World Journal of Advanced Research and Reviews
eISSN: 2581-9615 CODEN (USA): WJARAI
Cross Ref DOI: 10.30574/wjarr
Journal homepage: https://wjarr.com/

(RESEARCH ARTICLE)

Check for updates

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.

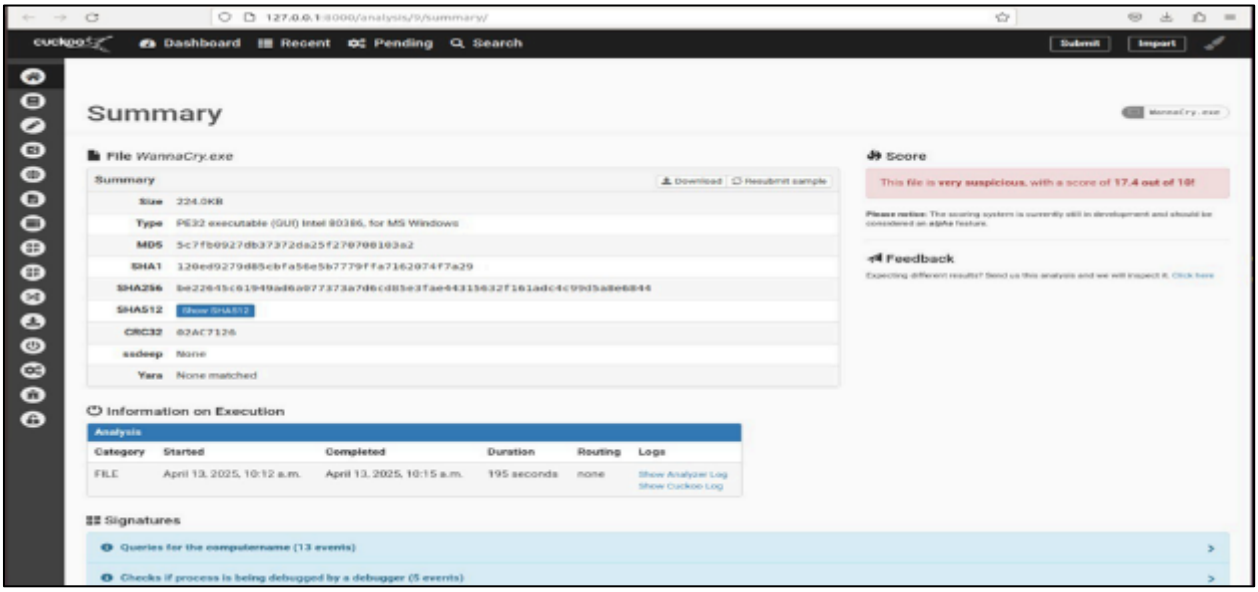

Figure 2 WannaCry Malware Execution Summary

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 CrytoLocker

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.

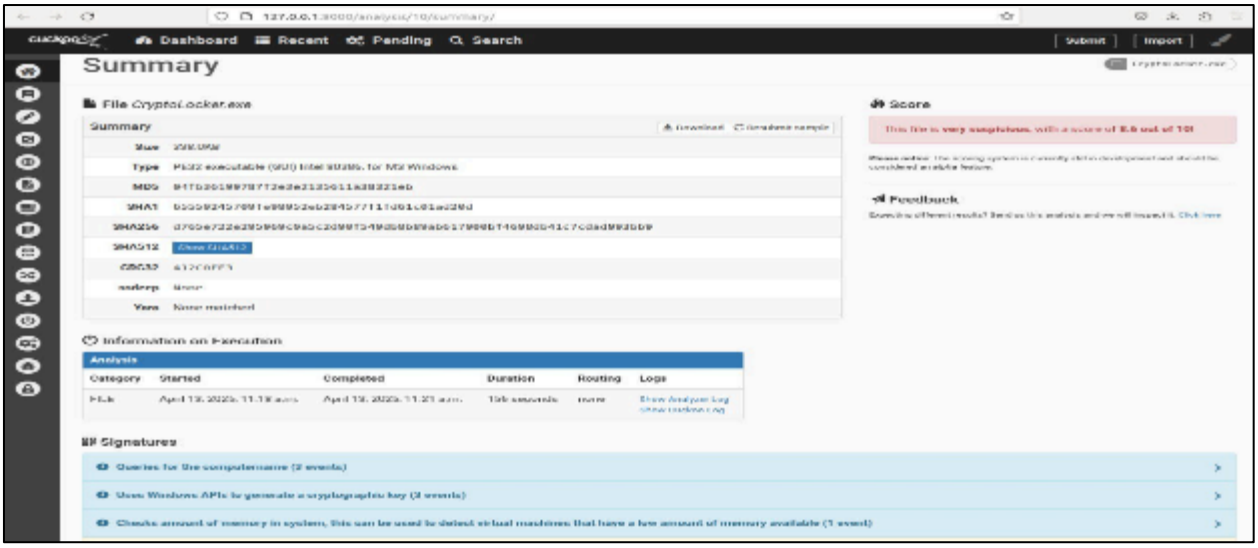

Figure 3 CrytoLocker Malware Execution Summary logs indicating numerous inject-x86.exe instances and process injection detected through volatility analysis. The analysis shows a successful detection of multiple process injections by malware into system processes such as explorer.exe and svchost.exe. 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.

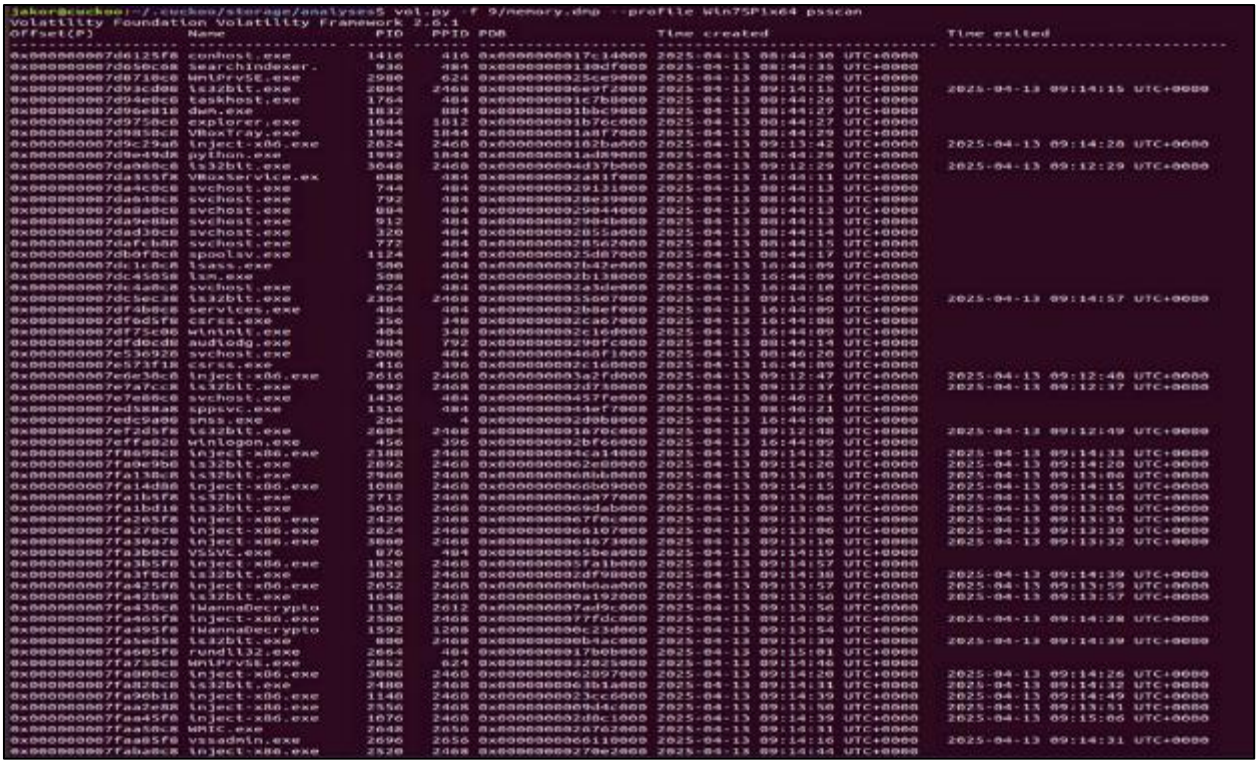

Figure 4 Memory Dump Analysis Using Volatility command outputs showing identified processes related to WannaCry and CryptoLocker

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