDA tool was used to obtain a debug view of the sample. The IDA tool creates maps of software's execution to display the binary instructions that are actually executed by the processor. The real-time import was used to obtain useful information, as shown in Fig. 19.

14-1-001011100	- Comment Annual Technology
	; Segment type: Externs
.idata:0040A130	imp CloseHandle dd offset kernel32 CloseHandle
.idata:0040A130	: DATA XREF: CloseHandletr
	imp CreateFileN dd offset kernel32 CreateFileN
.idata:0040A134	: DATA XREF: CreateFileWfr
.idata:0040A138	imp_DeleteCriticalSection dd offset unk_778B8CD0 : DATA XREF: DeleteCriticalSection1r
idata:0040413C	imp_EnterCriticalSection dd offset unk_77BAAFC0
	; DATA XREF: EnterCriticalSectionfr
	imp_ExitProcess dd offset kernel32_ExitProcess
.idata:0040A140	; DATA XREF: ExitProcessfr
	imp_FindClose dd offset kernel32_FindClose
.idata:0040A144	; DATA XREF: FindClosetr
	imp_FindFirstFileA dd offset kernel32_FindFirstFileA
.idata:0040A148	; DATA XREF: FindFirstFileAfr
	imp_FindNextFileA dd offset kernel32_FindNextFileA
.idata:0040A14C	; DATA XREF: FindNextFileAfr
	imp_FreeLibrary dd offset kernel32_FreeLibrary
.idata:0040A150	; DATA XREF: FreeLibraryfr
	imp_GetCommandLineA dd offset kernel32_GetCommandLineA
.idata:0040A154	; DATA XREF: GetCommandLineAfr
	imp_GetLastError dd offset kernel32_GetLastError
.idata:0040A158	; DATA XREF: GetLastErrontr
.idata:0040A15C	imp_GetModuleHandleA dd offset kernel32_GetModuleHandleA

Fig. 19. The debug view of WhisperGate using the IDA tool.

Three vital imported files, CreateFile, CommandLine, and WriteFile, were observed. Out of the 53 imports, four imported an msvcrt library, and the remaining imported a KER-NEL32 library. After creating a breakout for the WriteFile, the file resumed at the GetCommandLine parser. The "stepinto" feature of the tool provides a view of the kernel-based library (Fig. 19).

3) Hybrid analysis: Hybrid malware analysis overcomes the shortcomings of individual malware analysis types, as relying on a single malware analysis method does not provide a comprehensive malware analysis report. In this stage, the information generated using the tools Ghidra and Procmon were integrated. Procmon is a utility for Microsoft Windows OS that captures and displays system and network activity. The static analysis provided an overview of the static code along with the basic properties, while dynamic analysis was applied simultaneously to strengthen the information gathered during the static analysis phase. As soon as the file was run, its behavior could be monitored and compare with the relevant information extracted from the static code analysis. This process highlights the importance of synchronization of both static and dynamic analyses (Fig. 20).

<pre>(undefined4 *) (sparam_1 + iVar1) = 0;</pre>
<pre>(undefined4 *) (sstack0x00000000 + iVarl) = 0;</pre>
<pre>(undefined4 *)((int)sp_Stack4 + iVarl) = 3;</pre>
<pre>(undefined4 *) (sstack0xfffffff8 + iVarl) = 0;</pre>
<pre>(undefined4 *) (sstack0xfffffff4 + iVarl) = 3;</pre>
<pre>(undefined4 *) (sstack0xfffffff0 + iVarl) = PERM ALL;</pre>
(wchar t **) (sstack0xffffffec + iVarl) = L"\\\\.\\PhysicalDrive0";
<pre>(undefined4 *)((int)suStackY24 + iVarl) = 0x403bcf;</pre>
ovVar3 = CreateFileW(*(LPCWSTR *)(sstack0xffffffec + iVar1),*(DWORD *)(sstack0xfffffff0 + iVar1
*(DWORD *)(&stack0xfffffff4 + 1Varl),
*(LPSECURITY ATTRIBUTES *)(sstack0xfffffff8 + iVarl),
*(DWORD *)((int)sp Stack4 + iVarl),*(DWORD *)(sstack0x00000000 + iVarl),
*(HANDLE *)(sparam 1 + iVarl));
(HAMDLE *) (sstackOxffffffec + iVarl) = pvVar3;
(undefined4 *)((int)sp Stack4 + iVarl) = 0;
(undefined4 *) (sstack0xfffffff8 + iVar1) = 0;
(undefined4 *)(sstack0xfffffff4 + iVarl) = 0x200;

Fig. 20. Result of ghidra illustrating the buffer size.

The buffer size in Hexadecimal code 0x200 (last line in Fig. 20), is revealed as 512 in decimal. The first 512 bytes are equal to the exact size of the MBR buffer. The buffer contains the string "Your hard drive has been corrupted." It is possible that the sample made an effort to corrupt the MBR, but this hypothesis could not be confirmed without performing a hybrid analysis.

To add value to the information, the thread activity overwriting the device's hard disk with a length of 512 bytes was analyzed. The images shown below (Fig. 21 and 22) provide details such as operation, path, offset value, and the result status of the operation.

C:\Users\IEUser\E	Desktop\WhisperGate\Whispe	rGate\Stage1.exe rrentControlSet\Servi	SUCCESS	
F Stage 1.exe	6524 RegEnumKey	HKLM\System\CurrentControlSet\Servi	NO MORE ENTRI.	Index: 1, Length: 80
• Stage 1.exe	6524 RegCloseKey	HKLM\System\CurrentControlSet\Servi	SUCCESS	
■ Stage 1.exe	6524 🐂 Create File	\Device\Harddisk0\DR0	SUCCESS	Desired Access: Al Access, Disposition: Open, Options: Synchrono
Stage 1.exe	6524 WriteFile	\Device\Harddisk0\DR0	SUCCESS	Offset: 0, Length: 512, I/O Flags: Non-cached, Priority: Normal
Stage1.exe	6524 🐂 CloseFile	\Device\Harddisk0\DR0	SUCCESS	In second s

Fig. 21. The thread activity showing the overwriting of the hard disk.

Event Properti	es	-	×
🐓 Event	Process Stack		
Date:	4/4/2022 2:23:03.9166660 AM		
Thread:	6528		
Class:	File System		
Operation:	WriteFile		
Result:	SUCCESS		
Path:	\Device\Harddisk0\DR0		
Duration:	0.0386311		
Offset:	0		
Length:	512		
I/O Flags: Priority:	Non-c Norma		

Fig. 22. Event view of the result of the thread activity (see Fig. 21).

The thread confirms the successful execution of the operation WriteFile in overwriting 512 bytes of memory in the hard disk. By synchronizing the use of static code obtained under advanced static analysis and the behavioral characteristic observed under the dynamic analysis, critical information about the malware were successfully gathered. Specifically, the nature of the malware is to write the 512 bytes of the hard disk and corrupt the MBR.

4) Memory analysis: This phase involves a two-step approach: memory acquisition and memory dump analysis. First, a memory dump of the infected state was obtained, and then the analysis was completed by analyzing this memory dump. In the memory acquisition step, the tool AccessData FTK Imager was used to capture the memory dump of the infected state, as shown in Fig. 23. AccessData FTK Imager is a computer forensics software that can create copies, or forensic images of computer data without making changes to the original evidence.

iginar evide.	
Memory Progres	s
Destination:	C:\Users\IEUser\Desktop\WhisperGate\MemoryDump\Stag
Charles	·
Status:	Memory capture finished successfully
	· · · · · · · · · · · · · · · · · · ·
	Close
	Close

Fig. 23. Acquisition of the memory using the tool AccessData FTK imager.

Subsequently, the tool Volatility was used in the memory dump analysis step to analyze the memory dump of the infected state, which provided valuable information on the running processes (Fig. 24) and the mapping of physical offsets to virtual addresses (Fig. 25). Volatility is a command line memory analysis and forensics tool for extracting and analyzing the volatile data that is temporarily stored in random access memory from memory dumps.

rogri	ss: 100		PDB scanning fi								
		ImageFileName	Offset(V)	Threads	Handles		Wow64	CreateTime	ExitTi		
		System 0xc30dc						7:27.000000		Disabled	
		Registry	0xc30dc9ab3040			False	2022-04	-07 02:47:24.	000000		Disabled
100		smss.exe	0xc30dd8ee5040			False		-07 02:47:27.			Disabled
88	380	csrss.exe	0xc30de1e86140					-07 02:47:28			Disabled
	388	wininit.exe	0xc30dcc2df140			False		-07 02:47:29.			Disabled
64		csrss.exe	0xc30de07063c0			False		-07 02:47:29.			Disabled
		winlogon.exe	0xc30de0729200				2022-04	-07 02:47:29	000000		Disabled
68		services.exe	0xc30de072e080	18		False	2022-04	-07 02:47:29.	000000		Disabled
68		lsass.exe	0xc30de0731080				2022-04	-07 02:47:29.	000000		Disabled
68		fontdryhost.ex	0xc30dd4774080			False	2022-04	-07 02:47:29	000000		Disabled
	568	sychost.exe	8xc38dd4786888			False	2022-04	-07 02:47:29	666666		Disabled
se		sychost.exe	0xc38dd47af240			False	2822-84	-07 02:47:29	000000		Disabled
180		dum.exe 0xc30dd	18e89080 14				-07 02:4	7:29.000000		Disabled	
68	560	sychost.exe	0xc30dde65c240			False	2022-04	-07 02:47:30.	666666		Disabled
88	560	sychost.exe	0xc30dde658880			False	2822-84	-07 02:47:30	000000		Disabled
084	560	sychost.exe	0xc30dde66f080			False	2022-04	-07 02:47:30	000000		Disabled
	560	sychost.exe	0xc30dde6b5540			False	2022-04	-07 02:47:30.	666666		Disabled
48	560	sychost.exe	0xc30dde6d2080			False	2022-04	-07 02:47:30	666666		Disabled
096	560	sychost.exe	0xc30dc95a9080			False	2022-04	-07 02:47:30	000000		Disabled
200	560	sychost.exe	0xc30dc95d9880			False	2822-84	-87 02:47:30.	666666		Disabled
240		sychost.exe	0xc30dcd3d80c0				2022-04	-07 02:47:30	000000		Disabled
	1004	helperservice.	0xc30dcd3e2080			False	2022-04	-07 02:47:31.	000000		Disabled
388	560	VBoxService.ex	0xc30de1539080			False	2022-04	-07 02:47:31.	000000		Disabled
		MemCompression	0xc30de15b7040				2022-04	-06 14:17:32	000000		Disabled
		sychost, exe	0xc30de15c30c0				2022-04	-06 14:17:32.	000000		Disabled
		sychost.exe	0xc30dcf91e080				2822-64	-06 14:17:34.	000000		Disabled
784	560	sychost, exe	0xc30dcef86080				2022-04	-06 14:17:34	000000		Disabled
852	560	spoolsv.exe	0xc30dcf9c4080			False	2022-04	-06 14:17:35	000000	N/A	Disabled
	560	sychost, exe	0xc30dcfc26080			False	2022-04	-06 14:17:35	000000		Disabled
ARR	560	sychost, exe	0xc30dcfcdb080					-06 14:17:36.			Disabled

Fig. 24. View of the running processes using the tool volatility.

C:\Users\IEUser\Desktop\\		lop>vol.py -f Stag	e1.mem windows.virtmap
Volatility 3 Framework 2.			
Progress: 100.00	PDB scan	ning finished	
Progress: 100.00 Region Start offset E	ind offset		
MiVaBootLoaded 0xa900000	100000 0x100000	00000	
MiVaDriverImages @			
MiVaHal 0xa70000000000			
MiVaNonPagedPool 6		x38000000000	
MiVaPagedPool 0xf895e2			
MiVaPfnDatabase 0x808000			
MiVaProcessSpace @	xd6800000000	x100000000000	
MiVaSessionGlobalSpace @	axd5000000000000000000000000000000000000	000000000000	
MiVaSessionSpace 6			
MiVaSpecialPoolNonPaged @	0000000000000000000	0000000000000000	
MiVaSpecialPoolPaged @			
MiVaSystemCache 0xed80000	000000 0x100000	0000	
MiVaSystemPtes 0xf40000	000000 0x800000	0000	
MiVaUnused 0xc300000	000000 0x100000	00000	

Fig. 25. View of the mapping of physical offsets to virtual addresses using the tool volatility.

While the images show a list of processes running, the Stage1.exe file is not visible. This indicates that the file has been executed with immediate effect to remove its traces. The virtual mapping provides an overview of the start and close offsets of different regions, such as BootLoaded DriverImages.

B. Experimental Findings - BlackCat

BlackCat malware can be analyzed in a similar manner. However, the analysis is not restricted to the tools which were used to analyze WhisperGate; rather, an analyst can use alternate tools as per the situation, but it should be ensured that the necessary parameters are evaluated according to the target. This section focuses on the findings resulting from the analysis of the BlackCat malware sample.

1) Static analysis: The static analysis followed a three-step approach: deobfuscation, BPA, and ASA. Deobfuscation using the tool ExeinfoPE revealed that the file was not packed (Fig. 26).

Eile : E	BlackCat Sample.e	xe			₽н	
Entry Point	000014C0	00 <	EP Section :	.text		
File Offset :	000008C0]	First Bytes :	83.EC.0C.C7.0!	0	Plu
Linker Info :	2.30]	SubSystem :	Windows GUI	PE	G
File Size :	0022D000h	< <u>N</u>	Overlay :	NO 00000000	0	5
Image is 32	bit executable		RES/OVL : 0	/0% 2021	家	
CCC MINCH	N-64w compiler fr	r 32 / 64 h	it Windows (e	xe) - http://ming	Scan / t	Ri

Fig. 26. Deobfuscation of BlackCat using exinfo PE.

The tools HxD and Pestudnio were used in the BPA for file and signature analysis. The Hex view of the sample is provided in Fig. 27.

🕫 HxD - [(C:\L	Jsers\	EUse	r\De	skto	p\W	hispe	rGat	e\Bla	ackC	at\BI	ackC	at Sa	ampl	e.exe	1		
🕼 File E																		
					15	· .		015										-
📄 🔂 🔻			C.	C)	•	+ +	16	`	< \	Vind	ows	(ANS	SI)		\sim	hex		\checkmark
🕼 BlackC	at S	ample	.exe															
Offset	(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	OF	Decoded text
001A67	60	65	78	74	22	ЗA	22	3E	3E	20	49	6E	74	72	6F	64	75	ext":">> Introdu
001A67	70	63	74	69	6F	6E	5C	6E	5C	6E	49	6D	70	6F	72	74	61	ction\n\nImporta
001A67	80	6E	74	20	66	69	6C	65	73	20	6F	6E	20	79	6F	75	72	nt files on your
001A67	90	20	73	79	73	74	65	6D	20	77	61	73	20	45	4E	43	52	system was ENCR
001A67	0A	59	50	54	45	44	20	61	6E	64	20	6E	6F	77	20	74	68	YPTED and now th
001A67	B0	65	79	20	68	61	76	65	20	68	61	76	65	20	5C	22	24	ey have have ∖"\$
001A67	C0	7B	45	58	54	45	4E	53	49	4F	4E	7D	5C	22	20	65	78	{EXTENSION}\" ex
001A67	DO 1	74	65	6E	73	69	6F	6E	2E	5C	6E	49	6E	20	6F	72	64	tension.\nIn ord
001A67	'E0	65	72	20	74	6F	20	72	65	63	6F	76	65	72	20	79	6F	er to recover yo
001A67	FO	75	72	20	66	69	6C	65	73	20	79	6F	75	20	6E	65	65	ur files you nee
001A68	00	64	20	74	6F	20	66	6F	6C	6C	6F	77	20	69	6E	73	74	d to follow inst
001A68	810	72	75	63	74	69	6F	6E	73	20	62	65	6C	6F	77	2E	5C	ructions below.\
001A68	820	6E	5C	6E	ЗE	3E	20	53	65	6E	73	69	74	69	76	65	20	n\n>> Sensitive
001A68	30	44	61	74	61	5C	6E	5C	6E	53	65	6E	73	69	74	69	76	Data\n\nSensitiv
001A68	840	65	20	64	61	74	61	20	6F	6E	20	79	6F	75	72	20	73	e data on your s
001A68	850	79	73	74	65	6D	20	77	61	73	20	44	4F	57	4E	4C	4F	ystem was DOWNLO
001A68	60	41	44	45	44	20	61	6E	64	20	69	74	20	77	69	6C	6C	ADED and it will
001A68	870	20	62	65	20	50	55	42	4C	49	53	48	45	44	20	69	66	be PUBLISHED if
001A68	80	20	79	6F	75	20	72	65	66	75	73	65	20	74	6F	20	63	you refuse to c
001A68	90	6F	6F	70	65	72	61	74	65	2E	5C	6E	5C	6E	44	61	74	ooperate.\n\nDat
001A68	BAO	61	20	69	6E	63	6C	75	64	65	73	ЗA	5C	6E	2D	20	45	a includes:\n- E
001A68	BO	6D	70	6C	6F	79	65	65	73	20	70	65	72	73	6F	6E	61	mployees persona
001A68	8C0	6C	20	64	61	74	61	2C	20	43	56	73	2C	20	44	4C	2C	1 data, CVs, DL,
001A68	DO	20	53	53	4E	2E	5C	6E	2D	20	43	6F	6D	70	6C	65	74	SSN.\n- Complet
001A68	EO	65	20	6E	65	74	77	6F	72	6B	20	6D	61	70	20	69	6E	e network map in
001A68	FO	63	6C	75	64	69	6E	67	20	63	72	65	64	65	6E	74	69	cluding credenti
001A69		61	6C	73	20	66	6F	72	20	6C	6F		61	6C	20	61	6E	als for local an
001A69	10	64	20	72	65	6D	6F	74	65	20	73	65	72	76	69	63	65	d remote service
001A69		73	2E	5C	6E	2D	20	46	69		61		63	69	61	6C	20	s.\n- Financial
001A69				66	6F	72	6D	61	74	69	6F	6E	20	69		63	6C	information incl
001A69		75	64		6E	67					65		74			64	61	uding clients da
001A69		74	61	2C		62	69	6C	6C		2C		62	75	64	67	65	ta, bills, budge
001A69		74	73	2C	20		6E		75		6C		72	65	70	6F	72	ts, annual repor
001A69				_	20	_											65	ts, bank stateme
001A69					2E													nts.\n- Complete

Fig. 27. Decoded text message of BlackCat using Hxd.

The findings revealed the malware to be a PE file with the following decoded text, as shown in Fig. 27: "Important files on your system were ENCRYPTED, and now they have, have. In order to recover your files, you need to follow the instructions below. Sensitive data on your system were downloaded, and they will be published if you refuse to cooperate. Data include: employees' personal data, CVs, DL, SSN...Caution: do not modify files yourself. Do not use third party software to restore your data. You may damage your files; it will result in permanent data loss. Your data are strongly encrypted; you cannot decrypt it without a cipher key."

settings about		
■×8 %		
cr\users\ieuser\desktop\whispergate\b at indicators (31)	property	value
virustotal (52/70)	md5	AEA5D3CCED6725F37E2C3797735E6467
- b dos-header (64 bytes)	sha1	087497940A41D96E4E90786DC92F75F4A38D861A
dos-stub (64 bytes)	sha256	3D7CF20CA6476E14E0A026F9BDD8FF1F26995CDC5854C3ADB41A6135EF11BA83
→ > rich-header (n/a)	md5-without-overlay	n/a
file-header (Nov.2021)	sha1-without-overlay	n/a
optional-header (GUI)	sha256-without-overlay	n/a
directories (3)	first-bytes-hex	4D 5A 90 00 03 00 00 00 04 00 00 0FF FF 00 00 B8 00 00 00 00 00 00 40 00 00 00 00 00 00
sections (virtualized)	first-bytes-text	M Z
	file-size	2281472 (bytes)
	size-without-overlay	n/a
gexports (n/a)	entropy	6.863
	imphash	725A214FDF28F8DF86F69D9A6F865588
🗔 .NET (n/a)	signature	n/a
🚰 resources (n/a)	entry-point	83 EC 0C C7 05 38 E5 62 00 01 00 00 00 E8 BE E5 18 00 83 C4 0C E9 96 FC FF FF 8D B6 00 00 00 00 8
-atc strings (size)	file-version	n/a
	description	n/a
🔚 manifest (n/a)	file-type	executable
- ursion (n/a)	cpu	32-bit
- certificate (n/a)	subsystem	GUI
overlay (n/a)	compiler-stamp	0x6196252C (Thu Nov 18 02:04:28 2021)
	debugger-stamp	n/a
	resources-stamp	n/a
	import-stamp	0x00000000 (empty)
	exports-stamp	n/a
	version-stamp	n/a
	certificate-stamp	n/a

Fig. 28. PeStudio tool identifying the type of file.

After the signature identification, the tool PeStudio identified the type of file (Fig. 28). Based on the results, the file is built on a 32-bit CPU architecture with a file size of 2281472 bytes. In terms of hashing, the following values were obtained:

MD5 : AEA5D3CCED6725F37E2C3797735E6467

SHA-256: 087497940A41D96E4E907B6DC92F75F4 A38D 861°

SHA-1 : 3D7CF20CA6476E14E0A026F9BDD8FF1F26995C DC5854C3A

DB41A6135EF11BA83

The file was identified as a ransomware using Virustotal in the APU-based identification step, with a community score of 52/70.

String analysis through the tool PeStudio (Fig. 29) identified 13454 strings with 73 blacklisted. It was observed that the File-offset 0x0022C514 has a string value WriteFile. This is a critical finding that can assist in the advanced static analysis process.

encoding (2)	size (bytes)	file-offset	blacklist (73)	hint (208)	group (16)	value (13454)
ascii	7	0x0022C5DE	×	utility	network	connect
ascii	4	0x0022C628	×	utility	network	send
ascii	19	0x0022C044	×	import	synchronization	GetOverlappedResult
ascii	25	0x0022C20E	×	import	synchronization	QueryPerformanceFrequency
ascii	15	0x0022BE30	×	import	storage	FindVolumeClose
ascii	30	0x0022C4E2	×	import	storage	Wow64DisableWow64FsRedirection
ascii	14	0x0022BB24	×	import	services	ControlService
ascii	16	0x00228868	×	import	security	OpenProcessToken
ascii	22	0x0022C0DA	×	import	reckoning	GetTimeZoneInformation
ascii	16	0x0022C52C	×	import	network	NetApiBufferFree
ascii	13	0x0022C540	×	import	network	NetServerEnum
ascii	12	0x0022C550	×	import	network	NetShareEnum
ascii	10	0x0022C58C	×	import	network	WSACleanup
ascii	15	0x0022C59A	×	import	network	WSAGetLastError
ascii	10	0x0022C5BA	×	import	network	WSAStartup
ascii	11	0x0022C5D0	×	import	network	closesocket
ascii	12	0x0022C5E8	×	import	network	freeaddrinfo
ascii	11	0x0022C5F8	×	import	network	getaddrinfo
ascii	11	0x0022C606	×	import	network	ioctlsocket
ascii	8	0x0022C61C	×	import	network	recvfrom
ascii	10	0x0022C63A	×	import	network	setsockopt
ascii	28	0x00228FAE	×	import	file	GetFileInformationByHandleEx
ascii	13	0x0022C182	×	import	file	MapViewOfFile
ascii	15	0x0022C488	×	import	file	UnmapViewOfFile
ascii	9	0x0022C514	×	import	file	WriteFile
ascii	24	0x00228D14	×	import	execution	CreateToolhelp32Snapshot
accia	10	0-00228500		import	execution	GetCurrentProcessId

Fig. 29. String analysis of BlackCat using PeStudio.

In the advanced static analysis step, the tool IDA revealed the presence of GetCommanLineW, indicating the intended behavior of the sample when it utilized the command line for a specific task (Fig. 30).

No00000	FreeConsole	KERNEL32
20000000	FreeEnvironmentStringsW	KERNEL32
20000000	FreeLibrary	KERNEL32
10000000	GetCommandLineW	KERNEL32
M 0000000	GetComputerNameW	KERNEL32

Fig. 30. Advanced static analysis step results using the tool IDA.

2) Dynamic analysis: To analyze the malware behavior, the tool ApateDNS was used to monitor the DNS requests generated by the malware. However, no legitimate responses were identified (Fig. 31).

Capture Wind	low DNS Hex View		
Time	Domain Requested	DNS Returned	,
22:42:26	array612.prod.do.dsp.mp.microsoft.com	FOUND	
22:42:27	98.100.138.213.in-addr.arpa	FOUND	
22:42:33	10.62.76.144.in-addr.arpa	FOUND	
22:42:40	203.188.99.203.in-addr.arpa	FOUND	
22:42:49	client.wns.windows.com	FOUND	
22:42:49	79.15.196.201.in-addr.arpa	FOUND	
22:43:10	4ff.1.3.e.e.6.3.3.a.2.6.d.c.3.0.0.0.0.0.0.0	0.0.0.0.8.e.f.ip6.arpa FOUND	
22:43:10	4ff.1.3.e.e.6.3.3.a.2.6.d.c.3.0.0.0.0.0.0.0	0.0.0.0.8.e.f.ip6.arpa FOUND	
22:43:11	135.182.99.203.in-addr.arpa	FOUND	
22:43:19	123.93.58.176.in-addr.arpa	FOUND	
	127.0.0.1 as return DNS IP! : to 127.0.0.1 on Intel(R) PRO/1000 M		
+] Server	<pre>valid DNS response of first request started at 22:37:33 successfully. sh IB (Default Compart Gaturar/DNS): 1270</pre>		
+] server			

Fig. 31. DNS spoofing of BlackCat using the tool ApateDNS.

The snapshot comparison tool Regshot was used to compare the snapshot of the registry before and after executing the executable (Fig. 32). The snapshots indicate the changes in the keys and values.

💑 &1st shot 🛛 🗙	💑 &2nd shot 🛛 🗙 🗙
Datetime: 2022-04-05 05:34:42 Computer: MSEDGEWIN10 Username: IEUser Keys: 473047 Values: 848680 Dirs: 0 Files: 0	Datetime: 2022-04-05 05:44:18 Computer: MSEDGEWIN10 Username: IEUser Keys: 473049 Values: 848672 Dirs: 0 Files: 0
ОК	ОК

Fig. 32. The pre- and post-execution snapshots using regshot.

The tool Process Monitor was used for the process analysis, which revealed that 389633 processes were triggered on malware execution (Fig. 33). The CreateFile process was highlighted, but no such evidence of WriteFile was produced.

Re fail first filte	Tell Average Party Int		5 0 (2
		8 8 P 7 E R	
In Frank Saul	¥0 Dandari	Edu	Bud Md
10. Characteristic 17. Characteristic 17. <td>1012 of Load Heap 1012 of Load Heap 1012 of Load Heap 1012 III Teg/Dentife; 1012 III Teg/Dentife; 1012 III Teg/Dentife; 1013 III Teg/Dentife; 1014 Teg/Dentife; 1014 Teg/Dentife; 1015 III Teg/Dentif</td> <td>HEM Synam Consultance Ser Care HEBMS year Consultance Ser Care HEBMS year Consultance Ser Care HEBMS year Consultance Ser Care HEBMS STEP Consultance Ser Care</td> <td>Look Teach and a Concertor To Tomo The York and Mandal Cale Cale Concertor Town of the West Street S</td>	1012 of Load Heap 1012 of Load Heap 1012 of Load Heap 1012 III Teg/Dentife; 1012 III Teg/Dentife; 1012 III Teg/Dentife; 1013 III Teg/Dentife; 1014 Teg/Dentife; 1014 Teg/Dentife; 1015 III Teg/Dentif	HEM Synam Consultance Ser Care HEBMS year Consultance Ser Care HEBMS year Consultance Ser Care HEBMS year Consultance Ser Care HEBMS STEP Consultance Ser Care	Look Teach and a Concertor To Tomo The York and Mandal Cale Cale Concertor Town of the West Street S
EXE DecideSand	1012 Concerto Ter	T Mindow States Meson Office #	VANE VET DE VAD Destrue Accesse Fand Anthunes, Dasconters Course Courses Courses Texture and Course Note: Texture Accesses Fand Anthunes Courses Fand Anthunes
12 - Conteined 13 - Conteined 14 - Conteined 15 - Conteined	552 in Deate Pe 552 in DeepHarante 552 in DeepHarante 552 in December 553 in RecDenny Jaho	2 (Modowa 2 (Modowa 2 (Modowa 2 (Modowa) 2 (Modowa) (Modowa) (Modowa) 2 (Modo	1.0000 There is base is failed and an information Department of the failed of the failed on a Developer Table Accessible to 1.0000 The set Means Table 1.0000
1 8 The state of 1 8 The state	1972 Michael Angliant Angliant 1972 Cland Angli 1972 Michael Angliant 1972 Michael Angliant 1972 Michael Angliant 1972 Michael Angliant 1972 Michael Angliant	HEREOFFICIES and the set of the des- HEREOFFICE STATE is a set of the des- HEREOFFICE STATE and the set of the set of the HEREOFFICE STATE Conservations and the HEREOFFICE CONSERVATION AND THE HEREOFFICE CONSERVAT	s colone Locates la la la colona de la la colona de la c

Fig. 33. The result from the tool process monitor with CreateFile highlighted.

In the subsequent step, the tool Regshot was used for registry and file analysis, which highlighted 138 changes (Fig. 34)

🛓 C&ompare		×
Keys deleted: 13 Keys added: 15 Values deleted: 36 Values added: 28 Values modified: 4 Folders added: 0 Folders attributes Files deleted: 0 Files (attributes?] n Total changes: 138	- changed: 0 nodified: 0	
	OK	

Fig. 34. Registry and file analysis using the tool regshot.

Using the ADA tool IDA, the snapshot revealed that the GetCommandLineW process imported a kernel-based library (Fig. 35).

🛄 🖆 🖼	
.text:0058F620	
.text:0058F620	
.text:0058F620	; Attributes: thunk
.text:0058F620	
.text:0058F620	; LPWSTRstdcall GetCommandLineW()
.text:0058F620	GetCommandLineW proc near
.text:0058F620	jmp ds: imp SetCommandLinen
.text:0058F620	GetCommandLineW endp
.text:0058F620	

Fig. 35. Results of the snapshot using the tool IDA.

3) Hybrid analysis: In static analysis, information was gathered about the GetCommandLineW call. Through dynamic code analysis, the complete code was debugged to extract some useful keys: "h," "p," "e," "-," and "l." By using the command prompt feature running ProcMon, it was surmised that the keys could be the instructions used in the command prompt, which could be executed to further examine the intended purpose (Fig. 36). Here, the command prompt executes the sample and passes a log file to a particular directory.

07:13:3	2 [INFO]	locker::core::cluster: terminating
07:13:3	2 [INFO]	locker::core::cluster: terminated
07:13:3	2 [INFO]	locker::core::renderer: Speed: 0.01 Mb/s, Data: 0Mb/0Mb, Files processed: 7/7, Files scanned: 32
07:13:3	3 [INFO]	locker::core::renderer: Speed: 0.01 Mb/s, Data: 0Mb/0Mb, Files processed: 7/7, Files scanned: 32
07:13:3	3 [INFO]	locker::core::renderer: Time taken: 52.728618s
07:13:3	3 [INFO]	locker::core::stack: Platform Shutdown
07:13:3	3 [INFO]	encrypt app::windows: Shutdown Routine
		encrypt_app::windows: Dropping Note and Wallpaper Image
07:13:3	3 [INFO]	locker::core::windows::desktop_note: set_desktop_image=C:\Users\IEUser\Desktop\RECOVER-sykffle=FILES.
txt.png		
		locker::core::windows::desktop_note: deploy_note_and_image_for_all_users=C:\Users\Default
		locker::core::windows::desktop_note: deploy_note_and_image_for_all_users=C:\Users\Default_User
		locker::core::windows::desktop_note: deploy_note_and_image_for_all_users=C:\Users\IEUser
		encrypt app::windows: Trying to remove shadow copies
07:13:3	3 [INFO]	locker::core::windows::shadow_copy: shadow_copy::remove_all=2

Fig. 36. Results from the command prompt running ProcMon.exe.

		옮 🖗 🔎 利 📑 📷	2 % 1	
Time Process Name	PID Operation	Path	Result	Detail
12.12 ElackCatSampl	6440 The WriteFile	C:\Users\IEUser\Desktop\log.bd	SUCCESS	Offset: 71, Length: 2, Priority: Normal
12.12 FBlackCatSampl		C:\Users\IEUser\Desktop\log.txt	SUCCESS	Offset: 73, Length: 19, Pronty: Normal
12:12: ElackCatSampl	6440 In WrteFie	C:\Users\JEUser\Desktop\log.bd	SUCCESS	Offset: 92, Length: 2, Priority: Normal
12.12. BlackCatSampl_	6440 in WiteFile	C:\Users\IEUser\Desktop\log.txt	SUCCESS	Offset: 94, Length: 19, Priority: Normal
12:12: BlackCatSampl	6440 The Write File	C:\Users\JEUser\Desktop\log.txt	SUCCESS	Offset: 113. Length: 1. Priority: Normal
12:12: ElackCatSampl			SUCCESS	Thread ID: 4216
12.12 ElackCatSampl_	6440 In WiteFile	C:\Users\JEUser\Desktop\log.txt	SUCCESS	Offset: 114, Length: 8, Priority: Normal
12-12: ElackCatSampl	6440 The Write File	C:\Users\JEUser\Desktop\log.txt	SUCCESS	Offset: 122. Length: 1. Priority: Normal
12:12 BlackCatSampl	6440 In WriteFile	C:\Users\JEUser\Desktop\log.bt	SUCCESS	Offset: 123, Length: 1, Priority: Normal
12.12 BlackCatSampl	6440 in WiteFile	C:\Users\IEUser\Desktop\log.txt	SUCCESS	Offset: 124, Length: 4, Priority: Normal
12-12 BlackCatSampl	6440 The Write File	C:\Users\JEUser\Desktop \log bit	SUCCESS	Offset: 128. Length: 2. Priority: Normal
12:12: BlackCatSampl	6440 In WriteFie	C:\Users\JEUser\Desktop\log.txt	SUCCESS	Offset: 130. Length: 19. Pronty: Normal
12:12: BlackCatSampl	6440 The WriteFile	C:\Users\IEUser\Desktop\log.bt	SUCCESS	Offset: 149. Length: 2. Priority: Normal
12.12 BlackCat Sampl	6440 The Write File	C:\Users\JEUser\Desktop\log.txt	SUCCESS	Offset: 151, Length: 33, Priority: Normal
12:12: ElackCatSampl	6440 in WrteFie	C:\Users\IEUser\Desktop\log.txt	SUCCESS	Offset: 184, Length: 1, Priority: Normal
12.12 BlackCatSampl	6440 ReadFile	C:\Users\JEUser\Desktop\BlackCatSa	SUCCESS	Offset: 1,115,136, Length: 16,384, I/O Flags: Non-cached, Paging I/O. 1
12:12 ElackCat Sampl	6440 The ReadFile	C:\Users\JEUser\Desktop\BlackCatSa	SUCCESS	Offset: 996.352, Length: 32,768, I/O Rags: Non-cached, Paging I/O, Sy
12:12: ElackCatSampl	6440 cP Thread Create		SUCCESS	Thread ID: 6432
12-12 BlackCat Sampl	6440 c ^O Thread Create		SUCCESS	Thread ID: 4248
12.12 ElackCat Sampl	6440 In WiteFie	C:\Users\JEUser\Desktop\log.bd	SUCCESS	Offset: 185. Length: 8. Priority: Normal
12.12. ElackCatSampl	6440 The WriteFile	C:\Users\IEUser\Desktop\log.txt	SUCCESS	Offset: 193. Length: 1. Pronty: Normal
12:12 BlackCat Sampl	6440 R WrteFie	C:\Users\JEUser\Desktop\log.bt	SUCCESS	Offset: 194, Length: 1, Priority: Normal
12.12 ElackCat Sampl	6440 In WiteFie	C:\Users\JEUser\Desktop\log.bt	SUCCESS	Offset: 195. Length: 4, Priority: Normal
12:12 BlackCat Sampl	6440 In WiteFie	C:\Users\JEUser\Desktop\log.txt	SUCCESS	Offset: 199. Length: 2. Priority: Normal
12-12 BlackCatSampl	6440 In WriteFie	C:\Users\JEUser\Desktop\log.bt	SUCCESS	Offset: 201, Length: 56, Priority: Normal
12.12 BlackCat Sampl	6440 In WiteFile	C:\Users\JEUser\Desktop\log.bt	SUCCESS	Offset: 257, Length: 2, Priority: Normal
12:12 BlackCatSampl		C:\Users\JEUser\Desktop\log.txt	SUCCESS	Offset: 259, Length: 16, Prority: Normal
12:12: BlackCatSampl	6440 R WrteFile	C:\Users\JEUser\Desktop\log.bt	SUCCESS	Offset: 275. Length: 1. Priority: Normal
12.12 BlackCatSampl	6440 In WiteFile	C:\Users\JEUser\Desktop\log.bt	SUCCESS	Offset: 275. Length: 1. Priority: Normal
12:12 BlackCatSampl	6440 c [®] Thread Create		SUCCESS	Thread ID: 6640
12:12: ElackCatSampl			SUCCESS	Thread ID: 4364
12:12 BlackCatSampl	6440 cP Thread Create		SUCCESS	Thread ID: 240
12.12 ElackCatSamel	6440 cD Thread Create		SHOTESS	Thread ID: 6768

Fig. 37. Results from the tool process monitor.

When using the Process Monitor tool (Fig. 37), it was observed that the malware triggered almost 1.5 times (568,995) the number of processes executed by the same malware when compared to dynamic analysis (389,633). This time, the WriteFile operation was evident and confirmed based on the process executed through the command prompt. The directory set during the execution was successfully injected with the log file, thus corrupting the services.

Incompany blies on your subject was ENCRIPTED.	Important Files on your system was ENURTHED.
for For the arc even we were the of sensitive information follow	Sensitive data on your system was IOWNLOADED. To recover your files and prevent publishing of sensitive information
And the set of the set	instructions in "+00TE_FILE_NMES" file.
The Adv Average Department of Steamberryour system was ENCRYPTED.	Important files on your system was ENCRIPTED.
Sensitive data on your system was DDHNLOADED.	Sensitive data on your system was DOWNLOADED.
To recover your films and reavont molitibing of sensitive information follow rue to s if (000 FILE)wHEP* file.	To recover your files and prevent publishing of sensitive information instructions in "\$90TE_FILE_NMEE" file.
Senarcive With on our sector was DIANLOADED.	Important files on your system was ENCRYPTED. Sensitive data on your system was DOWNLOADED.
""To recover 'goor files and prevent poolishing of sensitive information follow	To recover your files and prevent publishing of sensitive information
Instructions in "+ONTE_FILE_NAME3" file.	instructions in "400TE_FILE_NWEE" file. Important files on your susten was ENCRIPTED.
Se two to the two to the set of t	Sensitive data on your system was DOWNLOADED.
o provide the second se	To recove pur files and prevent publishing of sensitive information instructions in "100DE_FILE_NAME>" file.
Territor Till German INCONTAnt PITes Digour System was ENCRYPTED.	Operation of the state of th
Sensitive data on your system was DOMPLOADED.	Sensitive data on your system was DOWN_DADED.
To recover the and reverted of sensitive information follow	To recover your files and prevent publishing of sensitive information instructions in "4040TE_FILE_NWED" file.
In ant es pur tex vas ENCRYPTED.	Important files on your system was ENCRYPTED.
To recover your files and prevent publishing of sensitive information follow	Sensitive data on your system was DOWNLOADED. To recover your files and prevent publishing of sensitive information
	instructions in "\$90TE FILE NEED" file.
In Cant des on your stea was ENCRYPTED.	Important files on your system was ENCRYPTED.
. Se live la en vas DOANLOADED.	Sensitive data on your system was DOWNLOADED.
Manual	To recover your files and prevent publishing of sensitive information instructions in "400TE_FILE_NMME)" file.
Andrew TerlAportant Files on your system was ENCRIPTED. Sensitive data on your system was DOANLOADED.	Important files on your system was ENCRYPTED. Sensitive data on your system was DOWNLOADED.
o n wer the first ar seven whilehing of sensitive information follow	To recover your files and prevent publishing of sensitive information
CONTRACTOR AND A SNOTE FILE NAME?" File.	instructions in "\$00TE FILE NEED" file.
In Ing was our system was ENCRYPTED.	Important files on your system was ENCRYPTED.
providings with Separate ve date on gourna sugaren was DDANLOADED.	Sensitive data on your system was DOWNLOADED.
To recover your files and genevent publishing of sensitive information follow instructions in "\$(NOTE_FILE_NAME)" file.	To recover your files and prevent publishing of sensitive information instructions in "4040TE_FILE_NVMED" file_
In International States Sour System was ENCRYPTED.	Important files on your system was ENCRIPTED.
To sensitive a o ur system was DOANLOADED.	Sensitive data on your system was DOWNLOADED. To recover your files and prevent publishing of sensitive information
deline in the Constant strate in the constant of the international strategy of the constant of the international strategy of the constant files on your system was ENCRYPTED.	instructions in '\$00TE_FILE_NWES' File. Instructions un uner switch was ENCROPTED.

Fig. 38. The wallpaper image dropped through the execution of the malware using cmd.

Once the file was executed through the cmd, it took 53 seconds to corrupt the services, along with dropping a note and wallpaper image (Fig. 38). This demonstrated the speed and potency of the malware in infecting the system.

4) Memory analysis: The tool AccessData FTK was used to capture the memory dump of the infected state (Fig. 39) at the initial memory acquisition step.

Memory Progr	255
Destination:	C:\Users\IEUser\Desktop\BlackCat.mem
Status:	Memory capture finished successfully
	Close

Fig. 39. Capturing the memory dump using the tool AccessData FTK.

Next, in the memory dump analysis step, the Processlist successfully showed the execution of the BlackCat sample in the infected memory dump (Fig. 40).

	chrumeleke OxdideEllis	9248 8		Falsa 2022-04-07 04:48:59,000000	s/a Disabled	
	rmd.exe AxdSAeAArdf54A 1			2822-04-07 04:49:53.000000 10/6		
4 5169	conhost.exe 8xd58e88636	6546 4		False 2022 84 87 84:49:54,000000	M/A Disabled	
	Sectoritiese Endsteiner			Falsa 2022-04-07 04:50:01.000000	M/A Disabled	
	SecurityHealth RedSdeBdcr3			Faise 3622-84-87 84:50:18.806806	N/A Disabled	
	sychost.exe Exd58e88cc4			False 2022 04 07 04:50:41.000000	M/A Disabled	
	Searchfiltertto &addeetBuck			Talsa 2022-04-07 04:50:51,000000	N/A Disabled	
38 5169	BlackCatSample RedS8e71e35			False 2022-84-87 84:52:88.000000	2022-04-07 04:53:12.000000	Disabled
6 696	willPrySE.exc 8xd58e88428			False 2022-04-07 04:52:11.000000	N/A Disabled	
	Handlight.ece Stdiffelline			false 3022-04-07 04:52:43.000000	N/A Disabled	
	cmd.exe 6xd50c90743540 2			2822-04-07 04:53:09.000000 1//4	Disabled	
6 3788	conhost.exe 8xd58e81378	8248 B		False 2022-04-07 04:53:00.000000	2022-04-07 04:53:12.000000	Disabled
		3240 G	1. False			



Fig. 41. Image showing the ProcessList details for the malware sample.

The Processtree provides details such as the execution time and the offset value for the malware sample (Fig. 41). The command line operation shows the request to memory accessibility at a particular offset value (Process ID 3788 in Fig. 42).

5812	SecurityHealth	Required memory at 0x2c11bbc020 is inaccessible (swapped)
6116	svchost.exe	Required memory at 0x2000000096 is not valid (process exited?)
4532	SearchFilterHo	Required memory at 0x278af8b4d78 is not valid (process exited?)
3068	BlackCatSample	Required memory at 0x3bd020 is inaccessible (swapped)
5136	WmiPrvSE.exe	Required memory at 0x34ff692020 is not valid (process exited?)
2776	UsoClient.exe	
3788	cmd.exe Require	d memory at 0xc36a545020 is inaccessible (swapped)
4156	conhost.exe	Required memory at 0xc9a3e4020 is not valid (process exited?)
4156	conhost.exe	Required memory at 0xc9a3e4020 is not valid (process exited?)

Fig. 42. Image showing the result of the memory accessibility (PID 3788)

V. DISCUSSION

The proposed methodology is applicable for analyzing different malicious files. The case study provides a demonstration of malware analysis on WhisperGate and BlackCat. It is advisable for an analyst to prepare a summary report based on the experimental results of the sample. This section presents a report of the results relative to the two candidate pieces of malware used in the case study.

C. WhisperGate

The analysis of WhisperGate shows a deobsfuscated .exe file with 32-bit CPU architecture carrying threatening information in a static approach. While performing dynamic analysis through the registry modification, the impact of malware was also noticeable. Its nature was identified through hybrid analysis that used both static and dynamic processes in which the malware overwrote the MBR. Changes in the disk led by the malware sample were observed through the offset mapping to the bootloader and driver images (Table III).

TABLE III. SUMMARY REPORT OF WHISPERGATE MALWARE

Static	Dynamic	Hybrid	Memory
An unpacked	Post-execution impact	Static analysis	Virtual
.exe file with a	was visible in the	indicated a	mapping of
CPU	registry modification.	buffer 0x200	the offsets
architecture of	Along with a trigger of	equal to 512 in	related to the
32 bits and	2,58,678 events, the	decimal,	BootLoaded
threatening	WriteFile event brought	which is	and
strings	attention to the	indeed the size	DriverImages
(message) was	modification/overwriting	of MBR.	brought
identified. The	of the file.	Dynamic	attention to
API-generated		analysis	the possible
score of 54/69		generated a	changes in
indicates the		thread to	the disk.
presence of		overwrite 512	
characteristics		bytes of	
typical of		memory in the	
ransomware and		hard disk. This	
wipers. This		finding	
finding allowed		confirmed the	
for classifying		nature of	
the malware as a		malware	
suspicious file		corrupting the	
with a monetary		MBR through	
purpose.		overwriting	
		-	

Fig. 43 indicates the impact of running the malware sample WhisperGate showing the output "Your hard drive has been corrupted". The sample overwrites the MBR and displays a ransom note demanding \$10k via cryptocurrency ("You should pay us \$10k via bitcoin wallet"), thus validating the experimental analysis.

Your hard drive has been corrupted.		
In case you want to recover all hard drives		
of your organization,		
You should pay us \$10k via bitcoin wallet		
1AVNM68gj6PGPFcJuftKATa4WLnzg8fpfv and send message via		
tox ID 8BEDC411012A33BA34F49130D0F186993C6A32DAD8976F6A5D82C1ED23054C057ECED5496		
F65		
with your organization name.		
We will contact you to give further instructions.		

Fig. 43. Image showing the impact of executing WhisperGate.

D. BlackCat

Like WhisperGate, the static analysis of BlackCat showed a deobfuscated .exe file with a 32-bit CPU architecture carrying a threatening note. The API score indicates that the sample is ransomware with a monetary purpose. Through registry modification performed in a dynamic approach, the use of the command prompt by the sample was observed. During the hybrid approach, both key identifications used for executing the command prompt were performed by the malware along with a threatening message, indicating the malicious nature of the sample as ransomware. In memory analysis, the same result was revealed through the command line request for memory access (Table IV).

TABLE IV. SUMMARY REPORT OF BLACKCAT MALWARE

Fig. 44 indicates the impact of running the malware sample BlackCat. The sample corrupted the directory and displayed a background image with a threatening note of the ransom, thus validating the results of the experimental analysis performed using static, dynamic, hybrid, and memory analysis.

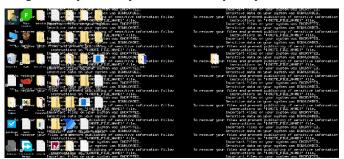


Fig. 44. Image showing the impact of executing BlackCat.

VI. CONCLUSION AND FUTURE WORK

The paper offers a comprehensive and practical approach to performing in-depth analyses of pseudo ransomware by illustrating two pieces of malware, namely WhisperGate (used in cyber warfare) and BlackCat (used to target critical organizations of target states). Twelve tools/techniques were selected, and a detailed description of the steps involved in the application, information extraction, analysis of results, and digital forensics is provided. The malware analysis was successfully executed through the use of static, dynamic, hybrid, and memory analysis and then validated. The detailed malware analysis using twelve tools revealed the embedded information and values in the malicious code for greater visibility and subsequent actions for information technology security personnel and forensic analysts. As malware attacks have rapidly risen with the appearance of innovative malware, the research demonstrated a successful methodology for analyzing potent malware through a comprehensive step-bystep approach. The work overcomes the limitations of relying on a single malware analysis technique thus providing a comprehensive approach to malware analysis.

WhisperGate came into the limelight at the beginning of 2022, when it was used to target multiple government and private organizations in Ukraine. The ransomware malware BlackCat was selected as a sample because it was reported to target European affiliations and U.S. organizations in late 2021. Out of the four malware analysis mentioned in the paper, hybrid analysis provided maximum information critical for the malware analyst to understand the extent of damage.

Three limitations have been observed in this study that can lead to further research. First, since only two pieces of malware (i.e., ransomware and pseudo ransomware) were observed, the experimented malware analysis methodology can be extended to diverse malware samples to validate the methodology. Secondly, since the study was limited to traditional malware analysis, appropriate machine learning methodologies can be deployed in future research to compare the findings with those obtained from traditional malware analysis. Thirdly, in this research, open-source tools were deployed for malware analysis that is already known to malicious hackers for circumventing the analysis process. Hence, future research can compare the results of open source tools with subscription based commercial tools.

REFERENCES

- J. Greig. "BlackCat ransomware targeting US European retail, construction and transportation orgs." ZD Net. https://www.zdnet.com/article/blackcat-ransomware-targeting-useuropean-retail-construction-and-transportation-orgs/ (accessed February, 2023).
- [2] O. Analytica, "Cyberattacks on European energy cement new dynamic," *Emerald Expert Briefings*, no. oxan-db, 2022.
- [3] World Economic Forum. "Global Cybersecurity Outlook 2022." World Economic Forum. https://www3.weforum.org/docs/WEF_Global_Cybersecurity_Outlook_ 2022.pdf (accessed January, 2023).
- [4] Centre for Internet Security. "Breaking Down the BlackCat Ransomware Operation." Center for Internet Security. (accessed February, 2023).
- [5] Microsoft Security. "The Many Lives of BlackCat Ransomware." Microsoft Inc. https://www.microsoft.com/enus/security/blog/2022/06/13/the-many-lives-of-blackcat-ransomware/

(accessed February, 2023).

- [6] E. Berrueta, D. Morato, E. Magaña, and M. Izal, "A survey on detection techniques for cryptographic ransomware," *IEEE Access*, vol. 7, pp. 144925-144944, 2019.
- [7] A. Kapoor, A. Gupta, R. Gupta, S. Tanwar, G. Sharma, and I. E. Davidson, "Ransomware detection, avoidance, and mitigation scheme: a review and future directions," *Sustainability*, vol. 14, no. 1, p. 8, 2021.
- [8] M. Sikorski and A. Honig, Practical malware analysis: the hands-on guide to dissecting malicious software. William Pollock, 2012.
- [9] H. Madani, N. Ouerdi, and A. Azizi, "Ransomware: Analysis of Encrypted Files,") *International Journal of Advanced Computer Science* and Applications, vol. 14, no. 1, pp. 213-217, 2023.
- [10] S. Alelyani and H. Kumar, "Overview of cyberattack on saudi organizations," *Journal of Information Security And Cybercrimes Research*, vol. 1, no. 1, pp. 42-50, 2018.
- [11] F. Mercaldo, V. Nardone, A. Santone, and C. A. Visaggio, "Ransomware steals your phone. formal methods rescue it," in *Formal Techniques for Distributed Objects, Components, and Systems: 36th IFIP WG 6.1 International Conference, FORTE 2016, Held as Part of the 11th International Federated Conference on Distributed Computing Techniques, DisCoTec 2016, Heraklion, Crete, Greece, June 6-9, 2016, Proceedings 36*, 2016: Springer, pp. 212-221.
- [12] U. Urooj, B. A. S. Al-rimy, A. Zainal, F. A. Ghaleb, and M. A. Rassam, "Ransomware detection using the dynamic analysis and machine learning: A survey and research directions," *Applied Sciences*, vol. 12, no. 1, p. 172, 2022.
- [13] P. Zavarsky and D. Lindskog, "Experimental analysis of ransomware on windows and android platforms: Evolution and characterization," *Proceedia Computer Science*, vol. 94, pp. 465-472, 2016.
- [14] M. Horduna, S.-M. Lăzărescu, and E. Simion, "A note on machine learning applied in ransomware detection," *Cryptology ePrint Archive*, 2023.
- [15] Trellix, M. Kersten, and R. Samani. "Return of Pseudo Ransomware." Trellix. (accessed February, 2023).
- [16] O. S. Carlos, "Using cyber threat intelligence to support adversary understanding applied to the Russia-Ukraine conflict," *arXiv preprint arXiv:2205.03469*, 2022.
- [17] N. Kostyuk and E. Gartzke, "Why Cyber Dogs Have Yet to Bark Loudly in Russia's Invasion of Ukraine (Summer 2022)," *Texas National Security Review*, 2022.
- [18] Cybersecurity and Infrastructure Security Agency: uscert. "Update: Destructive Malware Targeting Organizations in Ukraine." Department of Homeland Security. https://www.cisa.gov/uscert/ncas/alerts/aa22-057a (accessed February, 2023).
- [19] N. Biasini et al. "Ukraine Campaign Delivers Defacement and Wipers, in Continued Escalation." Cisco Talos. https://blog.talosintelligence.com/ukraine-campaign-deliversdefacement/ (accessed February, 2023).
- [20] S. Caltagirone, A. Pendergast, and C. Betz, "The diamond model of intrusion analysis," Center For Cyber Intelligence Analysis and Threat Research Hanover Md, 2013.
- [21] A. Tanner, A. Hinchliffe, and D. Santos. "Threat assessment: Blackcat ransomware." Palo Alto. https://unit42.paloaltonetworks.com/blackcatransomware/ (accessed January, 2023).
- [22] D. Ucci, L. Aniello, and R. Baldoni, "Survey of machine learning techniques for malware analysis," *Computers & Security*, vol. 81, pp. 123-147, 2019.
- [23] M. Egele, T. Scholte, E. Kirda, and C. Kruegel, "A Survey on Automated Dynamic Malware Analysis Techniques and Tools ACM Computing Surveys," 2012.
- [24] R. Sihwail, K. Omar, and K. Z. Ariffin, "A survey on malware analysis techniques: Static, dynamic, hybrid and memory analysis," *Int. J. Adv. Sci. Eng. Inf. Technol*, vol. 8, no. 4-2, pp. 1662-1671, 2018.
- [25] E. Gandotra, D. Bansal, and S. Sofat, "Malware analysis and classification: A survey," *Journal of Information Security*, vol. 2014, 2014.
- [26] P. Shijo and A. Salim, "Integrated static and dynamic analysis for malware detection," *Procedia Computer Science*, vol. 46, pp. 804-811,

2015.

- [27] O. Or-Meir, N. Nissim, Y. Elovici, and L. Rokach, "Dynamic malware analysis in the modern era—A state of the art survey," ACM Computing Surveys (CSUR), vol. 52, no. 5, pp. 1-48, 2019.
- [28] M. Ijaz, M. H. Durad, and M. Ismail, "Static and dynamic malware analysis using machine learning," in 2019 16th International bhurban conference on applied sciences and technology (IBCAST), 2019: IEEE, pp. 687-691.
- [29] R. Sihwail, K. Omar, K. A. Zainol Ariffin, and S. Al Afghani, "Malware detection approach based on artifacts in memory image and dynamic analysis," *Applied Sciences*, vol. 9, no. 18, p. 3680, 2019.
- [30] M. I. Sharif, V. Yegneswaran, H. Saidi, P. A. Porras, and W. Lee, "Eureka: A Framework for Enabling Static Malware Analysis," in *ESORICS*, 2008, vol. 8: Springer, pp. 481-500.
- [31] A. O. Almashhadani, M. Kaiiali, S. Sezer, and P. O'Kane, "A multiclassifier network-based crypto ransomware detection system: A case study of locky ransomware," *IEEE access*, vol. 7, pp. 47053-47067, 2019.
- [32] A. Ren, C. Liang, I. Hyug, S. Broh, and N. Jhanjhi, "A three-level

ransomware detection and prevention mechanism," EAI Endorsed Transactions on Energy Web, vol. 7, no. 26, 2020.

- [33] A. Sulaiman, K. Ramamoorthy, S. Mukkamala, and A. H. Sung, "Disassembled code analyzer for malware (DCAM)," in *IRI-2005 IEEE International Conference on Information Reuse and Integration, Conf,* 2005., 2005: IEEE, pp. 398-403.
- [34] C. Q. Nguyen and J. E. Goldman, "Malware analysis reverse engineering (MARE) methodology & malware defense (MD) timeline," in 2010 Information Security Curriculum Development Conference, 2010, pp. 8-14.
- [35] D. Vidyarthi, S. Choudhary, S. Rakshit, and C. S. Kumar, "Malware detection by static checking and dynamic analysis of executables," *International Journal of Information Security and Privacy (IJISP)*, vol. 11, no. 3, pp. 29-41, 2017.
- [36] J. Bermejo Higuera, C. Abad Aramburu, J.-R. Bermejo Higuera, M. A. Sicilia Urban, and J. A. Sicilia Montalvo, "Systematic approach to malware analysis (SAMA)," *Applied Sciences*, vol. 10, no. 4, p. 1360, 2020