Defensive Honeypots for IP IoT Devices: Quantitative Comparison between Vanilla and Sandboxed Honeypots

Franek Kruczynski

September 2025

Contents

Chapter 1

Introduction

1.1 Background

The Internet of Things (IoT) is exponentially expanding, driving a brand new and complex wave of device interconnectivity worldwide, with an expected 27 billion devices by the end of 2025 *(Jinesh, 2025)*. Consequently, malware is becoming much more sophisticated and advanced, with a parallel increase of its deployment against IoT devices *(Alasmary, 2019)*, resulting in modern malware being powered through machine learning and AI methods. In essence, this poses a significant risk to IoT devices, its consumers and network infrastructures.

With such, security researchers and organisations worldwide are faced with having to prevent malicious Cyber-attacks targeting IoT devices. Due to the variation of devices and different professional standards, many devices lack robust security measures. They are often deployed with out-of-date firmware, varying degrees of processing capabilities *(Gurunath, 2019)* and, varying levels of network defences – all of which malware may exploit to cause propagation *(its spread)*.

As a result, understanding and analysing both malware behaviour and patterns has become a fundamental practice in Cyber Security research. Traditional network defences, such as firewalls, VPNs and even IPS/IDS systems often fail in capturing and mitigating Zero-Day, XSS and MITM attacks. Decoy mechanisms mimicking IoT devices, commonly known as Honeypots, are used to lure attackers and have become the norm in network deployment *(Narendran, 2025)*, providing an effective environment to analyse such attacks with security.

However, the degree of an attacker's interaction with a Honeypot significantly influences both the data captured, and the severity of an attack. Low-interaction Honeypots (vanilla) are commonly preferred due to their ease of setup, but do not provide nearly as thorough security and analytical measures as high-interaction

sandboxed Honeypots do *(Kocaogullar, 2023)*. Therefore, the balance of containment and interactivity remains a vague area of research.

This project aims to address such a research gap by providing a quantitative comparison of varying Honeypots and their security measures, evaluating the effectiveness of isolation methods used to prevent malware mitigation within IoT Honeypot frameworks.

1.2 Aims & Objectives

1.2.1 Aim

To evaluate the effectiveness of containment and sandboxing mechanisms, in preventing malware propagation (specifically its spread into external systems) within a IoT IP device Honeypot framework. Such will be achieved through quantitatively comparing the same malware programs on two separate Honeypots, contained high-interaction against vanilla low-interaction.

1.2.2 Objectives

The objectives are as follows:

- Design, develop and deploy a secure Honeynet framework within Virtual Machines,
- To deploy two separate Honeypots:
  1. **Vanilla Honeypot:** Low-interaction with no advanced security,
  2. **Sandboxed Honeypot:** High-interaction within a secured container.
- To create and design a virtual network, providing both logical addressing to all IoT IP devices, Virtual Machines and, providing security through sub-netting. In essence mimicking a small office network.
- To collect and store the following malware properties for quantitative comparison and analysis:
  1. Network traffic,
  2. Payloads,
  3. Malware type
  4. Activity data

5. Propagation attempts outside the container.

- Quantitatively compare the data of all malware, and conclude whether attack behaviors differ based on environment.

1.3 Product Review

1.3.1 Scope

The project involves the design and deployment of a secure IoT Honeynet environment for a range of IP IoT devices simulated within a small office network. Two separate and distinct honeypots will be implemented, a high-interaction sandboxed Honeypot and, a low-interaction vanilla Honeypot operating within a secure, isolated Docker container. The Honeypots will operate within a singular Virtual Machine sequentially.

Each malware sample will be executed within the Honeypots, with all relevant data including its type, payloads and various activity logs securely collected. A dedicated Virtual Machine (separate to the Honeynet) will store and process the data using tools such as ElasticSearch, guaranteeing all data remains isolated from the external network.

In essence, the project will simulate realistic attacking behaviours and sequences of events that occur during malware spread within smaller networks. Findings will lead to a better understanding of the importance of segmentation within networks and Honeypots, to aid in strengthening IoT security by identifying attacker patterns *(Kocaogullar, 2023)*. Furthermore, it'll help expand on pre-existing research and highlight the importance of practical implementation methods for secure environments.

1.3.2 Audience

The primary audience for this project involves Cyber Security researchers, network analysists and academic institutions which study digital forensics, analyse malware behaviour and the administrators that deal with IoT security. Such groups will benefit from the projects structured and quantitative evaluation of containment-based Honeypots and networks, as it provides improved threat detection and response; enhanced threat intelligence through dynamic analysis and prevention; and the overall impact of a secure Honeypot environment.

Furthermore, Cyber Security lecturers and students will benefit through providing a practical framework for a safe and secure virtual environment for malware deployment, for experimentation and research. The projects architecture may server as a teaching tool for demonstrating both the dangers of types of malware, and analysis practices for security.

Lastly, IoT administrators and developers in smaller organisations may gauge a better understanding of malware behaviour within contained and non-contained environments. Understanding malware patterns with supporting data may help in more advanced intrusion detection systems (IDS), through ensuring comprehensive flaws within IoT devices are identified *(Fortinet, 2025)* and mitigated.

Chapter 2

Background Review

2.1 Existing Approaches

Numerous approaches exist for malware analysis, one of which is a static analysis approach using a Windows-based framework to execute malware *(Moser A, 2008)* within a contained environment that may be applied to IoT environments. The system focuses on registering system changes and runtime behaviour to analyse samples. However, the framework lacks the advanced emulation of IoT devices; has many scalability issues relating to resource management and, fails to highlight the importance of containment within malware analysis.

Another approach is a systematic, behaviour analysis of common malware samples within a Cuckoo Sandbox *(Genç, 2019)*, strongly relating to the project. The authors observed malware behaviours, however a specific sample, *TeslaCrypt*, managed to bypass detection within the Sandbox and was only identified post-execution. Ultimately, the Sandbox failed to capture critical data and the true payload data, highlighting the importance that Honeypot containment alone is not enough.

2.2 Related Literature

The rising frequency of the relationship between malware and IoT devices has lead to a surge in research development, dedicated in understanding both attacker behaviour patterns and, defensive techniques. *(Alasmary, 2019)* focuses on a graph-based approach to malware analysis, focusing on polymorphic malware utilising unreachable code. While their work demonstrated improvements in behavioural analysis, the approach was limited due to using historical data and

lacked an environment for dynamic data collection; a critical area in IoT analysis.

(Genç, 2019) explored analysis within a popular Sandbox, revealing that modern malware has the capability to evade security systems and remain undetected even post-execution. This finding identified a critical error within containments systems. While malware may be thoroughly analysed, the payload may be unable to be examined if the sample supresses itself. Therefore, the aspect of containment alone is not sufficient for IoT malware analysis and contained Honeypot environments require effective security measures to integrate deception.

Lastly, *(Narendran, 2025)* briefly mentions the importance of Honeypots within modern IoT architectures, rendering them an essential requirement for intelligence collection on active and common threats. However, their integration within IoT frameworks remains a vague area of research, as differentiating devices result in varying challenges.

Chapter 3

Methodology & Techniques

3.1 Approach

The project follows an experimental quantitative comparison approach to evaluate effectiveness of malware containment and propagation within IoT Honeypot frameworks. The methodology directly links back to Section 1.2.2, through providing a controlled environment for testing, research and deployment between two configurations. All testing will be performed in Ubuntu, a Linux distribution.

Two Virtual Machines deployed within Oracle's VirtualBox will form the overall framework. The Lab VM will sequentially run both Honeypots:

- **Vanilla Honeypot**: Low-interaction Cowrie instance operating without additional containment.
- **Sandboxed Honeypot**: Identical Cowrie instance deployed within a **FireJail and Docker** container. It will utilise Linux-based Kernel security like seccomp and AppArmor, used for filtering network packets and restricting sample system calls.

A separate Analysis VM will collect data, process logs and analyse packets using **ElasticSearch**, ensuring the VM remains disconnected from external networks.

Each sample will be executed within both Honeypot configurations using identical conditions to prevent data skew. Network traffic, pcaps and payload data will be recorded to generate the quantitative comparison, most likely stored within CSV files. Key metrics will include propagation attempts, program events and communication patterns.

Integrity will be maintained through host OS level kill-switches, VM snapshot rollbacks and network segmentation using subnetting. These measures ensure the

host machine remains unaffected, including the external network. Lastly, all collected data will be quantitatively analysed to determine whether Sandboxed Honeypots provide a noticeable difference in security, preventing malware propagation compared to the vanilla configuration. The results will demonstrate the benefits of secure containment for malware analysis.

3.2 Technologies

The project will utilise a range of open-source programs and technologies to aid in the analysis. The core framework will be built within Oracle VirtualBox due to its flexibility for Sandbox implementations, snapshot rollbacks and inter-connectivity between separate VMs. Each VM will run an instance of Lubuntu; a popular, stable and secure Linux distribution.

The Honeypot framework will operate within Cowrie, an SSH/Telnet Honeypot designed for IP-based malware capture. It is widely used due to integrated structural logging capabilities for analysis and a large public documentation. Docker, a lightweight portable container, will be used to Sandbox the high-interaction Honeypot, ensuring low system level isolation. Firejail and AppArmor will provide additional kernel-level security by restricting access to malware samples to specific system calls and functions.

ElasticSearch will provide real-time log indexing and data storage using the RESTful engine, supplemented by WireShark for packet capturing. Furthermore, Python scripts will automate data extraction from the samples and format them into appropriate CSV files to be used for the final quantitative comparison.

3.3 Version Control & Management

GitHub, a web-based platform for version control, will store all source code, configuration files and documentation, providing a transparent history of the project's development. All changes will be pushed with detailed commits to ensure professionalism. During varying development stages, branching will be used to manage maintain a streamlined approach.

Additionally, backups will be stored within a Google Drive accessible by the supervisor for data integrity, it'll ensure data synchronisation across various systems. Sensitive data (malware samples and system configurations) will mainly be stored within the Analysis VM, with the aim to maintain ethical and security compliances.

Chapter 4

Project Management

4.1 Activities

The activities you include here should relate back to the objectives of Section 1.2.2. What do you need to do to reach this objectives and eventually deliver on the aim?

4.2 Schedule and Time Management

The project will follow a weekly schedule to ensure a streamlined and continuous development approach.

- Supervisor Meetings: Weekly scheduled meetings used for formative feedback and validation of the project's methodology.
- Calendar Planning: A personal calendar used to dedicate weekly slots for all aspects of development, such as collecting sufficient malware samples.
- Time Management: Balancing between the dissertation and independent coursework's is required, therefore personal study and research time will be allocated in daily chunks to steady workflow.

A Gantt chart will be employed to visualise a clear roadmap for such a complex project, to ensure arbitrary milestones are met. Additionally, snapshot rollbacks will be integrated within the schedule to avoid testing errors.

4.3 Data Management

- How is this data going to be stored? Analysis vm stored sensitive samples etc - CSV files for extracting

4.4 Deliverables

??

Bibliography

Alasmary, H et al. (2019). *Analyzing, Comparing, and Detecting Emerging Malware: A Graph-based Approach*. URL: https://arxiv.org/abs/1902.03955.

Fortinet (2025). *What Is An Intrusion Detection System (IDS)?* URL: https://www.fortinet.com/uk/resources/cyberglossary/intrusion-detection-system.

Genç, Z. A et al. (2019). *Case Study: Analysis and Mitigation of a Novel Sandbox-Evasion Technique*. URL: https://pure.royalholloway.ac.uk/ws/portalfiles/portal/34760373/cecc2019GLS.pdf.

Gurunath, R et al. (2019). *An Overview: Security Issue in IoT Network*. URL: https://ieeexplore.ieee.org/abstract/document/8653728.

Jinesh (2025). *How Many IoT Devices Are There in 2025?* URL: https://autobitslabs.com/how-many-iot-devices-are-there/.

Kocaogullar, Y et al. (2023). *Hunting High or Low: Evaluating the Effectiveness of High-Interaction and Low-Interaction Honeypot*. URL: https://kar.kent.ac.uk/102122/1/STAST2022.pdf.

Moser A, Y et al. (2008). *Limits of Static Analysis for Malware Detection*. URL: https://ieeexplore.ieee.org/document/4413008.

Narendran, Vaideeswaran (2025). *What Is a Honeypot?* URL: https://www.crowdstrike.com/en-gb/cybersecurity-101/exposure-management/honeypots/.