Online-Journals.org

iJIM International Journal of Interactive Mobile Technologies

iJIM | eISSN: 1865-7923 | Vol. 19 No. 10 (2025) | OPEN ACCESS

https://doi.org/10.3991/ijim.v19i10.49245

PAPER

Deciphering Ransomware: Strategic API Usage and Behavioral Patterns for Advanced Detection Techniques

Khalid Zirari[1](✉), Hamza Kamal Idrissi[1], Ahmed El-Yahyaoui[2], Hicham Bensaid[1], Abdeslam En-Nouaary[1]

[1]Institut National des Postes et Télécommunications, Rabat, Morocco

[2]Mohammed V University in Rabat, Rabat, Morocco

zirari.khalid@doctorant.inpt.ac.ma

ABSTRACT

Ransomware has emerged as a critical cybersecurity threat, inflicting severe financial and operational damage across industries. Traditional signature-based detection systems struggle to detect zero-day and evolving ransomware strains, as they rely on known signatures that cannot capture new tactics. In contrast, behavioral detection methods analyze ransomware actions and patterns, making them more effective. Integrating artificial intelligence (AI) can further improve detection rates; however, effective AI models require diverse, up-to-date, and representative data. Previous research has often focused on isolated aspects of ransomware behavior. Our study addresses these gaps by providing a publicly accessible, up-to-date dataset covering multiple ransomware variants and families, including polymorphic and obfuscated strains not comprehensively explored in prior literature. Additionally, our approach identifies extensive ransomware metrics, including network interactions, registry modifications, file system changes, and low-level API call patterns, enabling real-time detection of malicious activities. Through a comprehensive behavior-based analysis of over 200 recent ransomware samples using the Cuckoo Sandbox platform and custom Python scripts, our study provides cybersecurity practitioners with valuable data and actionable insights, supporting faster responses, improved threat detection, and a proactive stance against evolving risks.

KEYWORDS

ransomware detection, behavioral analysis, dataset, AI-based detection, API call patterns, cuckoo sandbox, cybersecurity

1 INTRODUCTION

The integration of digital technologies across industries has significantly enhanced operational efficiency, transforming business processes and driving productivity gains. However, rapid digitalization has also exposed organizations to a wide array of cybersecurity challenges, including data breaches, distributed denial-of-service (DDoS) attacks, insider threats, and phishing [1]. Among these, ransomware has

Zirari, K., Idrissi, H.K., El-Yahyaoui, A., Bensaid, H., En-Nouaary, A. (2025). Deciphering Ransomware: Strategic API Usage and Behavioral Patterns for Advanced Detection Techniques. *International Journal of Interactive Mobile Technologies (iJIM)*, 19(10), pp. 199–221. https://doi.org/10.3991/ijim.v19i10.49245

Article submitted 2024-06-05. Revision uploaded 2025-02-25. Final acceptance 2025-02-25.

emerged as one of the most severe and widespread threats. Ransomware, a type of malicious software that encrypts an organization's data and demands a ransom for its release, is now recognized as a major global cybersecurity threat [2]. By 2021, two-thirds of organizations worldwide had experienced ransomware attacks, resulting in substantial financial losses, with ransom demands ranging from tens of thousands to millions of dollars [3]. Beyond financial damage, ransomware also disrupts business operations, degrades productivity, and can lead to long-term reputational harm [4]. The rise of Ransomware-as-a-Service (RaaS) platforms has further exacerbated the problem by reducing the technical barriers to launching ransomware attacks, thereby increasing both the complexity and frequency of such threats. RaaS operates as a subscription-based model, enabling cybercriminals to provide ready-to-deploy ransomware kits to others for a fee or a share of the profits [5]. This model allows even individuals with limited technical skills to execute sophisticated attacks, flooding the threat landscape with diverse ransomware variants that are challenging to detect and prevent.

Existing ransomware detection methods predominantly rely on two main techniques: static analysis and dynamic analysis [6]. Static analysis, which examines malware code without execution, is limited by attackers' obfuscation techniques that conceal malicious code. Traditional antivirus systems, primarily relying on signature-based detection, have struggled to keep up with the rapidly evolving ransomware landscape. While effective against known malware, these systems often fail to detect zero-day attacks and new ransomware strains that evade detection due to the absence of known signatures. Dynamic analysis, which observes malware behavior in a controlled environment, offers better insights but is resource-intensive and vulnerable to evasion tactics, where ransomware behaves benignly to avoid detection if it detects sandbox environments. Given these challenges, there is a pressing need for more adaptive, behavior-based detection systems capable of identifying new ransomware variants in real time. Behavior-based detection is favored over static signatures as it improves detection accuracy, reduces false positives, and enables effective identification of unknown ransomware strains. Integrating artificial intelligence (AI) can further enhance detection rates.

AI models require diverse, up-to-date, and representative datasets for effective training. Unfortunately, many existing datasets are either not publicly available, outdated, or unfiltered, limiting their utility for comprehensive analysis and model training. Additionally, previous research has provided valuable insights into ransomware behaviors, but these studies frequently focus on specific patterns, failing to capture the complete range of tactics or the critical metrics triggered by different ransomware types. Furthermore, high false-positive rates hinder practical application, as legitimate applications can exhibit behaviors similar to ransomware. Many behavioral analyses overlook critical indicators, such as network traffic interaction, memory-based activities, and sophisticated evasion techniques employed by modern ransomware.

To address these limitations, our study provides a publicly accessible, up-to-date dataset covering a broad spectrum of ransomware variants, including underrepresented and obfuscated strains often excluded in previous literature. An in-depth analysis of this dataset has allowed us to identify extensive ransomware metrics—including network interactions, registry modifications, file system changes, and low-level API call patterns—to enable real-time detection of malicious activities. This study employs a novel behavior-based detection approach by conducting a comprehensive analysis of over 200 recent ransomware variants through the Cuckoo Sandbox platform and custom Python scripts, capturing critical execution

traces such as network interactions, registry modifications, file system changes, and process activities. By focusing on fundamental ransomware behaviors rather than predefined signatures, our approach allows for real-time or near real-time detection of zero-day and unknown ransomware strains. This study not only fills a gap in the existing literature by providing a systematic exploration of ransomware behavior but also supports the cybersecurity community in developing adaptive, proactive defenses against the evolving threat of ransomware.

The remainder of this paper is organized as follows: Section 2 provides an overview of ransomware, including its history and analysis methods. Section 3 reviews related research, identifying critical gaps and our unique contributions. Section 4 describes our methodology and experimental setup. Section 5 presents a detailed analysis of ransomware behavior, focusing on execution patterns, network interactions, and encryption mechanisms, followed by a discussion of our findings and their implications. Finally, Section 6 concludes the paper, summarizing key insights and proposing directions for future ransomware research and mitigation strategies.

2 BACKGROUND

2.1 Ransomware overview

Ransomware, a specific type of malicious software designed to restrict access to computer systems until a ransom is paid, has been a prominent threat since its emergence. The earliest known instance dates back to 1989 with the creation of the PC Cyborg Trojan by Joseph Popp, which specifically targeted healthcare organizations [7]. The lifecycle of ransomware, as illustrated in Figure 1, begins with the deployment of malicious code and culminates in a financial extortion attempt. This malware takes control of the victim's files and systems through various vectors, including malicious email attachments, drive-by downloads, and compromised software updates. Key initial steps include generating a unique computer ID for the infected system, disabling backup mechanisms such as shadow copies, and acquiring the victim's public IP address. The ransomware then establishes a connection to its Command and Control (C&C) server to retrieve the encryption keys required for its operations. Subsequently, it scans for important files, identifies them by their extensions, encrypts them, and finally displays or drops a ransom note, demanding payment from the victim [8], [9].

Fig. 1. Ransomware lifecycle

2.2 Businesses victimized

In 2022, the Internet Crime Complaint Center (IC3) of the United States documented 870 ransomware attacks on various organizations, revealing key trends and sector vulnerabilities. As illustrated in Figure 2, the healthcare sector was most affected, with 210 complaints, accounting for 24% of all incidents, highlighting its

critical exposure to cyber threats. This was closely followed by the manufacturing sector, which constituted 18% of the attacks. Government entities also faced significant threats, with 115 complaints making up 13% of the total incidents. A common vulnerability across these sectors was the use of outdated and unpatched software systems, making them prime targets for ransomware attacks. These findings underline the crucial need for robust cybersecurity measures and regular system updates to defend against such cyber threats [10].

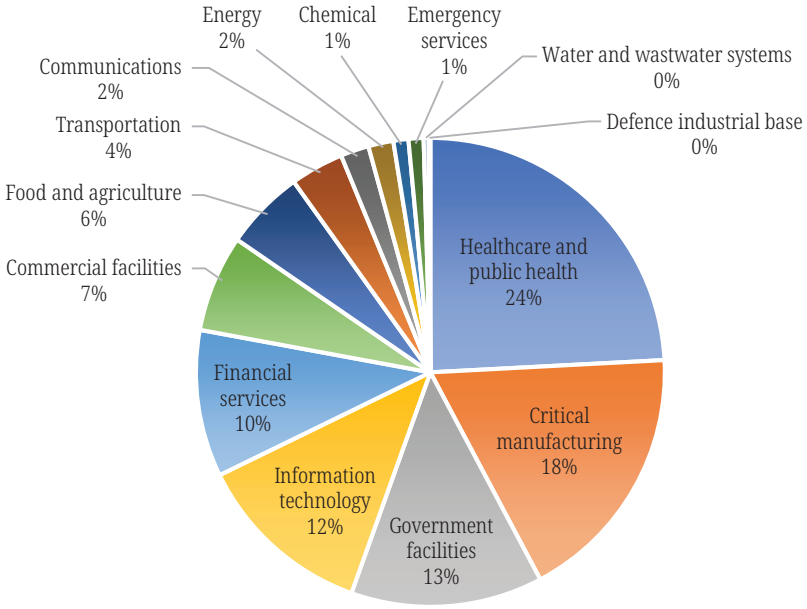

Fig. 2. Analysis of the sectors affected by ransomware

2.3 Ransomware analysis

The ongoing battle against ransomware has spurred the development of various analytical strategies, primarily focused on static and dynamic analysis techniques. These methodologies are designed to dissect the structure and behavior of ransomware during its infection and operational phases [11]. Despite these advancements, effectively defending against new and emerging ransomware variants remains a formidable challenge, exacerbated by the limited information available on newly identified strains [12]. Generally, these analysis approaches can be categorized into two primary types:

- **Static analysis:** This technique examines the ransomware code without executing it, enabling researchers to reveal its structure, dependencies (such as Dynamic Link Libraries or DLLs), and embedded URLs. These elements provide insights into the ransomware's potential behavior and communication strategies [13]. Although vital for building initial defenses, static analysis often struggles to penetrate the code obfuscation techniques that attackers employ to conceal their malware's true intentions.
- **Dynamic analysis:** This method involves executing ransomware within a controlled, isolated environment to monitor its behavior in real-time. Dynamic analysis is crucial for understanding how ransomware interacts with systems and networks, allowing for the identification of attack patterns, especially with new

or previously unknown variants [14]. While this approach demands substantial resources, it forms the foundation for developing robust detection and response mechanisms. The integration of AI can further enhance dynamic analysis by automating the detection of subtle behavioral patterns and improving accuracy in identifying malicious activity.

3 RELATED WORK

Ransomware research encompasses both proactive and reactive approaches, with researchers actively working to enhance our understanding and defenses against this evolving threat. Proactive ransomware research involves leveraging domain knowledge to minimize the damage inflicted by ransomware attacks. Researchers delve into various focal areas, including detection techniques, prevention strategies, blocking mechanisms, and the automation of analysis processes. By adopting a proactive stance, researchers aim to stay at the forefront of ransomware advancements, devising innovative approaches to effectively mitigate the impact of these malicious attacks [15].

The current body of literature on ransomware reveals a variety of detection and analysis techniques, largely centered on static and dynamic analysis [10]. Static analysis often involves examining ransomware without execution, identifying structural patterns and potential behaviors [16]. However, this method may fall short against sophisticated obfuscation techniques used by attackers [17]. Dynamic analysis, by contrast, involves executing ransomware in a controlled environment to observe its behavior in real-time, which, while resource-intensive, provides more detailed insights into attack patterns [18].

Recent studies have explored advanced static and dynamic analysis methods. For instance, research has highlighted the use of Windows API calls to detect behavioral differences between benign and malicious operations [19]. Another study utilized transfer learning with convolutional neural networks to classify malware based on API call patterns extracted during sandbox execution [20]. These approaches underscore the progression towards more sophisticated malware identification techniques that can potentially overcome the limitations of traditional signature-based systems.

Despite the advancements detailed in existing literature, several research gaps remain. Current methodologies still struggle to cope with the rapid evolution of ransomware variants, especially those employing new encryption methods and evasion techniques [21]. There is also a noticeable deficiency in real-world applicability of the proposed solutions, as many studies do not extend beyond theoretical or controlled environments [22].

Moreover, while recent advancements have improved detection rates, they often require extensive computational resources and are not scalable for real-time application across diverse systems. There is also a significant challenge in reducing false positives, where benign applications are mistakenly flagged as ransomware, leading to potential disruptions in normal operations.

This study aims to bridge these gaps by introducing an innovative, behavior-based detection framework that leverages deep learning and extensive behavioral analysis to improve the detection of ransomware across different systems and environments. By incorporating a novel dataset of ransomware samples from 2018 to 2023, including obfuscated variants, our approach enhances the adaptability and accuracy of ransomware detection systems. Furthermore, our study contributes to the field by providing:

- **Comprehensive, modern dataset:** The research provides a diverse and up-to-date dataset of ransomware samples, addressing the scarcity of modern data in the field and enabling more effective future research.
- **Accurate metric extraction for enhanced ransomware detection:** This study introduces a refined approach to extracting precise behavioral metrics through a novel analysis of ransomware behavior. By closely examining strategic API calls and detailed behavioral patterns, our method goes beyond traditional detection techniques, offering both academic and practical insights that can be directly applied to strengthen cybersecurity measures and improve ransomware detection accuracy.

4 METHODOLOGY AND EXPERIMENTAL SETUP

4.1 Experimental setup

In this study, we employed the widely recognized, open-source malware analysis platform, Cuckoo Sandbox, to investigate the behavioral characteristics of ransomware in a controlled and secure environment. Cuckoo Sandbox is celebrated for its modular and sophisticated design, allowing for detailed examination and manipulation of malicious binaries and comprehensive behavioral observation [23]. Its modular structure also supports the integration of additional tools, such as YARA, which we applied to further enhance our analysis by identifying and classifying ransomware patterns based on behavioral and structural characteristics effectively. The system is adept at monitoring critical API calls, including NtCreateFile, CryptEncrypt, and WriteProcessMemory, thereby enabling precise detection of ransomware activities, such as file encryption, process injection, and system manipulation. Moreover, the platform offers extensive network traffic analysis capabilities, capturing crucial data such as DNS queries, contacted domains, and potential command-and-control (C2) communications.

To conduct secure and efficient ransomware analysis, we configured a virtual machine with 14 GB of RAM, an 8-core CPU, and 200 GB of storage, running Ubuntu 16.04.6 LTS for its stability and security. The environment was managed using VirtualBox and fortified with several protective measures: a robust firewall restricted all external communications, network isolation prevented malware from spreading beyond the VM, and shared resources between the host and guest systems were disabled. Continuous antivirus monitoring safeguarded the host system, while VirtualBox's snapshot feature enabled us to restore the virtual machine to a clean state after each analysis, eliminating any residual threats. This highly secure and flexible setup, integrated with Cuckoo Sandbox, was precisely configured to meet our study objectives, ensuring accurate and reliable observation of ransomware behaviors.

4.2 Methodology

Our methodology for this study was designed to comprehensively analyze the behaviors and tactics of ransomware through a structured, multi-phase process. As illustrated in Figure 3, the workflow encompasses the entire research pipeline, from the initial collection of ransomware samples to advanced behavioral analysis

and correlation. This detailed flowchart provides a visual representation of how each phase seamlessly integrates, ensuring both efficiency and depth in our study.

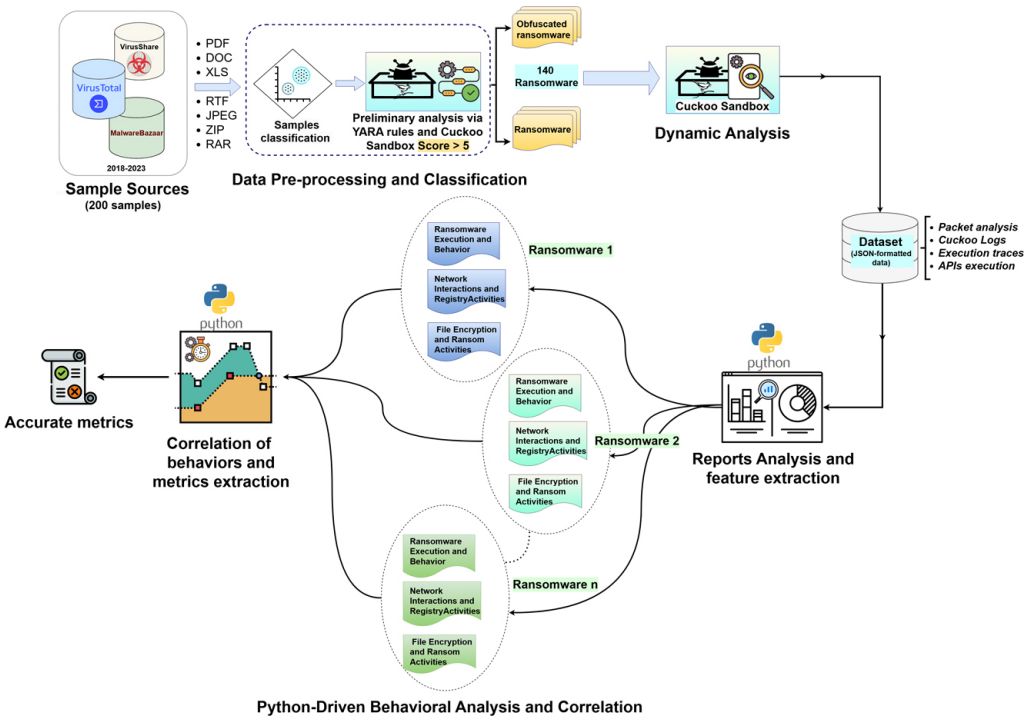

Fig. 3. Flowchart of ransomware detection and behavioral analysis process

Each component of this methodology was carefully chosen and optimized to address the challenges in analyzing modern ransomware, ensuring that the results are both comprehensive and relevant to the current cybersecurity landscape. The following subsections provide a detailed explanation of each phase, with Figure 3 serving as a guide through this rigorous analytical framework.

Ransomware sample collection. The starting point of this study was the collection of over 200 ransomware samples from well-established repositories such as VirusTotal, VirusShare, and MalwareBazaar, spanning the years 2018 to 2023. These repositories were selected for their reliability and diversity, providing access to both highly malicious and diverse ransomware samples. We collected a broad array of samples representing benign files, ransomware samples with various active and non-active variants, as well as obfuscated ransomware. The selection process was driven by the need to focus on the most impactful and prevalent ransomware variants, which have significantly influenced the cybersecurity landscape in recent years. This ensures that the dataset captures the evolution of ransomware tactics and covers a wide spectrum of threats, from simpler strains to more complex ones using advanced techniques.

The samples were collected in various file formats commonly used in ransomware distribution, such as PDF, DOC, XLS, and ZIP. These file types are frequently exploited by ransomware for concealment and propagation during the infection process. This diversity of file formats ensures that our analysis is comprehensive, covering the different methods ransomware employs to infiltrate and spread across systems. By targeting multiple infection vectors, this study aims to provide a thorough understanding of modern ransomware behaviors and their strategies.

Data preprocessing and classification. The data preprocessing phase was essential in refining our ransomware collection to ensure that only samples with the most relevant and informative behaviors were retained for deeper analysis. This initial stage was a focused filtering process, concentrating solely on identifying ransomware samples that would contribute meaningfully to the study.

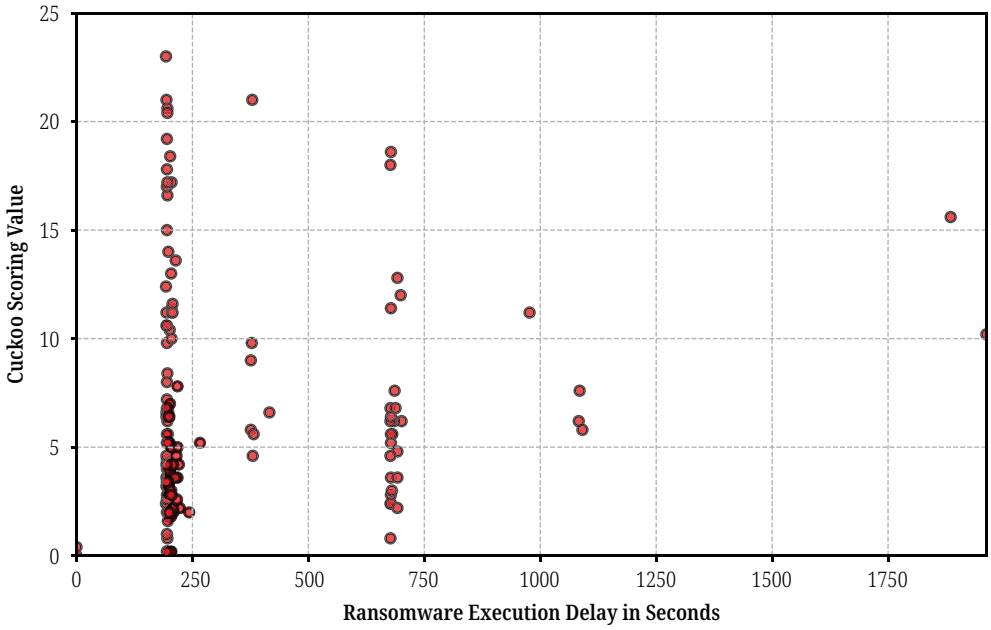

Fig. 4. Cuckoo scoring value of ransomware samples by their execution delay

Each sample was assigned a score in Cuckoo Sandbox based on its observed behavior within a controlled environment. Only samples with a Cuckoo score of 5 or higher out of 25 were selected for further analysis, as this threshold was set to indicate notable malicious behavior, such as encryption activities or privilege escalation attempts (see Figure 4). Although Figure 4 also displays the execution delay for each sample, this measure does not influence the scoring of ransomware samples. The threshold score of 5 was chosen after reviewing Cuckoo-generated JSON log files; samples scoring below this level produced logs that were either empty or negligible in size, reflecting minimal malicious behavior and indicating a lack of relevance for our analysis. Consequently, samples with lower scores were excluded due to their limited malicious behavior.

This rigorous filtering and classification process ensured that our sample collection consisted of high-impact ransomware samples and enabled us to:

- **Remove duplicate samples,** ensuring only unique ransomware variants were included to prevent redundancy.
- **Eliminate non-functional samples** that showed no detectable malicious behavior, as they provided no insights into ransomware tactics.
- **Exclude corrupted or incomplete samples,** which might compromise the accuracy and reliability of our findings.
- **Filter out benign or unrelated samples:** Cuckoo Sandbox, in combination with YARA rules, allowed us to exclude benign software and unrelated malware types, thus maintaining our focus strictly on ransomware-specific behaviors.

By prioritizing ransomware samples with active and harmful behaviors, we optimized the relevance of our dataset and ultimately selected 140 ransomware samples,

including obfuscated variants. This selection enables a more precise analysis of ransomware tactics and provides a better assessment of their potential impact on system security.

Dynamic analysis using Cuckoo Sandbox. Following the initial preprocessing phase, the selected ransomware samples underwent dynamic analysis using Cuckoo Sandbox. This platform provided a controlled environment in which ransomware samples were safely executed, enabling the observation and capture of real-time behaviors. The analysis focused on key ransomware interactions with the system, including:

- Execution traces: Tracking how ransomware manipulates system resources and attempts to escalate privileges.
- Packet analysis: Monitoring network traffic to identify communication with C2 servers.
- Cuckoo logs: Capturing system activity logs, including file system modifications, registry changes, and API calls during ransomware execution.

The analysis generated over 8GB of JSON-formatted data, including detailed logs, execution traces, and API call information, providing significant insights into post-infection ransomware behaviors [24]. This dataset offers a deeper understanding of how ransomware interacts with compromised systems and is one of the most recent and publicly accessible resources in the field. It addresses a critical gap where many existing datasets are outdated, restricted, or not publicly available [24].

A key contribution of this study is the creation of a comprehensive, up-to-date dataset that is available to researchers worldwide. By providing this publicly accessible resource, the study helps overcome limitations posed by existing datasets and enhances the flexibility and effectiveness of future ransomware behavior research.

Python-based behavioral analysis and correlation. To efficiently manage and analyze the extensive dataset generated through dynamic analysis, we developed a set of specialized Python scripts to automate feature extraction and filter key metrics [24]. By making these scripts publicly accessible, we aim to encourage replication and foster collaborative efforts within the cybersecurity community. These scripts were designed with two primary objectives:

1. **Automating feature extraction:** The scripts systematically extract essential ransomware behaviors, such as file encryption, network anomalies, and malicious API calls, ensuring that critical actions are accurately captured for analysis.
2. **Filtering and correlating key metrics across ransomware variants:** Following feature extraction, the scripts perform a comprehensive correlation analysis to compare behavioral patterns across various ransomware variants. Each variant is evaluated across several key behavioral dimensions, including file encryption activities, registry modifications, network interactions, and API execution patterns. This process helps to isolate the most relevant metrics and accurately identify recurring behaviors, providing a refined understanding of tactics used by different ransomware families.

This approach allowed us to identify and focus on the most accurate and significant metrics, a key contribution of our study. These metrics reveal common tactics and behaviors, offering deeper insights into the evolution of ransomware and supporting the development of more effective detection and mitigation strategies.

5 BEHAVIORAL ANALYSIS AND FINDING

5.1 Extracted key metrics for accurate ransomware detection

Following an extensive correlation analysis using custom Python scripts, we identified several key metrics that consistently characterize ransomware behavior across various samples [24]. Each metric reflects a unique aspect of ransomware operations, encompassing network interactions, file modifications, and registry alterations. Identifying these key metrics is crucial for accurate detection, as it enables security professionals to focus on the most indicative behaviors of ransomware activity.

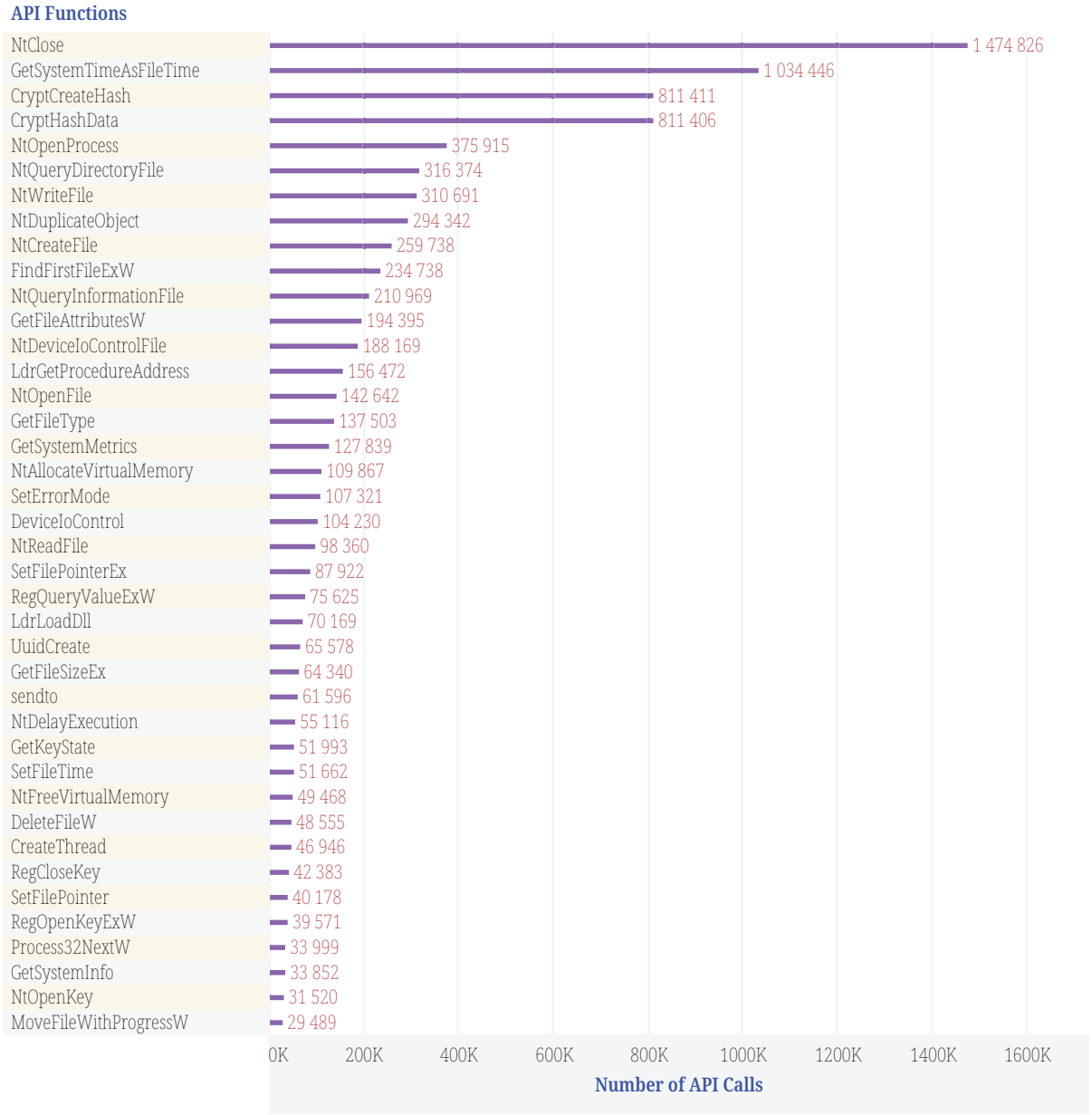

Fig. 5. Count of API calls across all ransomware samples

Figure 5 illustrates the number of calls made by each API function after correlating the activities of all the samples studied in our study. This graph highlights the most widely used APIs across different ransomware families, indicating the functions that are most frequently invoked during ransomware execution. The high frequency of these API calls underscores the extensive system interactions undertaken by ransomware and reflects the diverse techniques employed by different ransomware variants. The variability in API call counts among the samples is particularly noteworthy, suggesting that while each ransomware has its own specific behaviors, there are common patterns in how they interact with the system.

By identifying the most frequently used APIs, security professionals can focus on monitoring these calls to detect ransomware activities early. This information is essential for developing heuristic-based detection techniques that complement signature-based approaches. The APIs shown in the graph represent the most commonly used functions across ransomware variants; however, Table 1 offers a comprehensive categorization of additional API functions and specific metrics that are critical in understanding ransomware operations.

Table 1. Categorization of API functions by ransomware behavioral activities

Activity Type	Category	API Functions
Ransomware Execution and Behavior	**Process and Thread Manipulation**	AdjustTokenPrivileges, CloseWindow, CreateProcessA, CreateProcessInternalW, CreateProcessW, CreateRemoteThreadEx, CreateThread, CreateToolhelp32Snapshot, DuplicateTokenEx, ExitProcess, GetCurrentProcess, GetProcessWindowStation, GetWindowThreadProcessId, Module32NextW, NtCreateThreadEx, NtDuplicateObject, NtGetContextThread, NtOpenProcess, NtOpenThread, NtResumeThread, NtSuspendThread, NtTerminateProcess, OpenProcess, OpenProcessToken, PostQuitMessage, Process32NextW, ReadProcessMemory, Thread32Next, WriteProcessMemory.
	Anti-Analysis and Evasion Techniques	IsDebuggerPresent, LoadLibraryExA, NtDelayExecution, OutputDebugStringA, SetErrorMode, SetUnhandledExceptionFilter.
	System Interaction and Information Collection	EnumWindows, GetActiveWindow, GetAdaptersAddresses, GetCaretPos, GetClipboardOwner, GetClipboardViewer, GetComputerNameA, GetComputerNameW, GetCursorPos, GetDC, GetDesktopWindow, GetDeviceCaps, GetFileVersionInfoSizeW, GetFileVersionInfoW, GetFocus, GetForegroundWindow, GetInputState, GetKeyState, GetKeyboardState, GetMessageTime, GetModuleFileNameA, GetNativeSystemInfo, GetOpenClipboardWindow, GetShellWindow, GetSystemDirectoryA, GetSystemDirectoryW, GetSystemInfo, GetSystemMetrics, GetSystemTimeAsFileTime, GetSystemWindowsDirectoryW, GetTimeZoneInformation, GetUserNameExW, GetUserNameW, GlobalMemoryStatusEx, IWbemServices_ExecQuery, LookupAccountSidW, ObtainUserAgentString, SHBrowseForFolderA, SHChangeNotify, SHFileOperationA, SHGetFileInfoA, SHGetFolderPathW, SHGetPathFromIDListA, SHGetSpecialFolderPathA, SystemParametersInfoW.
	Privilege Escalation and Security Token Manipulation	AdjustTokenPrivileges, GetTokenInformation, LookupPrivilegeValueA, SetTokenInformation.
Network Interactions and Registry Activities	**Network Communications and Command-and-Control (C2) Interactions**	bind, Closesocket, getaddrinfo, HttpOpenRequestA, HttpSendRequestA, HttpSendRequestW, InternetCloseHandle, InternetConnectA, InternetReadFile, InternetSetOptionA, ioctlsocket, select, sendto, setsockopt, shutdown, socket, WSASocketW, WSAStartup, WinHttpCloseHandle, WinHttpConnect, WinHttpOpen, WinHttpOpenRequest, WinHttpReceiveResponse, WinHttpSendRequest.
	Registry Manipulation and Configuration Changes	NtCreateKey, NtEnumerateValueKey, NtOpenKey, NtOpenKeyEx, NtQueryKey, NtQueryMultipleValueKey, NtQueryValueKey, NtSetValueKey, RegCloseKey, RegCreateKeyExA, RegCreateKeyExW, RegDeleteKeyW, RegDeleteValueW, RegEnumKeyA, RegEnumKeyExA, RegEnumKeyExW, RegEnumKeyW, RegEnumValueA, RegEnumValueW, RegOpenKeyExA, RegOpenKeyExW, RegQueryInfoKeyW, RegQueryValueExA, RegQueryValueExW, RegSetValueExA, RegSetValueExW.

(Continued)

Table 1. Categorization of API functions by ransomware behavioral activities *(Continued)*

Activity Type	Category	API Functions
File Encryption and Ransom Activities	**File System Access and Manipulation**	CloseHandle, CompareFileTime, CopyFileW, CreateDirectoryA, CreateDirectoryW, CreateFileA, CreateFileW, DeleteFileA, DeleteFileW, DeleteObject, FindFirstFileA, FindFirstFileExA, FindFirstFileW, FindNextFileA, FindNextFileExA, FindNextFileExW, FindNextFileW, GetFileAttributesA, GetFileAttributesExW, GetFileAttributesW, GetFileInformationByHandle, GetFileInformationByHandleEx, GetFileSize, GetFileSizeEx, GetFileType, GetFileVersionInfoSizeW, GetFileVersionInfoW, GetModuleFileNameA, GetShortPathNameA, GetShortPathNameW, GetTempFileNameA, GetTempPathW, MoveFileA, MoveFileExW, MoveFileW, MoveFileWithProgressW, NtCreateFile, NtOpenFile, NtQueryAttributesFile, NtQueryDirectoryFile, NtQueryInformationFile, NtReadFile, NtSetInformationFile, NtWriteFile, ReadFile, RemoveDirectoryW, SearchPathA, SearchPathW, SetEndOfFile, SetFileAttributesA, SetFileAttributesW, SetFilePointer, SetFilePointerEx, SetFileSecurityA, SetFileTime, WriteFile, WritePrivateProfileStringA.
	Cryptographic Operations	CryptAcquireContextA, CryptAcquireContextW, CryptCreateHash, CryptDecodeObjectEx, CryptDecrypt, CryptDestroyKey, CryptEncrypt, CryptExportKey, CryptGenRandom, CryptHashData, CryptImportKey, CryptReleaseContext, CryptSetKeyParam, UuidCreate.
	Ransom Note Deployment and User Notification	SystemParametersInfoW, SHChangeNotify, WriteFile, CreateFileA, CreateFileW.

Our study categorizes these API functions according to three core activities of ransomware:

- **Ransomware execution mechanisms and behavioral analysis:** This section delves into how ransomware manipulates processes and threads, employs anti-analysis and evasion techniques, and interacts with the system to collect information. By understanding these behaviors, we can identify key indicators of ransomware execution and develop strategies to mitigate their impact.
- **Network communications and registry manipulation by ransomware:** Here, we examine how ransomware establishes C&C communications and alters system configurations through registry changes. Our analysis revealed that ransomware often manipulates proxy settings and employs malicious proxies to facilitate communication with C&C servers, aiding in data exfiltration and encryption key retrieval. Additionally, ransomware frequently modifies registry keys related to 'Internet Settings' and 'Restart Manager,' indicating tactics for persistence and evasion, such as disabling security features and altering network settings to evade detection and maintain control, as illustrated in Figure 6. These registry alterations are critical for ransomware's ability to modify network configurations and ensure sustained operation, underscoring the significance of monitoring unauthorized registry changes for effective detection and prevention.
- **File encryption mechanisms and ransom deployment strategies:** This section examines how ransomware gains access to and manipulates the file system, executes cryptographic operations, and ultimately delivers ransom notes to the victim. Key findings highlight the use of advanced cryptographic functions for generating and exporting encryption keys, with high entropy values indicating obfuscation techniques within binary sections to evade detection. Identifying the specific APIs involved in these encryption processes is crucial for recognizing the hallmark encryption activities of ransomware attacks.

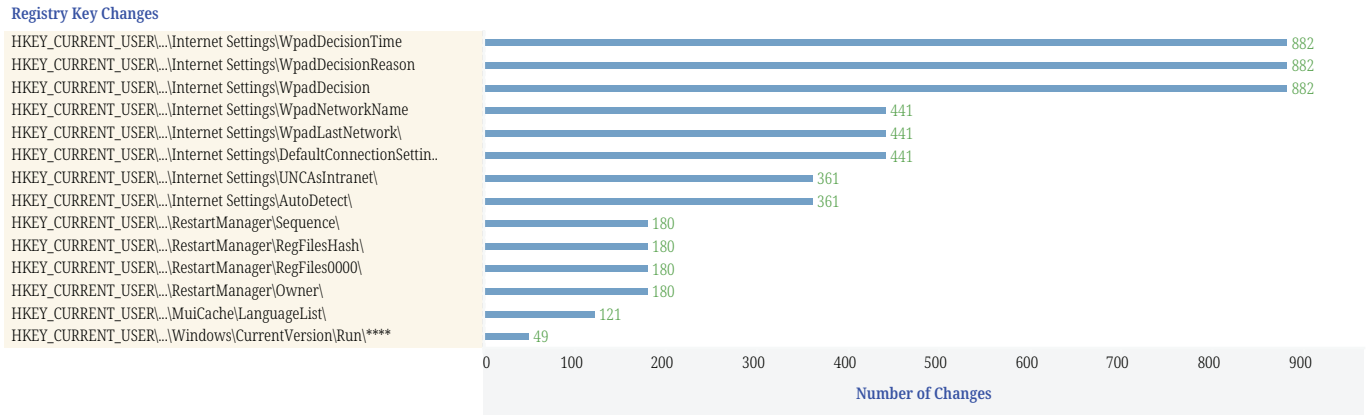

Fig. 6. Count of registry key changes across all ransomware samples

Our study contributes to bridging the gap in traditional detection methods by pinpointing the most critical APIs across different ransomware samples. While each ransomware variant exhibits its own specific behaviors and techniques, there are commonalities in their operational patterns. By focusing on these key metrics, we enhance heuristic-based detection techniques that complement signature-based approaches. Monitoring the most frequently used and significant API calls enables earlier detection of ransomware activities, ultimately providing more accurate detection and mitigation strategies that can adapt to the evolving landscape of ransomware threats.

5.2 Ransomware trace analysis: key metric exploitation

Ransomware execution and behavior traces

- **Trace 1: Disabling security features**
  - HKEY_LOCAL_MACHINE\SOFTWARE\MICROSOFT\Windows\CurrentVersion\Policies\System\EnableLUA
  - cmdline netsh advfirewall set currentprofile state off
  - cmdline netsh firewall set opmode mode=disable

The ransomware attempts to disable essential security features, such as user access control (UAC) and Windows Firewall, to achieve persistence, elevate privileges, and bypass security mechanisms.

- **Trace 2: Keylogging installation**
  - SetWindowsHookExW
  - thread_identifier: 0
  - callback_function: 0x00000000ffeda89c
  - hook_identifier: 13 (WH_KEYBOARD_LL)
  - module_address: 0x00000000ffe30000

By installing a keylogger, the ransomware records keystrokes to capture sensitive information such as passwords and credit card details, which can be used for exploitation or as part of the encryption process.

- **Trace 3: Manipulating boot configuration**
  - "c:\windows\system32\cmd.exe" /c bcdedit /set {current} bootstatuspolicy ignoreallfailures

- "c:\windows\system32\cmd.exe" /c bcdedit /set {current} recoveryenabled no
- cmdline C:\Windows\System32\cmd.exe /C REG ADD "HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Run" /v "svchos" /t REG_SZ /d "C:\Users\Administrator\AppData\Local\Temp\40b865d1c3ab1b8544bcf57c88edd30679870d40b27d62feb237a19f0c5f9cd1*.exe" /f /reg:64*

The ransomware manipulates boot configuration settings to evade detection during system startup and disables the recovery mechanism. It also creates an autostart registry entry to ensure continued execution upon user logon.

- **Trace 4: Disabling critical Windows services and shadow copies**
  - cmdline "C:\Windows\System32\net.exe" stop "samss" /y
  - cmdline "C:\Windows\System32\net.exe" stop "audioendpointbuilder" /y
  - cmdline WMIC.exe shadowcopy delet
  - cmdline cmd /c "WMIC.exe shadowcopy delet"
  - cmdline net stop "audioendpointbuilder" /y
  - cmdline "C:\Windows\System32\schtasks.exe" /CREATE /TN "N0mFUQoa" /TR "C:\Users\Administrator\AppData\Roaming\Rj3fNWF3.exe" /SC ONLOGON /RL HIGHEST /F
  - cmdline schtasks /CREATE /TN "N0mFUQoa" /TR "C:\Users\Administrator\AppData\Roaming\Rj3fNWF3.exe" /SC ONLOGON /RL HIGHEST /F
  - LookupPrivilegeValueW
  - privilege_name: SeBackupPrivilege

The ransomware stops critical Windows services and deletes shadow copies to prevent data recovery. It also creates a high-privilege scheduled task to ensure persistence and attempts to gain specific privileges to enhance its control over system resources.

Network interactions and registry activities

- **Trace 1: Deleting proxy and intranet settings**
  - RegDeleteValueW
    regkey_r: ProxyBypass
    regkey:HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\ProxyBypass
  - RegDeleteValueW
    regkey_r: IntranetName
    regkey:HKEY_CURRENT_USER|Software|Microsoft|Windows|CurrentVersion|Internet Settings|ZoneMap|IntranetName

These registry modifications are typical behavior exhibited by ransomware to manipulate system configurations and evade security mechanisms. By deleting the "ProxyBypass" value, the ransomware may attempt to bypass proxy settings that could potentially block its communication with malicious servers or hinder its ability to establish C2 connections. Additionally, the removal of the "IntranetName" value might indicate an effort to obscure its presence or evade detection by altering network zone mappings related to intranet access.

- **Trace 2: Manipulating WPAD settings**
  - RegSetValueExA

regkey_r: WpadDecisionReason
regkey:HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Wpad\{300B6E7D-99E6-48B2-A45D-5CE456541676}\WpadDecisionReason

By manipulating web proxy auto-discovery (WPAD) settings, the ransomware redirects network traffic through a malicious proxy, enabling data interception and manipulation.

- **Trace 3: C&C server communication**

Fig. 7. Ransomware communication with C&C server: Key exchange

The infected machine communicates with a C&C server at "zexeq.com," as illustrated in Figure 7, sending GET requests and receiving encryption keys in response. This communication is crucial for the ransomware's operations, as it allows the malware to receive instructions and encryption keys.

- **Trace 4: HTTP communication**
  - HttpSendRequestW
    headers: Content-Type: application/x-www-form-urlencoded
    Host: whyers.io request_handle: 0x00cc000c
    post_data:user=panda&TargetID=B2D8FAFDCC7631230A2E65E3&System Information=Windows%207%20Enterprise%20x64,%20FR,%20196.119.167.214,%20DEVELOPMENT&max_size_of_file=0.0&size_of_hdd=9
  - dead_host 192.168.56.10:49226
  - dead_host 192.168.56.1:139

The ransomware communicates with the C&C server, sending system information and receiving instructions.

- **Trace 5: Recurring GET requests for payload delivery**

  The provided HTTP traffic shows a recurring pattern of the infected machine making GET requests to download the file "download.zip" from the C&C server with the IP address "116.202.6.47." As depicted in Figures 8 and 9, the source machine uses a web browser User-Agent header, and the server responds with an HTTP/1.1 200 OK status, indicating successful requests. The "download.zip" file has a size of 698,036 bytes, and it is of type "application/zip." The repetitive nature of these requests, combined with the use of a zip archive format, raises suspicions of potentially malicious activity. The ransomware may use the zip archive to deliver its payload to the infected machine. The zip file could contain executable files, scripts, or other components that carry out the malicious activities on the victim's system.

Fig. 8. Ransomware communication with C&C server: GET requests for 'download.zip'

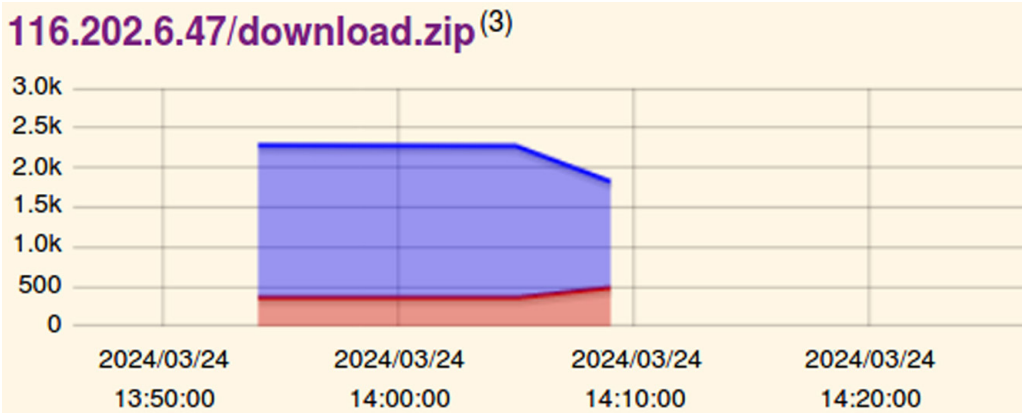

Fig. 9. Ransomware communication with (C&C) server: Payload delivery

- **Trace 6: Creating ransomware specific registry keys**
  - RegCreateKeyExW
    regkey_r: Software\WanaCrypt0r base_handle: 0x80000002
    regkey: HKEY_LOCAL_MACHINE\Software\WanaCrypt0r
  - RegSetValueExA
    reg_type: 1 (REG_SZ) value: C:\Users\Administrator\AppData\Local\Temp

regkey:HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\WanaCrypt0r\wd

The ransomware creates specific registry keys to maintain persistence and store configuration data.

- **Trace 7: Manipulating cryptography providers**
  - RegOpenKeyExW
    regkey_r:Software\Microsoft\Cryptography\Providers\Trust\Certificate\{00AAC56B-CD44-11D0-8CC2-00C04FC295EE}
    base_handle: 0x80000002 key_handle: 0x00000154
    regkey:HKEY_LOCAL_MACHINE\Software\Microsoft\Cryptography\Providers\Trust\Certificate\{00AAC56B-CD44-11D0-8CC2-00C04FC295EE}
  - RegQueryValueExW
    key_handle: 0x00000154 regkey_r: $DLL
    reg_type: 1 (REG_SZ) value: WINTRUST.DLL
    regkey:HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Microsoft\Cryptography\Providers\Trust\Certificate\{00AAC56B-CD44-11D0-8CC2-00C04FC295EE}\$DLL

The ransomware manipulates cryptography-related registry entries to replace or inject malicious DLLs, potentially compromising security components. This information suggests that the ransomware is attempting to replace or inject a malicious DLL file, potentially compromising SSL or TLS communication and intercepting sensitive data.

File encryption and ransom activities

- **Trace 1: Cryptographic key generation and export**
  - CryptGenKey
    crypto_handle: 0x0074cc40
    algorithm_identifier: 0x0000a400 (CALG_RSA_KEYX)
    flags: 134217729 key: provider_handle: 0x00748e58
  - CryptExportKey
    buffer:¤RSA1·…xìE°¢;šÙUhÄŒl¾õ,F/ü}bGÎ:ë â…–[cwšfü;kn6›¥å,Ã#îxw">ƒÞvÁÈ‹ÑvÞÙS¬6Ý…BO5»Ù~M§Ä%¦ÞkûV.ôš<e□Ò'&r9□ÂÍZ0□□,ì□õÛƒ¡°@|X¸ÚÆ(r¹ì®üÐÿJåÛÑÂðÍj4kR-B#gÂ“Æ×ÊU`H_Õ¤ÞÀ†Â7ØæÂýuS”‡;Tð«p“ÓbéÍq[[D¯Þ} □□Dˆ×”†r/6(□dfGÆñôÛèb+¤Ó³“H‰ä‹ï¥°ôÓ>†³`ìÇóÉ¹Ç•T-NÖ¿
    crypto_handle: 0x0074cc40 flags: 0
    crypto_export_handle: 0x00000000 blob_type: 6
  - LdrGetProcedureAddress
    function_address: 0x7627e5ee
    function_name: CryptGenRandom
    module: ADVAPI32 module_address: 0x76270000
  - CryptAcquireContextW
    crypto_handle: 0x00000000
    provider_type: 1 flags: 0
    provider: Microsoft Enhanced Cryptographic Provider v1.0

The ransomware generates cryptographic keys using the RSA algorithm and exports these keys for encryption tasks. This sophisticated behavior indicates a high level of technical capability aimed at securing communications and file encryption.

- **Trace 2: High entropy in binary section**

  section {u'size_of_data': u'0x00004600', u'virtual_address': u'0x0000c000', u'entropy': 7.213436333949118, u'name': u'.rdata', u'virtual_size': u'0x00004488'} entropy 7.21343633395

  The trace indicates the presence of a packer, a technique used to compress or encrypt ransomware code to obfuscate its malicious intent and evade detection. High entropy values in these sections suggest that the data within is encrypted or compressed, hindering analysis.

- **Trace 3: File manipulation and encryption**
  - **NtCreateFile**

    create_disposition: 5 (FILE_OVERWRITE_IF)
    file_handle: 0x00000330
    filepath: C:\Users\Administrator\Desktop\KH-1\5143-Article Text-10703-1-10-20220415.pdf
    desired_access:0x40100080(FILE_READ_ATTRIBUTES|SYNCHRONIZE|GENERIC_WRIT)
    filepath_r: \??\C:\Users\Administrator\Desktop\KH-1\5143-Article Text-10703-1-10-20220415.pdf...
    create_options:96(FILE_NON_DIRECTORY_FILE|FILE_SYNCHRONOUS_IO_NONALERT)
    status_info: 2 (FILE_CREATED)
    share_access: 1 (FILE_SHARE_READ)
  - **NtWriteFile**

    buffer: -Þ□ZoH7}¥ãºÉ]7MXu©1ûŸ]ùñ=□Fù‰É® C`ÉÒóß÷ß□7óú8À Üß‹e´ÖÌˆÅO’ãÔhƒÊ%˜ ¾º□Z
    filepath: C:\Users\Administrator\Desktop\KH-1\5143-Article Text-10703-1-10-20220415.pdf
  - **MoveFileWithProgressW**

    newfilepath_r: \\?\C:\Users\Administrator\Desktop\KH-1\5143-Article Text-10703-1-10-20220415.pdf.sage
    oldfilepath_r: \\?\C:\Users\Administrator\Desktop\KH-1\5143-Article Text-10703-1-10-20220415.pdf...
    newfilepath: C:\Users\Administrator\Desktop\KH-1\5143-Article Text-10703-1-10-20220415.pdf.sage
    oldfilepath: C:\Users\Administrator\Desktop\KH-1\5143-Article Text-10703-1-10-20220415.pdf
  - **DeleteFileW**

    filepath_r: \\?\C:\Users\Administrator\Desktop\KH-1\5143-Article Text-10703-1-10-20220415.pdf
    filepath: C:\Users\Administrator\Desktop\KH-1\5143-Article Text-10703-1-10-20220415.pdf

The ransomware encrypts files by creating new files with encrypted content, renaming the original files with a new extension, and then deleting the original files. This sequence of events showcases the ransomware's capability to obfuscate its actions and evade detection. The use of low-level Windows APIs such as NtCreateFile and NtWriteFile indicates sophisticated behavior by the ransomware to evade traditional security measures.

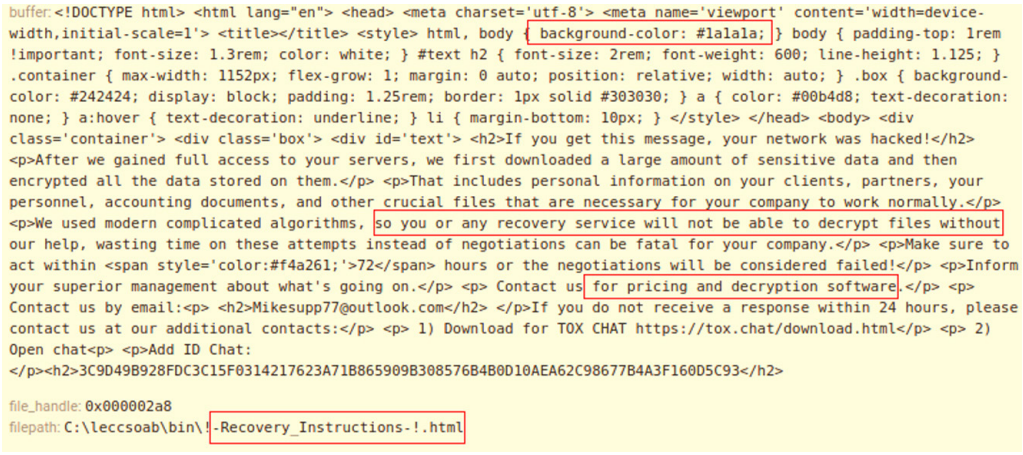

Fig. 10. Ransom note and the setup of the desktop background

- **Trace 4: Ransom note and desktop modification**
  - NtWriteFile
    buffer: ATTENTION! YOUR NETWORK HAS BEEN BREACHED AND ALL DATA WAS ENCRYPTED. PLEASE CONTACT US AT: https://bastad5huzwkepdixedg-2gekg7jk22ato24zyllp6lnjx7wdtyctgvyd.onion/ filepath: C:\Program Files\instructions_read_me.txt
  - NtWriteFile
    buffer: background: #8 offset: 0
    filepath: C:\Users\Public\Videos\Sample Videos\how_to_back_files.html

The ransomware finalizes its actions by modifying the desktop wallpaper as shown in Figure 10 and leaving a ransom note that directs victims to a website for payment and decryption instructions. The note highlights the severity of the attack and the demand for Bitcoin payment to restore access to encrypted files.

5.3 Comparison and discussion

This study introduces a robust behavior-based detection framework that leverages the Cuckoo Sandbox and custom scripts to monitor a broad range of ransomware activities, including API calls, network interactions, and registry modifications. Current detection frameworks, while effective in specific metrics, generally lack the comprehensive behavioral insights necessary for identifying advanced ransomware strains. For instance, some models achieve high accuracy but are limited to API call analysis, overlooking critical indicators such as registry changes and network anomalies that are essential for detecting zero-day threats. Other approaches rely on static defenses, such as monitoring file encryption or CPU usage, which are inadequate in capturing the adaptive and evasive behaviors characteristic of modern ransomware and do not support real-time detection capabilities.

A comparative summary of relevant existing frameworks is presented in Table 2 below, illustrating the key contributions, methods, dataset details, and limitations of each approach.

Table 2. Comparative overview of ransomware detection frameworks

Research Paper	Key Contributions	Metrics & Detection Method	Dataset Details	Limitations
Nguyen & Lee [16]	High accuracy and low false positives using LightGBM on API sequences	API call sequences only (LightGBM model on API sequences)	Not public; 1,803 ransomware, 4,008 benign samples	Limited to API sequences; lacks registry, network anomalies, and sandbox integration
RansomWall (Shaukat et al.) [22]	Layered defense with data backup, static and dynamic analysis	Monitors file encryption, network changes, backups (Layered defense)	Not public; 574 samples from 12 ransomware families	No public dataset; lacks registry and network anomaly monitoring
Arabo et al. [6]	CPU and memory monitoring as early indicators of ransomware	CPU, memory, API usage (Process-based monitoring)	Not public; 7 ransomware, 34 malware, 41 benign	Dataset not public; lacks network and registry monitoring; no real-time requirements
Rosli et al. [18]	Host-based approach using Process Monitor for tracking ransomware actions on host	Observes file and network events only (Host-based monitoring)	Local dataset, not public; 10 ransomware strains	Limited to file and network; lacks API and registry tracking; lacks comprehensive detection
Lin & Lee [17]	Uses decoy files to detect ransomware and trigger shutdown	Decoy file encryption activity only (Decoy-triggered shutdown)	Custom dataset, not public	Relies solely on decoy files; lacks comprehensive behavior tracking (e.g., registry, network)

As shown, a significant limitation across current studies is the absence of a publicly accessible and updated dataset, which restricts reproducibility and prevents community researchers from benchmarking new frameworks against established models.

Our proposed framework addresses these gaps by enabling real-time detection across a comprehensive set of ransomware behaviors and providing an open-access dataset of over 200 ransomware samples from 2018 to 2023. This dataset enhances reproducibility and fosters community-driven research in behavior-based ransomware detection. By focusing on core behavioral patterns instead of static indicators, our framework significantly improves detection accuracy, reduces false positives, and enhances adaptability to newly emerging ransomware strains. This approach positions our framework as a substantial advancement in ransomware defense, particularly in its proactive ability to identify and mitigate threats through real-time, behavior-focused analysis.

6 CONCLUSION

This study presents a thorough exploration of ransomware behaviors and strategies, addressing the substantial cybersecurity challenges posed by this evolving threat. A key contribution of this study is the development of a refined metric extraction approach that enables precise and nuanced analysis of ransomware behavior. By focusing on critical API calls, network interactions, and registry modifications, our method captures core ransomware actions, facilitating deeper behavioral insights that enhance both detection accuracy and real-world applicability. This approach empowers cybersecurity practitioners with actionable information on ransomware tactics, moving beyond traditional signature-based methods to address zero-day and evolving ransomware strains. Additionally, the study contributes a comprehensive, modern dataset of over 200 recent ransomware samples, bridging a significant gap in accessible, up-to-date data and supporting future research. This dataset enables the testing and refinement of advanced detection algorithms on

contemporary ransomware variants while also fostering collaborative efforts within the cybersecurity community toward developing more adaptive, behavior-based detection systems.

Looking forward, integrating AI and machine learning into ransomware detection frameworks will be crucial for proactively identifying and countering ransomware threats. Future studies should leverage the novel metrics extraction techniques introduced here to improve real-time threat intelligence and adaptive response capabilities. Collaboration across sectors will be essential for exchanging knowledge and developing robust, adaptive defenses that meet the challenges of an evolving ransomware landscape. These initiatives promise to enhance not only ransomware countermeasures but also foster resilience and innovation in cybersecurity practices overall.

7 REFERENCES

[1] S. Anawar, N. F. Othman, S. R. Selamat, Z. Ayop, N. Harum, and F. A. Rahim, "Security and privacy challenges of big data adoption: A qualitative study in telecommunication industry," *International Journal of Interactive Mobile Technologies (iJIM)*, vol. 16, no. 19, pp. 81–97, 2022. https://doi.org/10.3991/ijim.v16i19.32093

[2] A. H. M. Alaidi, R. M. Al_Airaji, H. Th. S. Alrikabi, I. A. Aljazaery, and S. H. Abbood, "Dark web illegal activities crawling and classifying using data mining techniques," *International Journal of Interactive Mobile Technologies (iJIM)*, vol. 16, no. 10, pp. 122–139, 2022. https://doi.org/10.3991/ijim.v16i10.30209

[3] Fortinet, "Two-thirds of organizations targeted by a ransomware attack, according to Fortinet ransomware survey," *Fortinet Newsroom*, 2021. [Online]. Available: https://www.fortinet.com/corporate/about-us/newsroom/press-releases/2021/two-thirds-organizations-target-ransomware-attack-fortinet-ransomware-survey [Accessed: Feb. 20, 2024].

[4] N. I. Mustapha, Y. Vaicondam, N. A. Jahanzeb, N. B. A. Usmanovich, and S. H. B. Yusof, "Cybersecurity challenges and solutions in the fintech mobile app ecosystem," *International Journal of Interactive Mobile Technologies (iJIM)*, vol. 17, no. 22, pp. 100–116, 2023. https://doi.org/10.3991/ijim.v17i22.45261

[5] P. H. Meland, Y. F. F. Bayoumy, and G. Sindre, "The ransomware-as-a-service economy within the darknet," *Computers & Security*, vol. 92, p. 101762, 2020. https://doi.org/10.1016/j.cose.2020.101762

[6] A. Arabo, R. Dijoux, T. Poulain, and G. Chevalier, "Detecting ransomware using process behavior analysis," *Procedia Computer Science*, vol. 168, pp. 289–296, 2020. https://doi.org/10.1016/j.procs.2020.02.249

[7] M. Ryan, "Ransomware revolution: The rise of a prodigious cyber threat," *Advances in Information Security*, vol. 85, Switzerland, Charm: Springer, 2021, pp. 1–15. https://doi.org/10.1007/978-3-030-66583-8_1

[8] E. Berrueta, D. Morato, E. Magana, and M. Izal, "A survey on detection techniques for cryptographic ransomware," *IEEE Access*, vol. 7, pp. 144925–144944, 2019. https://doi.org/10.1109/ACCESS.2019.2945839

[9] K. Zirari, H. K. Idrissi, A. El-Yahyaoui, H. Bensaid, and A. En-Nouaary, "Enhancing ransomware detection: A registry analysis-based approach," in *10th International Conference on Future Internet of Things and Cloud (FiCloud)*, 2023, pp. 65–70. https://doi.org/10.1109/FiCloud58648.2023.00018

[10] Federal Bureau of Investigation (FBI), "Internet crime report 2022," *Internet Crime Complaint Center (IC3)*, 2022. [Online]. Available: https://www.ic3.gov/Media/PDF/AnnualReport/2022_IC3Report.pdf [Accessed: Feb. 28, 2024].

[11] C. Young, R. McArdle, N.-A. Le-Khac, and K.-K. R. Choo, "Forensic investigation of ransomware activities—Part 1," in *Studies in Big Data*, 2020, pp. 51–77. https://doi.org/10.1007/978-3-030-47131-6_4

[12] C. Boyton, N.-A. Le-Khac, K.-K. R. Choo, and A. Jurcut, "Forensic investigation of ransomware activities—Part 2," in *Cyber and Digital Forensic Investigations. Studies in Big Data*, vol. 74, N. A. Le-Khac and K. K. Choo, Eds., Switzerland, Charm: Springer, 2020, pp. 79–108. https://doi.org/10.1007/978-3-030-47131-6_5

[13] S. S. Chakkaravarthy, D. Sangeetha, and V. Vaidehi, "A survey on malware analysis and mitigation techniques," *Computer Science Review*, vol. 32, pp. 1–23, 2019. https://doi.org/10.1016/j.cosrev.2019.01.002

[14] E. Ahmed, A. A. Sorrour, M. A. Sobh, and A. M. Bahaa-Eldin, "A cloud-based malware detection framework," *International Journal of Interactive Mobile Technologies (iJIM)*, vol. 11, no. 2, pp. 113–127, 2017. https://doi.org/10.3991/ijim.v11i2.6577

[15] J. Pont, O. A. Oun, C. Brierley, B. Arief, and J. Hernandez-Castro, "A roadmap for improving the impact of anti-ransomware research," in *Secure IT Systems. NordSec 2019,* in Lecture Notes in Computer Science, A. Askarov, R. Hansen, and W. Rafnsson, Eds., vol. 11875, Switzerland, Charm: Springer, 2019, pp. 137–154. https://doi.org/10.1007/978-3-030-35055-0_9

[16] D. T. Nguyen and S. Lee, "LightGBM-based ransomware detection using API call sequences," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 10, 2021. https://doi.org/10.14569/IJACSA.2021.0121016

[17] Y.-S. Lin and C.-F. Lee, "Ransomware detection and prevention through strategically hidden decoy file," *International Journal of Network Security*, vol. 25, no. 2, pp. 212–220, 2023. [Online]. Available: http://ijns.jalaxy.com.tw/contents/ijns-v25-n2/ijns-2023-v25-n2-p212-220.pdf

[18] M. S. Rosli, R. S. Abdullah, W. Yassin, and F. M. A. Hussin, "Discovering ransomware behavior by host-based approach," *Journal of Theoretical and Applied Information Technology*, vol. 97, no. 14, pp. 3848–3854, 2019. [Online]. Available: https://www.jatit.org/volumes/Vol97No14/6Vol97No14.pdf

[19] N. Hampton, Z. Baig, and S. Zeadally, "Ransomware behavioral analysis on windows platforms," *Journal of Information Security and Applications*, vol. 40, pp. 44–51, 2018. https://doi.org/10.1016/j.jisa.2018.02.008

[20] M. Goyal and R. Kumar, "AVMCT: API calls visualization based malware classification using transfer learning," *Journal of Algebraic Statistics*, vol. 13, no. 1, pp. 31–41, 2022. [Online]. Available: https://www.publishoa.com/index.php/journal/article/view/59/57

[21] P. O'Kane, S. Sezer, and K. McLaughlin, "Obfuscation: The hidden malware," *IEEE Security & Privacy*, vol. 9, no. 5, pp. 41–47, 2011. https://doi.org/10.1109/MSP.2011.98

[22] S. K. Shaukat and V. J. Ribeiro, "RansomWall: A layered defense system against cryptographic ransomware attacks using machine learning," in *10th International Conference on Communication Systems & Networks (COMSNETS)*, 2018, pp. 356–363. https://doi.org/10.1109/COMSNETS.2018.8328219

[23] Cuckoo Sandbox, Automated Malware Analysis, 2024. [Online]. Available: https://github.com/cuckoosandbox [Accessed: Mar. 22, 2024].

[24] K. H. Zirari, "Ransomware behavior," *GitHub Repository*, 2024. [Online]. Available: https://github.com/KH-ZIRARI/Ransomware_Behavior/ [Accessed: Jun. 05, 2024].

8 AUTHORS

Khalid Zirari holds an engineering degree in IT infrastructure security from the National School of Applied Sciences (ENSA) Oujda, Morocco, obtained in 2017.

Since 2021, he has been pursuing Ph.D. at the National Institute of Posts and Telecommunications (INPT). His research focuses on ransomware analysis and the development of advanced threat mitigation strategies, with additional interests in cybersecurity and cloud security. He can be contacted at zirari.khalid@doctorant.inpt.ac.ma.

Hamza Kamal Idrissi holds an engineering degree in software engineering from the National School for Computer Science (ENSIAS), Rabat, Morocco. He earned his Ph.D. in computer security from Mohammed V University in Rabat. In 2020, he joined the Mathematics, Computing, and Networks Department at the National Institute of Posts and Telecommunications (INPT), Rabat, Morocco, as an Assistant Professor in cybersecurity. His research interests include intrusion detection systems (IDS), cloud security, and cryptography. He can be contacted at kamalidrissi@inpt.ac.ma.

Ahmed El-Yahyaoui holds an engineering degree in software engineering from the National Institute of Posts and Telecommunications (INPT), Rabat, Morocco. He earned his Ph.D. in computer security from the National School for Computer Science (ENSIAS), Morocco. He is currently an Associate Professor at the Faculty of Sciences, Mohammed V University in Rabat. His research interests include applied cryptography, encryption, and cloud security. He can be contacted at a.elyahyaoui@um5r.ac.ma.

Hicham Bensaid received an engineering degree in computer engineering from ENSIMAG, Grenoble, France, in 2001 and earned a Ph.D. in computer science from Grenoble Alpes University, France, in 2011. He is currently an Associate Professor in the Mathematics, Computing, and Networks Department at the National Institute of Posts and Telecommunications (INPT). His research interests include automated reasoning and intelligent systems. He can be contacted at bensaid@inpt.ac.ma.

Abdeslam En-Nouaary received an engineering degree in computer engineering from the National School for Computer Science (ENSIAS), Rabat, Morocco, in 1996. He obtained his master's and Ph.D. degrees in computer science from the University of Montreal in 1998 and 2001, respectively. He has been a Professor at Concordia University, Montreal, and is currently a full professor in the Mathematics, Computing, and Networks Department at the National Institute of Posts and Telecommunications (INPT), Rabat, Morocco. His research interests include software engineering, protocol engineering, and real-time systems. He can be contacted at abdeslam@inpt.ac.ma.