

Dynamic Analysis of Android Malware Using Cuckoo Sandbox

Akor Jacob Terungwa *

Department of Security and Network Engineering, Innopolis University, Russia.

World Journal of Advanced Research and Reviews, 2025, 28(03), 521-526

Publication history: Received on 26 October 2025; revised on 05 December 2025; accepted on 08 December 2025

Article DOI: <https://doi.org/10.30574/wjarr.2025.28.3.4054>

Abstract

This research investigates dynamic analysis for Android malware detection, addressing the challenges posed by sophisticated, evasive mobile threats. The study employs a controlled Cuckoo Sandbox environment within a Windows-based virtualized environment to reveal malicious runtime behaviors by dynamically executing real malware samples, including WannaCry and CryptoLocker, in isolated virtual machines. Comprehensive behavioral features, such as API usage, system calls, network activity, and file system events, are robustly extracted and analyzed. The approach enables the identification of advanced malicious techniques, including process injection and anti-forensics, which often evade static detection. Despite high detection effectiveness, the study identifies limitations related to sandbox evasion and observation windows. Recommendations are made to enhance runtime simulation and adopt hybrid analysis strategies. The findings provide a practical, scalable framework for Android malware investigation, advancing dynamic analysis accuracy and resilience for operational cybersecurity applications.

Keywords: Android Malware; Cuckoo Sandbox; Dynamic Analysis; CuckooDroid; Behavioral Detection; Network Security; Cyber Threats

1. Introduction

Android has emerged as the world's most widely adopted mobile operating system, a trend that has simultaneously increased its exposure to sophisticated cyber threats (Singh et al., 2024). The surge in malicious Android applications, now numbering in the hundreds of thousands monthly, continues to challenge security systems across government, enterprise, and personal devices. Traditional static analysis, widely used due to its speed and scalability, is increasingly bypassed by malware employing heavy code obfuscation, dynamic loading, and encryption (Kumar et al., 2024). These techniques obscure malicious logic and limit the visibility of static scanners, especially when behavior is triggered only during execution.

Dynamic analysis has therefore become an essential complementary approach. By monitoring system calls, file system events, network communication, and runtime resource interactions, dynamic analysis offers deeper insight into how applications behave under real conditions. It is particularly effective against malware that leverages environmental awareness or delayed activation to evade detection (Check Point Research, 2015). However, most dynamic analysis tools still rely on virtualized or emulated environments that malware can easily fingerprint and evade, suppressing harmful behavior during runtime (Ruggieri et al., 2024). Short observation windows and lack of realistic user interaction further limit the behavior captured.

Given these challenges, there is a pressing need to improve dynamic analysis techniques to more accurately detect sophisticated, context-aware Android malware in realistic settings. This study addresses this gap by deploying a robust Cuckoo Sandbox environment within a Windows-based virtualized environment for Android malware execution. Through real malware samples and multi-dimensional monitoring, the study demonstrates how dynamic analysis can

* Corresponding author: Akor Jacob Terungwa

reveal complex behavioral and forensic indicators while identifying limitations that must be overcome for improved real-world applicability.

2. Problem Statement and Objectives

Current dynamic analysis systems often rely on emulated environments that sophisticated Android malware can detect and evade, resulting in incomplete behavior visibility and reduced detection accuracy. Furthermore, short monitoring windows and limited interaction simulation fail to trigger deeper malicious logic.

The study's objectives are to:

- Deploy a Cuckoo Sandbox environment for safe, controlled Android malware execution.
- Extract behavioral indicators including API calls, system operations, and network traffic.
- Evaluate the strengths and limitations of dynamic sandbox analysis for advanced malware.
- Recommend improvements for scalable and realistic malware analysis environments.

3. Methodology

This research adopts an empirical experimental design, deploying Cuckoo Sandbox with CuckooDroid to execute malware samples inside isolated virtual machines. Malware behavior was monitored across system calls, API usage, network traffic, file system events, and inter-process communication. Memory dumps were collected for Volatility-based forensic analysis.

The network configuration used for the malware execution environment is illustrated in Figure 1, which shows the VirtualBox Host-Only Adapter settings applied during sandbox deployment.



Figure 1 VirtualBox network adapter configuration GUI showing Host-Only Adapter settings

4. Findings

4.1. Behavioral Indicators

The comprehensive analysis of behavioral indicators documented in Figure 2 demonstrates the effectiveness of the deployed dynamic sandbox environment in detecting and characterizing sophisticated malware activities. Through detailed monitoring of process injections, system calls, and network communications—especially within critical system processes such as `svchost.exe` and `explorer.exe`—the framework successfully uncovered hallmark malicious

tactics including process hollowing and masquerading. Supplementary network analysis exposed command-and-control communications evidenced by DNS query patterns.

These findings validate the sandbox's capability to expose complex malicious behaviors that are often elusive to traditional static methods, thereby fulfilling a key measure of effectiveness. Moreover, the identification of both behavioral and network signatures contributes to robust malware fingerprinting and enhances detection accuracy essential for real-world applications.

4.1.1. WannaCry

Execution of WannaCry revealed encryption attempts, shadow copy deletion, and outbound C2 communication behaviors consistent with known ransomware activity.

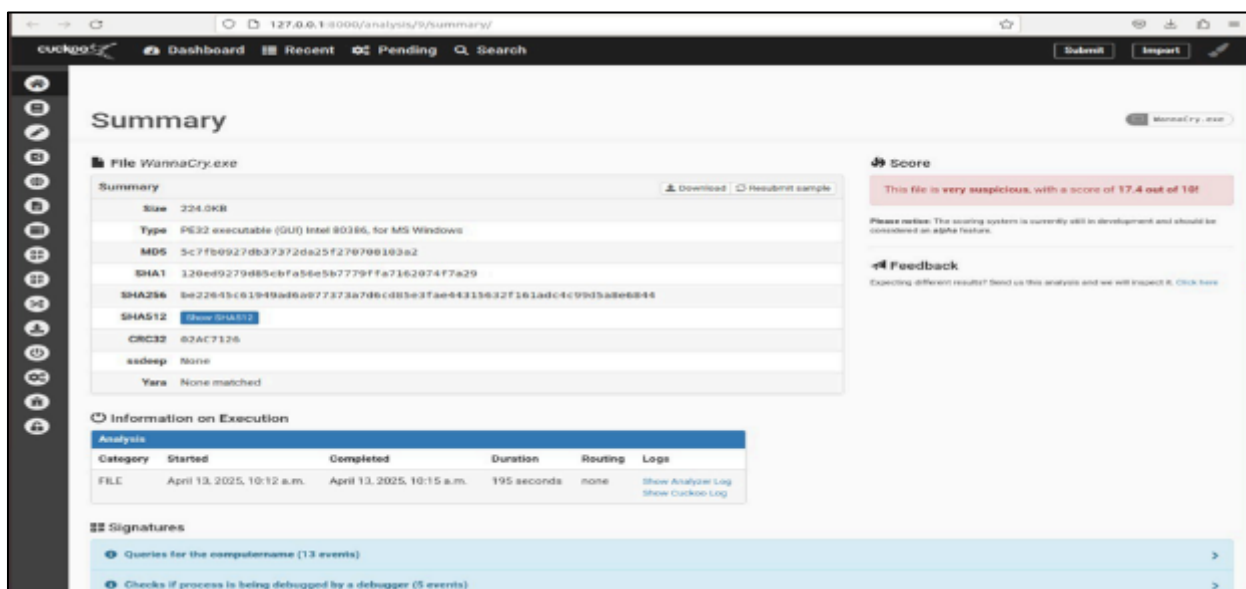


Figure 2 WannaCry Malware Execution Summary

From the above Figure 2, the log from the analysis confirms the execution of WannaCry ransomware (or a variant) with clear behavioral patterns matching its known malicious activities.

4.1.2. Analysis Summary of CryptoLocker

The behavioral patterns of the CryptoLocker sample revealed extensive process injection and disguised execution threads. These activities are summarized in Figure 3, which presents the execution logs highlighting injected processes and forensic evidence captured during Volatility analysis.

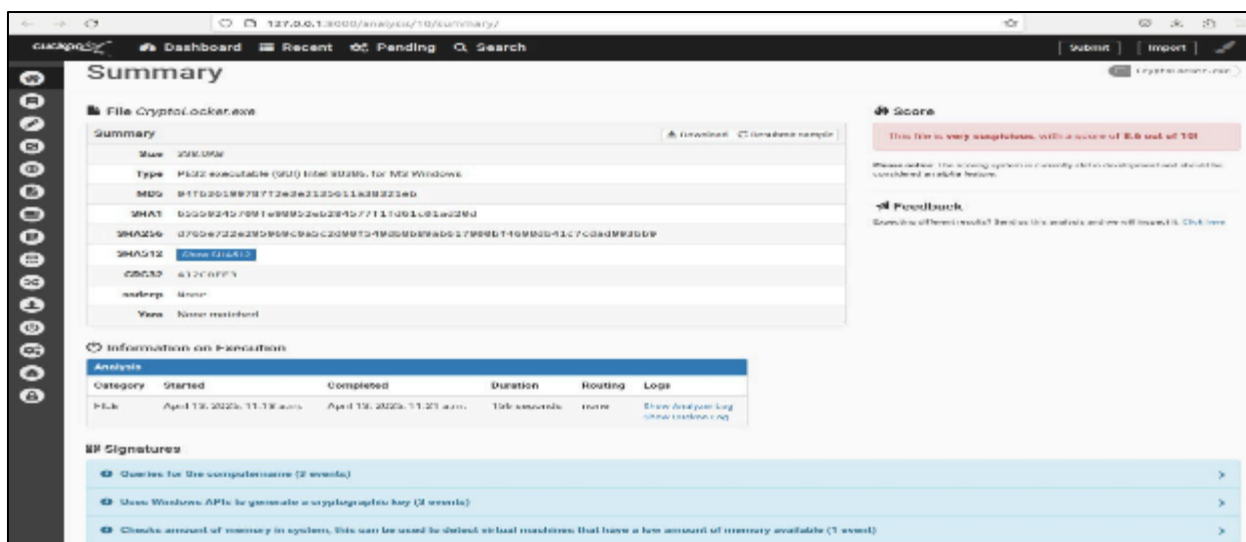


Figure 3 CryptoLocker Malware Execution Summary logs indicating numerous inject-x86.exe instances and process injection detected through volatility analysis. The analysis shows a successful detection of multiple process injections by malware into system processes such as explorer.exe and svchost.exe. Evidence of process hollowing and use of legitimate process names observed in Logs

4.2. Memory Forensics and Artifact Extraction

The deep forensic examination, detailed in Figure 4, further reinforces the sandbox's detection efficacy. Memory dumps captured during malware execution were subjected to analysis using the Volatility framework, revealing concealed and terminated processes along with injected code segments pivotal for forensic insight.

Significantly, volatility analysis detected sophisticated evasion techniques such as process name obfuscation and the presence of counterfeit 32-bit processes, corroborating the dynamic analysis platform's heightened sensitivity to hidden malware traits. These memory artifacts disclose persistence frameworks and rapid process cycling common in ransomware operations, underscoring the value of integrated memory forensics in augmenting visibility beyond surface-level activity.

This forensic capability complements active behavioral monitoring, delivering a multidimensional threat characterization and fortifying investigative accuracy.

Volatility Foundation Volatility Framework 2.0.1

OffSet(P) PID PPID Name Time created Time exited

0x000000007d6125f8	1416	416	0x0000000017c14000	2025-04-13 09:14:43:30 UTC+0000	
0x000000007d6125f8	916	484	0x0000000013bd7900	2025-04-13 09:14:43:33 UTC+0000	
0x000000007d6125f8	2980	624	0x0000000025c09000	2025-04-13 09:14:43:20 UTC+0000	
0x000000007d6125f8	2084	2468	0x0000000006e9f72000	2025-04-13 09:14:43:15 UTC+0000	2025-04-13 09:14:15 UTC+0000
0x000000007d6125f8	1764	484	0x000000001c7b1900	2025-04-13 09:14:43:26 UTC+0000	
0x000000007d6125f8	1812	884	0x000000001b3c9000	2025-04-13 09:14:43:27 UTC+0000	
0x000000007d6125f8	1044	1012	0x000000001b7c0000	2025-04-13 09:14:43:27 UTC+0000	
0x000000007d6125f8	1984	1844	0x000000001a8f7900	2025-04-13 09:14:43:29 UTC+0000	
0x000000007d6125f8	2024	2468	0x00000000102b0000	2025-04-13 09:14:43:42 UTC+0000	2025-04-13 09:14:28 UTC+0000
0x000000007d6125f8	1982	1844	0x000000001a8f9000	2025-04-13 09:14:43:29 UTC+0000	
0x000000007d6125f8	3048	2468	0x000000004d37b000	2025-04-13 09:12:29 UTC+0000	2025-04-13 09:12:29 UTC+0000
0x000000007d6125f8	688	484	0x000000002a81f900	2025-04-13 10:44:11 UTC+0000	
0x000000007d6125f8	744	484	0x0000000029131000	2025-04-13 09:14:43:13 UTC+0000	
0x000000007d6125f8	792	484	0x0000000023863900	2025-04-13 09:14:43:13 UTC+0000	
0x000000007d6125f8	884	484	0x0000000022944000	2025-04-13 09:14:43:13 UTC+0000	
0x000000007d6125f8	912	484	0x0000000022984b00	2025-04-13 09:14:43:13 UTC+0000	
0x000000007d6125f8	120	484	0x000000002b55a000	2025-04-13 09:14:43:14 UTC+0000	
0x000000007d6125f8	772	484	0x0000000023862900	2025-04-13 09:14:43:15 UTC+0000	
0x000000007d6125f8	1124	484	0x0000000025d67000	2025-04-13 09:14:43:17 UTC+0000	
0x000000007d6125f8	580	484	0x0000000027b42e00	2025-04-13 10:44:09 UTC+0000	
0x000000007d6125f8	588	484	0x000000002b138000	2025-04-13 10:44:09 UTC+0000	
0x000000007d6125f8	624	484	0x0000000023d0e000	2025-04-13 10:44:10 UTC+0000	
0x000000007d6125f8	2364	2468	0x0000000055007900	2025-04-13 09:14:50 UTC+0000	2025-04-13 09:14:57 UTC+0000
0x000000007d6125f8	484	484	0x000000002b8e7900	2025-04-13 10:44:09 UTC+0000	
0x000000007d6125f8	436	484	0x0000000022c40700	2025-04-13 10:44:09 UTC+0000	
0x000000007d6125f8	484	484	0x0000000022c16000	2025-04-13 10:44:09 UTC+0000	
0x000000007d6125f8	918	792	0x000000002290f000	2025-04-13 09:14:43:14 UTC+0000	
0x000000007d6125f8	2000	484	0x00000000460f1900	2025-04-13 09:14:40:20 UTC+0000	
0x000000007d6125f8	416	396	0x0000000022c16000	2025-04-13 10:44:09 UTC+0000	
0x000000007d6125f8	2616	2468	0x000000003a2fd000	2025-04-13 09:12:47 UTC+0000	2025-04-13 09:12:48 UTC+0000
0x000000007d6125f8	912	484	0x0000000027138000	2025-04-13 09:12:37 UTC+0000	2025-04-13 09:12:37 UTC+0000
0x000000007d6125f8	1436	484	0x00000000457f9000	2025-04-13 09:14:40:21 UTC+0000	
0x000000007d6125f8	1316	484	0x00000000344f7900	2025-04-13 09:14:40:21 UTC+0000	
0x000000007d6125f8	244	484	0x000000002d9b0000	2025-04-13 10:44:09 UTC+0000	
0x000000007d6125f8	2696	2468	0x000000001670c000	2025-04-13 09:12:48 UTC+0000	2025-04-13 09:12:49 UTC+0000
0x000000007d6125f8	456	396	0x000000002bf60000	2025-04-13 10:44:09 UTC+0000	
0x000000007d6125f8	2180	2468	0x000000004ca14000	2025-04-13 09:14:12 UTC+0000	2025-04-13 09:14:13 UTC+0000
0x000000007d6125f8	2692	2468	0x0000000026000000	2025-04-13 09:14:20 UTC+0000	2025-04-13 09:14:20 UTC+0000
0x000000007d6125f8	2712	2468	0x00000000a97900	2025-04-13 09:13:05 UTC+0000	2025-04-13 09:13:06 UTC+0000
0x000000007d6125f8	1088	2468	0x00000000c0c00000	2025-04-13 09:14:15 UTC+0000	2025-04-13 09:14:15 UTC+0000
0x000000007d6125f8	2468	2468	0x0000000009dab000	2025-04-13 09:13:05 UTC+0000	2025-04-13 09:13:06 UTC+0000
0x000000007d6125f8	3036	2468	0x00000000677f0000	2025-04-13 09:13:06 UTC+0000	2025-04-13 09:13:11 UTC+0000
0x000000007d6125f8	2674	2468	0x0000000060107900	2025-04-13 09:13:06 UTC+0000	2025-04-13 09:13:10 UTC+0000
0x000000007d6125f8	876	484	0x000000006055e900	2025-04-13 09:14:19 UTC+0000	
0x000000007d6125f8	1076	2468	0x000000001f41b000	2025-04-13 09:14:57 UTC+0000	
0x000000007d6125f8	3032	2468	0x000000002d790000	2025-04-13 09:14:38 UTC+0000	2025-04-13 09:14:39 UTC+0000
0x000000007d6125f8	2652	2468	0x00000000b0baa000	2025-04-13 09:13:57 UTC+0000	2025-04-13 09:13:59 UTC+0000
0x000000007d6125f8	1136	2612	0x000000007e4d9000	2025-04-13 09:13:56 UTC+0000	2025-04-13 09:13:57 UTC+0000
0x000000007d6125f8	2580	2468	0x0000000077fd0000	2025-04-13 09:14:02 UTC+0000	2025-04-13 09:14:28 UTC+0000
0x000000007d6125f8	1592	1308	0x00000000e13f0000	2025-04-13 09:13:54 UTC+0000	
0x000000007d6125f8	880	2468	0x00000000b13ac000	2025-04-13 09:14:39 UTC+0000	2025-04-13 09:14:39 UTC+0000
0x000000007d6125f8	2684	484	0x0000000017b0b000	2025-04-13 09:15:01 UTC+0000	
0x000000007d6125f8	2852	624	0x000000003202f000	2025-04-13 09:15:46 UTC+0000	
0x000000007d6125f8	3080	2468	0x00000000672027000	2025-04-13 09:14:20 UTC+0000	2025-04-13 09:14:26 UTC+0000
0x000000007d6125f8	2480	2468	0x00000000601b1a00	2025-04-13 09:14:11 UTC+0000	2025-04-13 09:14:12 UTC+0000
0x000000007d6125f8	1140	2468	0x0000000023cc0000	2025-04-13 09:14:39 UTC+0000	2025-04-13 09:14:49 UTC+0000
0x000000007d6125f8	736	2468	0x00000000914c00	2025-04-13 09:13:40 UTC+0000	2025-04-13 09:13:51 UTC+0000
0x000000007d6125f8	1076	2468	0x000000002d8c1000	2025-04-13 09:14:39 UTC+0000	2025-04-13 09:15:06 UTC+0000
0x000000007d6125f8	2648	2636	0x0000000026702000	2025-04-13 09:14:11 UTC+0000	
0x000000007d6125f8	2696	2636	0x0000000060110000	2025-04-13 09:14:16 UTC+0000	
0x000000007d6125f8	2520	2468	0x00000000279e2000	2025-04-13 09:14:44 UTC+0000	

Figure 4 Memory Dump Analysis Using Volatility command outputs showing identified processes related to WannaCry and CryptoLocker

4.3. Sandbox Limitations and Observed Evasion Techniques

Multiple evasion techniques were observed:

- Delayed execution exceeding common sandbox analysis windows
- Virtual machine and debugger detection, preventing full payload activation
- Randomized process names and dynamic code loading to bypass pattern matching
- Encrypted communication masking true network intent

These evasion behaviors confirm the need for extended runtime simulation and anti-evasion enhancements in analysis systems.

5. Implications for Practice

The study highlights several practical considerations:

- Dynamic analysis is essential for detecting behavioral traits of advanced Android malware.
- Hybrid analysis pipelines combining static, dynamic, and memory forensics provide stronger detection accuracy.
- Sandboxes must mimic realistic device behavior to reduce fingerprinting risks.
- Extended execution windows and automated user-interaction simulation can reveal more complete behavior.
- Behavioral logs can be further used for machine-learning-based threat classification.

6. Conclusion

This study set out to explore dynamic analysis methodologies for malware detection and characterization, centering on deploying the Cuckoo Sandbox framework in a controlled virtualized environment. The research established a reproducible and effective sandbox infrastructure, executed real-world malware samples including prominent ransomware families such as WannaCry and CryptoLocker, and leveraged automated behavioral monitoring alongside forensic analysis techniques to extract meaningful indicators of compromise (IOCs) (Singh et al., 2024).

References

- [1] Alruhaily, M. (2018). Limitations in malware dynamic analysis (Doctoral dissertation). University of Birmingham, UK. <https://etheses.bham.ac.uk/id/eprint/8457/1/Alruhaily18PhD.pdf>
- [2] Kaspersky. (2020, June 7). Ransomware WannaCry: All you need to know. Kaspersky Resource Center. <https://www.kaspersky.com/resource-center/threats/ransomware-wannacry>
- [3] Ruggieri, S., et al. (2024). Unveiling the dynamic landscape of malware sandboxing. International Journal of Advanced Computer Science and Applications, 15(3). <https://thesai.org/Publications/ViewPaper?Volume=15&Issue=3&Code=IJACSA&SerialNo=137>
- [4] Singh, A., Tanha, M., Girdhar, Y., & Hunter, A. (2024). Interpretable Android malware detection based on dynamic analysis. International Conference on Security and Privacy. <https://www.scitepress.org/Papers/2024/124158/124158.pdf>