The Indonesian Journal of Computer Science
www.ijcs.net
Volume 14, Issue 2, April 2025
https://doi.org/10.33022/ijcs.v14i2.4839

Static and Dynamic Analysis of Ransomware: Insight from Babuk and Lockbit 3.0

Kukuh Iskandar Rizqi[1], Kurnia Wahyu[2], Dana Indra Sensuse[3], Sofian Lusa[4], Prasetyo Adi Wibowo Putro[5], Sofiyanti Indrisari[6]
kukuh.iskandar@ui.ac.id[1], kurnia.wahyu@ui.ac.id[2], dana@cs.ui.ac.id[3], sofiyanti.indrisari11@office.ui.ac.id[4], sofian.lusa@trisakti.ac.id[5], prasetyo.adi@poltekssn.ac.id[6]
[1,2,3,4] Faculty of Computer Science, Universitas Indonesia
[5] Faculty of Tourism, Institute of Tourism Trisakti
[6] Cryptography Engineering, National Cyber and Crypto Polythecnic

Article Information

Received : 9 Apr 2025
Revised : 17 Apr 2025
Accepted : 30 Apr 2025

Keywords

Babuk, Lockbit 3.0, Ransomware Analysis, Static Analysis, Dynamic Analysis

Abstract

Ransomware remains a significant cybersecurity threat, targeting both private and public sectors with increasing sophistication. This study analyzes Babuk and Lockbit 3.0 ransomware through static and dynamic methods to uncover their technical characteristics and runtime behaviors. Static analysis reveals differences in structural complexity, with Babuk employing a simpler architecture while Lockbit 3.0 incorporates advanced features such as additional sections and dynamic functionality. Dynamic analysis highlights distinct operational strategies, including encryption patterns and registry modifications for persistence and obfuscation. These findings provide critical insights into ransomware behavior, serving as a foundation for developing AI and ML-based detection systems to identify and mitigate evolving threats effectively.

A. Introduction

The rise in ransomware attacks over the past decade has presented an escalating threat to individuals, organizations, and critical infrastructures worldwide. Ransomware, which encrypts victim data and demands a ransom for decryption, has evolved in sophistication, with variants like Lockbit 3.0 and Babuk employing advanced evasion and encryption techniques. These developments pose significant challenges to traditional cybersecurity defenses, making it imperative to study the underlying mechanics of ransomware to develop effective detection and mitigation strategies [1][2].

One notable example of this threat is the 2024 ransomware attack on Indonesia's Temporary National Data Center (PDNS). This incident involved Lockbit 3.0 and Babuk ransomware targeting both Windows-based systems and ESXi hypervisors, disrupting essential public services [3]. Such cases highlight the urgency of understanding ransomware behavior, particularly in critical infrastructure contexts where the consequences extend beyond financial loss to societal disruption.

Despite existing advancements in ransomware analysis, several critical questions remain unanswered. This study focuses on three key issues: the structural and technical characteristics of Babuk and Lockbit 3.0 ransomware [4]; the differences in their runtime behaviors and operational strategies in controlled environments [5]; and the insights these behaviors provide for developing more effective detection and mitigation mechanisms [6]. By addressing these questions, this research seeks to fill the gaps in comparative studies of ransomware families, providing actionable insights for designing adaptive security frameworks capable of mitigating diverse ransomware threats.

The remainder of this paper is structured as follows. The next section reviews related theories and previous research on ransomware characteristics and analysis methods. Section 3 describes the methodology, including data collection, virtual machine setup, and evaluation metrics. Section 4 presents the results of static and dynamic analyses, highlighting key findings for both Babuk and Lockbit 3.0. In Section 5, the discussion focuses on the implications of these findings for cybersecurity practice and future research directions. Finally, Section 6 concludes the study, summarizing its contributions and proposing recommendations for enhancing ransomware detection and mitigation.

B. Related Theory

1. Overview od Ransomware

Ransomware is malicious software that targets users by denying access to their data or systems through encryption or system lockdown. The attackers then demand payment, typically in cryptocurrency, to restore access, exploiting the anonymity of blockchain transactions to avoid detection. This type of malware has become one of the most significant threats in cybersecurity, impacting not only individuals but also organizations and critical infrastructures. The financial and operational repercussions of ransomware attacks make them a key focus of research and defense strategies [6].

The sophistication of ransomware has evolved dramatically over time. Early variants, such as the AIDS Trojan in 1989, used simple encryption techniques and

demanded payment via postal mail. In contrast, modern ransomware employs advanced encryption algorithms like AES (Advanced Encryption Standard) and RSA (Rivest–Shamir–Adleman), making decryption without the private key virtually impossible. Additionally, many modern ransomware variants, such as Babuk and Lockbit 3.0, employ data exfiltration tactics alongside encryption, threatening victims with data leaks to increase pressure for ransom payment [7].

The advent of Ransomware-as-a-Service (RaaS) has lowered the technical barriers for launching ransomware attacks. This model enables cybercriminals to lease or purchase ransomware kits from developers, expanding the pool of attackers and increasing the frequency of attacks. The financial and operational damage caused by ransomware is immense, with organizations often facing weeks of downtime, significant recovery costs, and reputational harm. These challenges underscore the importance of understanding ransomware behavior to develop effective countermeasures [8].

2. Ransomware Analysis

The analysis of ransomware is critical for uncovering its functionality and developing detection and mitigation strategies. The two primary approaches to ransomware analysis are static analysis and dynamic analysis. Static analysis focuses on examining the ransomware's code without executing it, while dynamic analysis observes the behavior of ransomware when executed in a controlled environment. Both methods provide complementary insights, enabling a comprehensive understanding of ransomware mechanisms [9].

Static analysis involves disassembling the ransomware binary to examine its structure and logic. Tools like IDA Pro and Ghidra are used to translate machine code into human-readable assembly code, revealing the ransomware's intended functions. Additionally, metadata and string extraction techniques can identify embedded URLs, file extensions, or ransom notes. However, static analysis is often limited by the use of obfuscation techniques, which hide the ransomware's true functionality [10].

Dynamic analysis, on the other hand, involves executing ransomware in a sandboxed environment to observe its runtime behavior. This method provides insights into how ransomware interacts with files, registry keys, and network connections. For example, monitoring tools like Process Monitor and Wireshark can capture system modifications and outbound communications to command-and-control servers. While dynamic analysis is effective for revealing real-time behavior, it must be conducted in isolated environments to prevent the ransomware from causing unintended damage [7].

C. Research Method

This research adopts an empirical approach aimed at understanding the internal characteristics and behaviors of ransomware. Static analysis is used to examine the ransomware binaries without executing them, providing insights into their structure, cryptographic techniques, and embedded artifacts. Meanwhile, dynamic analysis involves executing ransomware samples in a controlled virtual environment to observe their behavior during runtime, such as file encryption, registry modifications, and network communication.

These methods address the following research questions:

a. What are the technical characteristics of Babuk and Lockbit 3.0 ransomware?

b. How do Babuk and Lockbit 3.0 behave during runtime in a controlled Windows 11 virtual environment?

This dual-analysis approach allows for a comprehensive understanding of the ransomware's functionality and potential mitigative strategies [9].

1. Data Collection

Ransomware samples for Babuk and LockBit 3.0 were obtained from MalwareBazaar, a trusted repository frequently used in malware research. Samples were selected based on their compatibility with the Windows platform. This decision was informed by the fact that Windows remains the most widely used operating system globally, with approximately 76% market share among desktop users as of 2024 [11].

Windows popularity makes it a primary target for ransomware developers, as it increases the likelihood of infections and the impact on end users. Moreover, the prevalence of ransomware targeting Windows systems highlights the need for specific countermeasures tailored to this platform. To ensure the validity and authenticity of the samples, SHA-256 hashes were used for verification, and all samples were cross-checked with VirusTotal before analysis [10].

2. Virtual Machine Setup

To conduct a secure static and dynamic analysis, a virtual machine was configured with the following specifications in Table 1.

Table 1. Virtual machine specifications

Aspect	Details
Platform	Oracle VM VirtualBox
Operating System	Windows 11
RAM	16 GB
Disk Space	50 GB
Network Configuration	Host-only
Installed Tools	PEStudio, Detect It Easy, Process Monitor, RegShot, Virus Total, HxD

3. Evaluation Metrics

Evaluation metrics in this study are designed to comprehensively assess ransomware behavior through both static and dynamic analysis. These metrics aim to uncover the structural attributes and operational behaviors of ransomware, providing a deeper understanding of its design and attack strategies. This approach aligns with methodologies described in previous research, which emphasize the importance of integrating static and dynamic evaluations to capture both inherent properties and runtime behaviors of ransomware [12] analysis metrics focus on evaluating the architectural aspects of ransomware, including its structural components, imported libraries, and potential obfuscation techniques.

These analyses provide insights into the ransomware's construction and its mechanisms for avoiding detection. Similarly, dynamic analysis metrics are used to monitor ransomware behavior during execution, capturing its interactions with critical system components, such as the file system, registry, and active processes. These metrics are crucial for identifying the techniques ransomware uses for persistence and encryption, as highlighted [13].

By cosights from static and dynamic analysis, this study adopts a holistic framework for evaluating ransomware behavior. This dual approach allows for a more complete understanding of ransomware strategies, enabling the identification of critical attack mechanisms used by ransomware families like Babuk and Lockbit 3.0. The findings align with existing studies that advocate for comprehensive approaches to ransomware analysis to inform detection and mitigation strategies [14].

D. Result and Discussion

1. Static Analysis

Static analysis aims to understand the internal structure of ransomware without executing the binary file. Using tools such as PEStudio, Detect It Easy (DIE), and VirusTotal, critical information on file size, entropy, loaded libraries, and the binary's section structure was extracted. The results of this analysis are summarized in Table 2.

Babuk ransomware has a significantly larger file size (1,184,258 bytes) compared to Lockbit 3.0 (163,328 bytes), which may indicate the presence of additional static data or code within Babuk. According to entropy analysis conducted using Detect It Easy, Babuk has an entropy score of 7.677, suggesting that it is 96% packed, while Lockbit 3.0 has an entropy score of 7.309, indicating 91% packing. These high entropy levels suggest that both ransomware families use packing techniques to obfuscate their internal structures and evade detection.

VirusTotal analysis revealed that Babuk loads libraries such as mscore.dll and kernel32.dll for basic system functions. In contrast, Lockbit 3.0 includes additional libraries like gdi32.dll and user32.dll, which are likely used for graphical and user interface operations, indicating a broader functional scope. A notable difference lies in the section structure of the binaries: Babuk includes essential sections like .text, .sdata, .rsrc, and .reloc, while Lockbit 3.0 adds the .pdata section for exception handling. This highlights Lockbit 3.0’s more complex and flexible design compared to Babuk.

Table 2. File Metadata of Babuk and Lockbit 3.0

Metrics	Babuk	Lockbit 3.0
File Size	1,184,258 bytes	163,328 bytes
SHA-256	9f7d694f35...0f0	9db515b9...cbd
Timestamp	2021-08-22 23:52:38	2022-07-06 00:49:15
Entropy	7.677 (96% Packed)	7.309 (91% Packed)
Lib Imports	mscore.dll, kernel32.dll	gdi32.dll, user32.dll, kernel32.dll
Section	.text, .sdata, .rsrc, .reloc	.text, .itext, .data, .rdata, .pdata, .reloc

2. Dynamic Analysis

Dynamic analysis provides critical insights into the runtime behavior of ransomware, allowing us to observe how Babuk and Lockbit 3.0 interact with system resources, files, and registry entries during execution. This section discusses the observed Babuk and Lockbiit 3.0 ransomware behaviors, including file modifications, registry changes, multi-threading, and shadow copy deletion.

2.1. Babuk Ransomware Behavior

The Babuk ransomware demonstrates a systematic and structured attack mechanism, starting from initialization to file encryption, system modification, and temporary file activity. Analysis through Procmon and RegShot, along with additional insights into shadow copy deletion and temporary file usage, highlights Babuk's effective exploitation of core Windows components.

Babuk begins its attack by loading critical libraries such as wow64.dll and kernel32.dll ensuring compatibility and access to system functions. The ransomware accesses key registry entries such as HKLM\SYSTEM\CurrentControlSet\Control\Session Manager to gather system configuration details while leveraging USN Journal entries to identify recently modified files. Simultaneously, Babuk adds registry keys like bam\State\UserSettings to maintain persistence and monitor system activity. This preparatory phase establishes the groundwork for the subsequent encryption process (see Table 3).

During the encryption phase, Babuk efficiently overwrites original files with encrypted versions without creating duplicates, employing multi-threading to accelerate parallel processing. Key directories, such as C:\, Recycle.Bin, and WinREAgent are targeted, and ransom notes like How To Restore Your Files.txt are generated in each affected directory. Additionally, Babuk executes the vssadmin delete shadows /all /quiet command to delete shadow copies, obstructing recovery efforts. Temporary files in C:\Windows\Temp are utilized for caching data during encryption (see Table 4 and Figure 1).

Table 3. Sample Procmon data used for Babuk Dinamic Analysis

Operation	Path	Detail
CreateFile	C:\Windows\Temp	Access: Reading attributes
Process Create	C:\Windows\System32\cmd.exe	Command: vssadmin delete shadows
WriteFile	C:\How To Restore Your Files.txt	Writing ransom note

Table 4. Sample RegShot data used for Babuk Dinamic Analysis

Registry Path	Action	Details
HKLM\SYSTEM\CurrentControlSet\Control\Session Manager	Query	Accessed for configuration data
HKLM\SYSTEM\CurrentControlSet\Services\bam\State\UserSettings	Add	Added for persistence tracking
HKLM\SOFTWARE\Microsoft\Windows Defender	Modify	Altered to evade detection

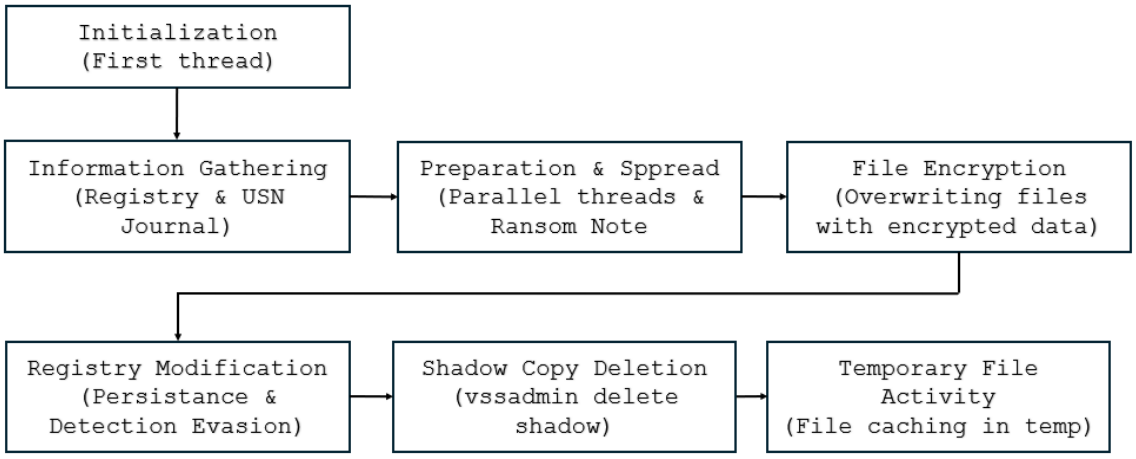

Figure 1. Flowchart of Babuk Operations

2.2. Lockbit 3.0 Ransomware Behavior

Lockbit 3.0 employs a structured and efficient attack strategy. Based on Procmon analysis (Table 5), the ransomware begins its operation by loading critical system libraries such as gdi32.dll, responsible for graphical functions, and user32.dll, used for user interface interactions. Temporary files are created in the C:\Windows\Temp directory to cache data, while the command vssadmin delete shadows is executed to delete shadow copies, preventing data recovery.

The registry plays a vital role in Lockbit's operation. As shown in the RegShot data (Table 6), significant modifications are made to registry entries, including HKLM\SYSTEM\CurrentControlSet\Services for persistence and HKLM\SOFTWARE\Microsoft\Windows Defender to disable security features. Additionally, Lockbit leverages an extra registry entry in HKLM\SOFTWARE\Policies\Microsoft\Windows\Safer to bypass application restrictions.

Finally, Lockbit employs multi-threading to accelerate file encryption across multiple directories simultaneously, as illustrated in Figure 2. The ransomware overwrites original files directly with encrypted versions and creates a README.txt file in each target directory containing ransom instructions.

Table 5. Sample Procmon data used for Lockbit 3.0 Dinamic Analysis

Operation	Path	Detail
LoadImage	C:\Windows\SysWOW64\gdi32.dll	Access: graphical functions
LoadImage	C:\Windows\SysWOW64\user32.dll	Access: user interface functions
CreateFile	C:\Windows\Temp	Temporary file for caching
WriteFile	C:\README.txt	Writing ransom note
ProcessCreate	C:\Windows\System32\cmd.exe	Deletes shadow copies

Table 6. Sample RegShot data used for Lockbit 3.0 Dinamic Analysis

Registry Path	Action	Details
HKLM\SYSTEM\CurrentControlSet\Services	Modify	Persistence through system services
HKLM\SOFTWARE\Microsoft\Windows Defender	Modify	Disables security features
HKLM\SOFTWARE\Policies\Microsoft\Windows\Safer	Add	Bypasses application restrictions

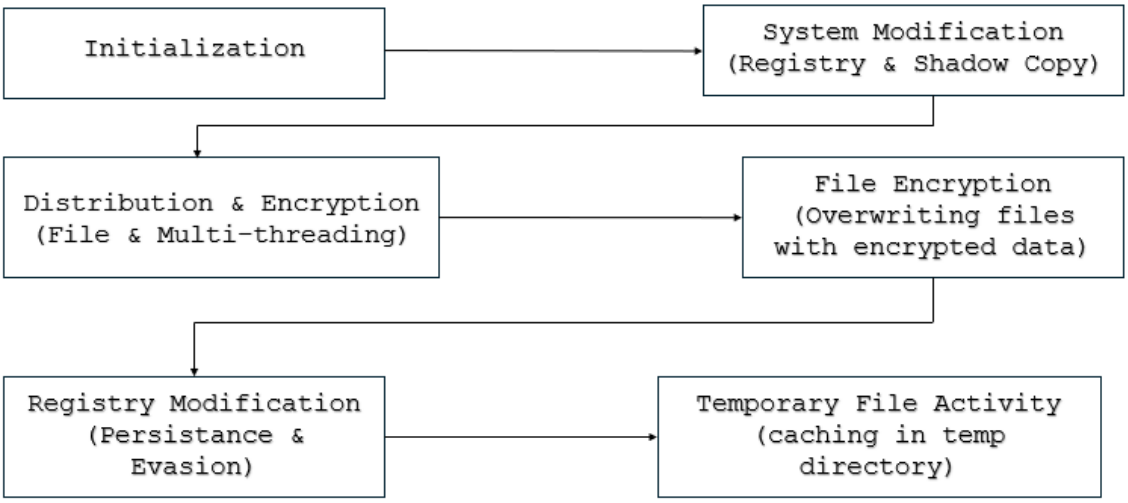

Figure 2. Flowchart of Lockbit 3.0 Operations

3. Encryption Patterns Analysis

Encryption pattern analysis was conducted on Babuk and Lockbit 3.0 ransomware using six different file sizes: 1MB, 10MB, 100MB, 1GB, 5GB, and 10GB. The results highlighted significant differences in encryption approaches adopted by the two ransomware variants.

Babuk ransomware exhibits a relatively consistent encryption pattern, encrypting files up to a certain fixed point without substantial proportional adjustments according to file size. For a 1MB file, Babuk encrypted data up to the hexadecimal address 00022BD0. For files sized 10MB, 100MB, 1GB, and 5GB, Babuk consistently encrypted up to the hexadecimal address 00100000. However, for extremely large files (10GB), Babuk increased encryption coverage significantly up to the hexadecimal address 00F00000.

In contrast, Lockbit 3.0 displayed a dynamic and adaptive encryption approach based on file size. For smaller files (1MB and 10MB), Lockbit encrypted up to hexadecimal address 00080000. Starting from the 100MB file, the encryption range expanded considerably. Lockbit 3.0 encrypted the 100MB file up to the hexadecimal address 00500000, whereas files of 1GB, 5GB, and 10GB experienced encryption extending up to hexadecimal address 00F00000. This demonstrates that Lockbit 3.0 adjusts its encryption depth progressively with increasing file size.

This analysis indicates that Lockbit 3.0 is more aggressive in encrypting larger files compared to Babuk, which employs a more conservative encryption strategy until dealing with extremely large file sizes. To visually illustrate these differences,

the encryption patterns for both ransomware variants across different file sizes are presented in Figure 3 below.

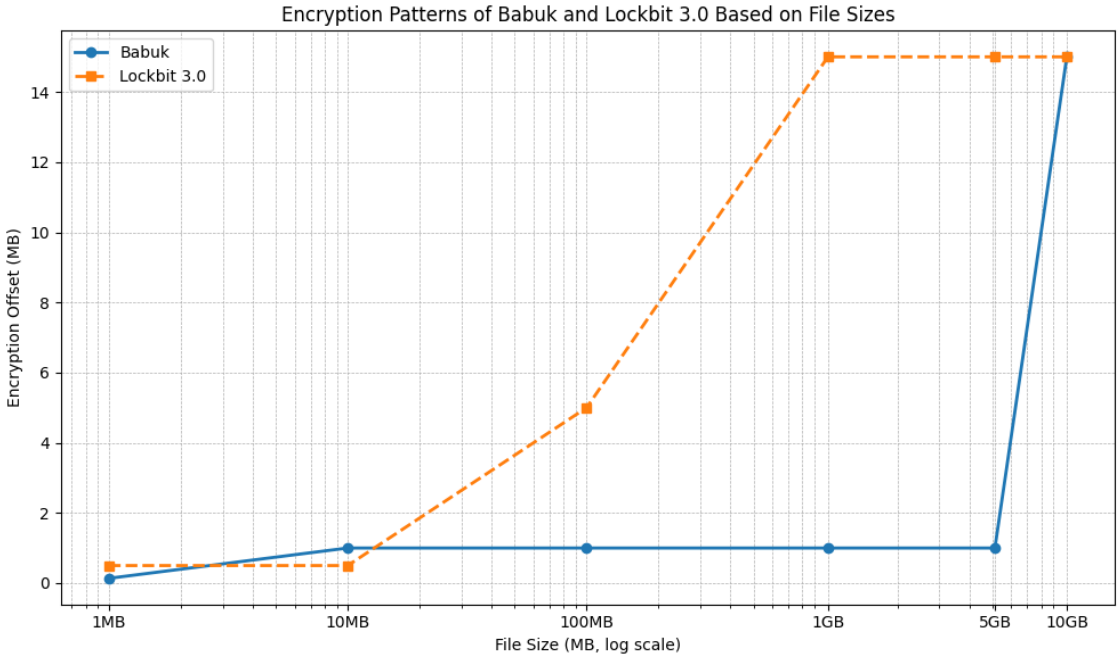

Figure 3. Encryption Patterns of Babuk and Lockbit 3.0 Based on File Sizes

4. Discussion

The comparative analysis of Babuk and Lockbit 3.0 ransomware reveals significant distinctions in their structural and operational complexity, offering critical insights into the evolution of ransomware strategies and their implications for cybersecurity defense mechanisms. These findings underscore the importance of tailoring detection systems to the unique characteristics of each ransomware variant.

Structural Complexity and Obfuscation Techniques

Babuk's relatively simpler structure, as evidenced by its limited section types (".text," ".sdata," ".rsrc," and ".reloc"), reflects a design focused on achieving operational objectives with minimal sophistication. Its high entropy score (96% packed) indicates the use of packing techniques to evade static analysis tools, but the absence of advanced sections like ".pdata" limits its flexibility. In contrast, Lockbit 3.0 incorporates additional sections such as ".pdata," supporting enhanced functionality, including exception handling. This structural complexity, combined with moderately high packing (91% entropy), suggests a balance between obfuscation and operational flexibility, presenting a more robust challenge to static analysis.

These structural differences emphasize the need for adaptive detection systems. Babuk's simplicity may allow heuristic-based methods to identify its patterns, while Lockbit 3.0's complexity necessitates more sophisticated analysis techniques, such as deep learning models capable of recognizing nuanced structural attributes.

Operational Behaviour and Threat Dynamics

Dynamic analysis highlights further distinctions in the runtime behaviours of Babuk and Lockbit 3.0. Babuk's uniform encryption approach, targeting the first 1 MB of files, prioritizes speed and resource efficiency. This design enables rapid propagation but may leave residual data vulnerable to recovery techniques. Conversely, Lockbit 3.0's dynamic encryption strategy, scaling its depth based on file size, maximizes damage while maintaining reasonable execution speed. This nuanced behavior reflects a higher level of threat sophistication, requiring behavior-based detection methods capable of tracking adaptive encryption patterns.

Registry manipulation and persistence mechanisms also differentiate these ransomware families. Babuk's reliance on core registry keys for system configuration and persistence ("HKLM\SYSTEM\CurrentControlSet\Control\Session Manager") underscores its focused attack strategy. In contrast, Lockbit 3.0's addition of advanced registry modifications ("HKLM\SOFTWARE\Policies\Microsoft\Windows\Safer") highlights its capacity to bypass application restrictions, enhancing its evasion tactics.

Implications for Detection and Mitigation

The findings reveal that traditional signature-based detection systems are insufficient for addressing the adaptive strategies employed by modern ransomware. Lockbit 3.0's advanced features, including dynamic encryption and bypassing application restrictions, exemplify the limitations of conventional defenses. Developing AI/ML-based detection mechanisms that integrate findings from static and dynamic analysis—such as entropy variations, section structures, and encryption behaviors—is critical for proactive mitigation.

Furthermore, these insights highlight the need for cross-disciplinary approaches that combine technical expertise with policy development. For example, enhanced collaboration between cybersecurity professionals and policymakers can establish guidelines for detecting and mitigating ransomware threats targeting critical infrastructure.

Future Threat Landscape

The evolution of ransomware from Babuk to Lockbit 3.0 demonstrates a trajectory toward greater sophistication and adaptability. Future variants are likely to exploit vulnerabilities in emerging technologies, such as IoT devices and personal cloud accounts, where security measures are often weaker. Proactive research into these domains, leveraging the methodologies presented in this study, can prepare defenses against next-generation threats.

E. Conclusion

This study provides a comprehensive analysis of Babuk and Lockbit 3.0 ransomware using static and dynamic methodologies, addressing the research questions about their technical characteristics and runtime behaviors. The findings demonstrate that while Babuk employs a simpler architecture optimized for speed

and efficiency, Lockbit 3.0 exhibits a higher degree of complexity and adaptability, reflected in its advanced encryption strategies and persistence mechanisms.

Through static analysis, the structural distinctions between Babuk and Lockbit 3.0 were uncovered, highlighting differences in section configurations, entropy, and library usage. Dynamic analysis further revealed their contrasting operational behaviors, with Babuk prioritizing rapid, resource-efficient attacks and Lockbit 3.0 focusing on maximizing damage through adaptive encryption and advanced registry manipulations.

These insights not only validate the research questions but also provide actionable indicators of compromise (IOCs) for enhancing ransomware detection and mitigation. The dual-method approach adopted in this study highlights the necessity of combining static and dynamic evaluations for a holistic understanding of ransomware threats

Contributions to Cybersecurity Practice and Theory

The study's findings contribute to cybersecurity practice by identifying specific characteristics that can inform the development of AI/ML-based detection systems. For example, Lockbit 3.0's encryption depth scaling and registry modification patterns offer valuable input features for machine learning algorithms. Similarly, Babuk's reliance on packed binaries and limited sections can guide heuristic and signature-based defenses.

From a theoretical perspective, this research underscores the importance of understanding ransomware's evolution in response to detection strategies. The observed advancements in Lockbit 3.0 highlight a trajectory toward more sophisticated threats, demanding equally adaptive defensive measures

Recommendations and Future Work

The results advocate for the integration of behavior-based detection systems into existing cybersecurity frameworks, leveraging the nuanced insights from this analysis. Future research should extend this study to include other ransomware families and explore vulnerabilities in emerging technologies such as IoT and cloud environments.

The growing sophistication of ransomware, as exemplified by Lockbit 3.0, underscores the urgent need for cross-disciplinary collaboration in addressing these threats. By aligning technical analysis with policy and strategy, this study provides a foundation for building resilient defenses against the evolving landscape of ransomware

F. Acknowledgment

The authors would like to express sincere gratitude to the Ministry of Communication and Digital of the Republic of Indonesia (Komdigi RI) for funding and supporting this research through the scholarship program provided. Special thanks are also addressed to Universitas Indonesia for providing resources and facilities necessary for completing this study.

G. References

[1] P. D. Kurnia, "Analisis Malware," *Anal. Malware*, 2021, [Online]. Available: http://edocs.ilkom.unsri.ac.id/1175/1/09011181320019_TUGAS_ANALISIS MALWARE.pdf

[2] M. N. Olaimat, M. Aizaini Maarof, and B. A. S. Al-Rimy, "Ransomware Anti-Analysis and Evasion Techniques: A Survey and Research Directions," *2021 3rd Int. Cyber Resil. Conf. CRC 2021*, 2021, doi: 10.1109/CRC50527.2021.9392529.

[3] F. De Gaspari, D. Hitaj, G. Pagnotta, L. De Carli, and L. V. Mancini, "Evading behavioral classifiers: a comprehensive analysis on evading ransomware detection techniques," *Neural Comput. Appl.*, vol. 34, no. 14, pp. 12077–12096, 2022, doi: 10.1007/s00521-022-07096-6.

[4] B. A. S. Al-rimy, M. A. Maarof, and S. Z. M. Shaid, "Ransomware threat success factors, taxonomy, and countermeasures: A survey and research directions," *Comput. Secur.*, vol. 74, pp. 144–166, 2018, doi: 10.1016/j.cose.2018.01.001.

[5] Palo Alto, "Ransomware and Extortion Report 2023 | 2," 2023, [Online]. Available: https://start.paloaltonetworks.com/2023-unit42-ransomware-extortion-report

[6] K. Khaliq, N. Z. Ab Rahim, K. Hamid, M. Ibrar, U. Ahmad, and M. U. Ullah, "Ransomware Attacks: Tools and Techniques for Detection," *2nd Int. Conf. Cyber Resilience, ICCR 2024*, pp. 1–5, 2024, doi: 10.1109/ICCR61006.2024.10532926.

[7] B. Fiore, K. Ha, L. Huynh, J. Falcon, R. Vendiola, and Y. Li, "Security Analysis of Ransomware: A Deep Dive into WannaCry and Locky," *2023 IEEE 13th Annu. Comput. Commun. Work. Conf. CCWC 2023*, pp. 285–294, 2023, doi: 10.1109/CCWC57344.2023.10099114.

[8] N. Scaife, H. Carter, P. Traynor, and K. R. B. Butler, "CryptoLock (and Drop It): Stopping Ransomware Attacks on User Data," *Proc. - Int. Conf. Distrib. Comput. Syst.*, vol. 2016-Augus, pp. 303–312, 2016, doi: 10.1109/ICDCS.2016.46.

[9] Michael Sikorski and Andrew Honig, *Practical Malware Analysis: The Hands-On Guide to Dissecting Malicious Software*. San Francisco, 2012.

[10] S. S. Bhat, M. Banu, G. A. Ansari, and V. Selvam, "Diabetes detection system using machine learning algorithms," *Int. J. Electron. Healthc.*, vol. 13, no. 3, pp. 231–246, 2023, doi: 10.1504/IJEH.2023.135804.

[11] StatCounter Globalstat, "Desktop Operating System Market Share Worldwide." Accessed: Jan. 03, 2025. [Online]. Available: https://gs.statcounter.com/os-market-share/desktop/worldwide/#monthly-202403-202503

[12] J. Raiyn and B. Alqarbiah, "A survey of Cyber Attack Detection Strategies," vol. 8, no. 1, pp. 247–255, 2014.

[13] T. A. Cahyanto, V. Wahanggara, and D. Ramadana, "Analisis dan Deteksi Malware Menggunakan Metode Malware Analisis Dinamis dan Malware Analisis Statis," *J. Sist. Teknol. Inf. Indones.*, vol. 2, no. 1, pp. 19–30, 2017.

[14] D. Sgandurra, L. Muñoz-González, R. Mohsen, and E. C. Lupu, "Automated Dynamic Analysis of Ransomware: Benefits, Limitations and use for Detection," 2016, [Online]. Available: http://arxiv.org/abs/1609.03020