National College of Ireland

Using Honeypots and Deceptive Technology Approach to Improve IoT security by Preventing MITM Attacks
IoT Security

MSc Research Project
Programme Name

Goutham Krishna Plamparambil Dinumon
Student ID: 23277840

School of Computing
National College of Ireland

Supervisor: Mark Monaghan

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Goutham Krishna Plamparambil Dinumon
Student ID:	23277840
Programme:	MSc Cybersecurity **Year:** 2024-2025
Module:	Practicum Part 2
Supervisor:	Mark Monaghan
Submission Due Date:	15/09/2025
Project Title:	Using Honeypots and Deceptive Technology Approach to Improve IoT security by Detecting MITM Attacks
Word Count:5347	**Page Count-20**

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:

Date: 15/09/2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the	□

project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Using Honeypots and Deceptive Technology Approach to Improve IoT security by Preventing MITM Attacks

IoT Security
Goutham Krishna Plamparambil Dinumon
Student ID-23277840

Abstract

The proliferation of resource-constrained Internet-of-Things (IoT) devices has greatly increased the potential attack surface since Man-in-the-Middle (MITM) interception is a harmful threat that is nearly invisible. Conventional IoT intrusion-detection approaches do typically operate passively and they flag anomalies only after a compromise, yet hardware-focused countermeasures neglect network-layer exploits. A slim proactive guard can tempt foes then log actions as they happen.

This dissertation designs and also implements then evaluates a medium-interaction honeypot framework that can detect MITM attacks within IoT environments. The aims specifically include emulating Telnet-based IoT services, which should attract MITM actors, also capturing and logging attacker login attempts, session commands, and packet traces. Also, the goals include assessing detection ability, attacker lag time, and system resource load.

Inside a VirtualBox sandbox totally isolated were an Ubuntu honeypot, a Kali Linux attacker, and a Windows victim machine. The Ubuntu honeypot did run Cowrie while the Kali Linux attacker did use Bettercap for ARP spoofing. Telnet session interception as well as malicious command injection simulated MITM attacks. Key behavioural metrics were extracted by analysing Cowrie JSON logs with network traffic captured using tcpdump.

Across repeated MITM attack simulations, the honeypot did successfully capture all connection attempts as well as record complete session transcripts and quantitatively measure attacker dwell-time. Average dwell-time was longer than a truthful baseline. The extension was for over six minutes now. Resource consumption remained minimal, CPU overhead stayed below 3 %, memory usage increased to approximately 18 MB, thus the solution was viable for constrained environments.

Combination of medium-interaction service emulation along with targeted MITM engagement is closing of the gap that is existing between passive anomaly detection and active opponent interaction. These findings do add to the body of knowledge on lightweight cyber-deception for IoT networks since they do show that fact.

To collect live attack traces, improve IDS signature accuracy, also strengthen firewall rules without specialized hardware, small and medium-sized enterprises can deploy the framework, a reproducible, low-cost blueprint, within virtualised labs.

The current implementation focuses on a single-protocol Telnet lure; it would be a promising direction for further research toward extending the approach to multi-protocol scenarios, encrypted traffic analysis, as well as automated lure adaptation via reinforcement learning.

1. Introduction

1.1 Background and Context

The Internet of Things (IoT) now connects up billions of different sensors and actuators. It connects embedded controllers within factories, homes, hospitals and smart-city infrastructures. This ubiquitous connectivity allows unprecedented automation and data-driven decision-making. It also shows many cheap devices with few resources that do not have enterprise security inside. Man-in-the-Middle attacks are especially pernicious among all of the threats that do exploit this weakness. An opponent is able to intercept silently and also alter traffic that is between endpoints thus enabling data theft and command manipulation in addition to full device compromise. Many IoT protocols are still transmitted within clear text or rely on weak authentication. Successful MITM incursions can actually falsify sensor readings then disrupt critical actuators that exist in smart grids also hijack consumer devices without any detection.

1.2 Problem Statement

Sivasankari and Kamalakkannan (2022) use a regression-modelling approach which is one example of existing IoT security research that has produced lightweight anomaly-detection engines. This method identifies odd delay variance, RTT spikes and packet-loss patterns as possible MITM evidence. Such static statistical methods can be efficient, yet attacks are what they observe only after indicators surface and they do not engage the opponent nor collect the needed artefacts that are needed to harden future defences. Conversely, silicon-layer Trojans can be found with hardware-centric honeypots (Omar et al., 2024) yet network-layer MITM manoeuvres are not perceived, and HoneyIoT (Guan et al., 2023) such as adaptive reinforcement-learning honeypots deceive dynamically yet do not optimise resource-bound deployments or MITM traffic flows.

1.3 Research Gap

Consequently, no integrated lightweight framework couples' network-layer deception alongside real-time MITM analysis for IoT devices. Earlier studies predict attacks using statistics without engaging attacks or stress hardware Trojans instead of network interception. Furthermore, it advances adaptive honeypots in the absence of tailoring them to MITM protocols such as HTTP, MQTT, as well as Telnet. In the present study, this deficit is addressed by fusing honeypot-based cyber-deception with protocol-specific lures that attract, log, and study MITM adversaries as they operate.

1.4 Research Question

In the context of IoT-device security, how can Man-in-the-Middle attacks be identified, mitigated and resisted using honeypots and deceptive technology?

1.5 Objectives

 Develop an IoT honeypot system with configured features suitable for MITM-attack detection via Telnet protocol emulation.
 Capture and document attacker behaviour in real time as they interact with the decoys.

 Evaluate the effectiveness of the deception measures as a defensive layer against MITM attacks.

1.6 Significance and Contribution

This study matters because proactive deception protects IoT deployments with limited resources practically at a low overhead. The framework shields genuine devices at the same time that it is generating rich behavioral telemetry by way of luring would-be intruders in the direction of carefully created decoy services like Telnet and after that recording each and every stage of their Man-in-the-Middle maneuvers. These live attack traces feed directly into refined intrusion-detection signatures also blacklists in addition to automated response rules, which then close the feedback loop between reconnaissance, detection, together with policy hardening. The project offers up a reproducible blueprint that small as well as medium-sized enterprises can deploy inside nauthorize labs because it broadens access so advanced deception techniques that are often considered the preserve of well-resourced organisations. Beyond its immediate practical value, the research fills a prominent gap within current cybersecurity literature and practice, contributing to academic comprehension of how lightweight, protocol-aware honeypots can reduce MITM threats in the rapidly expanding—but chronically under-secured-IoT ecosystem.

1.7 Scope, Assumptions and Limitations

Researchers all experiment in a sandbox based on VirtualBox that is fully isolated; personal data do not expose, and no networks produce. The honeypot is a medium-interaction replica of a Linux-based IoT endpoint because it prioritizes safety along with low resource consumption over full-stack emulation. Presently, the implementers focus just on Telnet traffic, not including fake IoT commands or multi-protocol lures like HTTP or MQTT. Thus, results apply mostly to IoT Linux small devices using unencrypted or weakly authenticated channels. Systems which are of high bandwidth or of industrial grade or fully TLS enabled lie beyond what is the immediate scope. Future paths for extension exist from them.

2.0 Literature Review

In this chapter, academic and technical literature related to IoT security is critically reviewed, especially focusing on Man-in-the-Middle (MITM) attacks and on honeypot-based deception. Emerging deception strategies as well as honeypot architectures (software and hardware), intrusion detection systems (IDS), and statistical detection are covered in the review, grouped by theme. The study concludes by synthesizing research gaps it seeks to address.

2.1 Understanding the IoT Security Landscape

The Internet of Things (IoT) ecosystem has grown from zero but continues to suffer from serious security vulnerabilities. Device heterogeneity, limited hardware capabilities, coupled with inadequate embedded safeguards cause these vulnerabilities. Studies have highlighted key categories of risk that do include weak authentication and default credentials. Weaknesses at the protocol level and poor firmware patching are also categories that present some risk (Kumar et al., 2022; Khanam, 2023).

Securing of millions of distributed endpoints is indeed complex, Khraisat and Alazab (2021) and Bryant and Saiedian (2022) emphasise, especially for when many do rely on lightweight protocols like MQTT and CoAP. Attacks at various network and hardware levels can equally affect business-critical infrastructure (Alawadhi et al., 2022) and consumer devices (Joel et al., 2023).

Man-in-the-Middle (MITM) attacks are particularly dangerous among these in that they compromise confidentiality as well as inject falsified commands and often remain undetected in plaintext protocol contexts (Haris et al., 2023). Because MITM threats are subtle, persistent, and protocol-driven, they are harder to trace or prevent. Against MITM attacks, customary firewalls or anomaly-based detectors are less effective than DdoS (Amit et al., 2022).

2.2 Statistical and Regression-Based Detection of MITM

Network-statistics-based regression modeling is just one lightweight method for the detecting of MITM attacks. This is on account of its light weight. Sivasankari and Kamalakkannan (2022) presented a solution using Gaussian Process Regression that was trained over NS2-simulated IoT traffic. Delay anomalies, packet loss, and RTT variance indicated MITM activity to the model. Because it is appropriate for IoT nodes with few resources, its detection accuracy exceeded 97%.

Alyasiri et al. (2021) introduced an IDS based on grammar evolution that could evolve patterns automatically to identify cyber-attacks like MITM because it made new signatures from network flow data. However, such statistical systems are reactive because they flag anomalies only after they surface in traffic. In adversarial settings, in which attackers do alter traffic characteristics for evasion of detection, their accuracy is reduced too (Khraisat and Alazab, 2021).

Neto et al. (2023) developed the CICIoT2023 dataset in order to improve nauthorized n in addition to testing and also to offer real-time labelled data that is for large-scale IoT attacks. This improves reproducibility within IDS research. However, it still does not replace real-time engagement with deception strategies that actively manipulate how adversaries behave.

2.3 Intrusion Detection Systems and Their Limitations

More general IDS systems have been widely applied to IoT networks for the reason that they cover both signature-based and anomaly-based models. Work by Kumar et al. (2022), Gupta

and Lingareddy (2021), and Hromada et al. (2021) is highlighting the accuracy versus system resource usage trade-off.
Systems of machine learning like those Lightbody et al. (2023) developed showed high accuracy detecting power side-channels yet need much computation. IDS frameworks when deployed on Raspberry Pi environments (Mohd Bakry et al., 2022) are likewise promising but vulnerable to adversarial bypasses if lacking embedded deception mechanisms.

2.4 Honeypots and Deception as Proactive Defences

2.4.1 Low- and Medium-Interaction Honeypots

Low-interaction honeypots include IoTPOT (Pa et al., 2016) and U-PoT (Hakim et al., 2018). They mimic familiar IoT services like Telnet and UpnP to find brute-force and vulnerability-scanning actions. Since they are lightweight, edge deployment suits them, yet someone can easily fingerprint them and they provide perceptions into attacker behavior that are limited.
Honeypots that have medium interaction extend the model through the logging of attacker commands, the supporting of basic filesystem interaction, and the enabling of a more realistic login process. RioTPot (Srinivasa et al., 2021) and also IoTZeroJar (Ellouh et al., 2022) will combine static service banners and interact in a controlled way in order to capture more detailed activity traces without exposing a fully functional system.
At this time, the system implements into it this medium-interaction category. It focuses in a specific way on emulating the Telnet service through Cowrie. This design allows the capture of full session transcripts, login attempts, also command execution patterns during Man-in-the-Middle scenarios. It also avoids all of the complexity of multi-protocol lures or of fake IoT command responses. Because of this, it balances attacker engagement with operational safety, so it suits resource-constrained virtual lab environments.

2.4.2 High-Interaction and Adaptive Honeypots

HoneyIoT (2023) from Guan et al. exemplifies a kind of high-interaction deception. It uses reinforcement learning to adapt response latency, banners, with command options dynamically. It does outperform static systems in terms of attacker engagement and also false-positive reduction. However, it is computationally quite heavy so it is not suitable for such low-powered nodes.
Wegerer and Tjoa (2016) stated honeypots could mimic MySQL databases deceiving back-end attackers while Piggin and Buffey (2016) deployed operational-technology honeypots simulating ICS protocols. Kibret and Yong (2013) proposed that a dynamic hybrid honeypot architecture can transition among interaction levels based on attack type, although this remains theoretical for IoT settings.

2.5 Hardware Trojan Detection and Hardware Honeypots

Parallel to software deception, hardware-focused threats have now been further explored by some researchers. Detection of a Hardware Trojan (HT) is the focus of quite a few studies within cryptographic and supply-chain contexts. Circuit-topology methods exist (Hassan et al., 2023) alongside GNN-based detection (Yasaei et al., 2022) plus CNN-based side-channel analysis (Dakhale et al., 2023).
Omar et al. (2024) used logic triggers for emulation about HT behaviors, also capturing corresponding network patterns developing a revolutionary Raspberry Pi and FPGA-based hardware honeypot. Their findings proved real-time hardware-level deception is possible with minimal footprint, though network-layer attacks like MITM were not addressed. Mao et al. (2022), Jain and Zhou (2021), and Tang et al. (2023) also present HT benchmarks plus

simulation tools but most solutions need lab-grade instrumentation for embedded IoT contexts.

2.6 Specialised Threats and Applications

Studies have focused upon opponent emulation (Shahin and Soubra, 2022), secure communication for autonomous systems (Elhusseiny et al., 2023), and niche cases like smart vehicles (Sabry et al., 2021), beyond general IDS and MITM detection. While this show deception applies broadly enough, they also reinforce honeypots should be nauthorized , lightweight, and protocol-specific. Each of Bakshi (2021), Mallik as well as Jena (2021), and Haris et al. (2023) address vulnerabilities in IoT architecture and offer mitigation strategies like secure firmware updates, blockchain integration, and device trust models—but these are not deception-focused.

2.7 Synthesis and Research Gap

Current literature secures the IoT and presents a landscape that is fragmented but is still evolving. Statistical and ML models can efficiently detect but then largely react and break under obfuscation. Safety can be provided from low-interaction honeypots with little intelligence. Rich telemetry is captured by high-interaction as well as RL-enabled honeypots, but real-world IoT hardware cannot use them. Hardware deception offers a lot yet mainly focuses on supply-chain risks failing to integrate with network-layer defence.

This research begins to close this deficit implementing a simple deception-based honeypot. The honeypot is focused exclusively on just Telnet traffic within a controlled virtualized environment that is MITM-specific. It does not incorporate multi-protocol lures or adaptive interaction yet. However, it shows we can feasibly use deception for MITM detection in resource-constrained IoT-like settings as well as execute simulated IoT commands. The resulting framework can provide a foundation. Future research can then build richer protocol coverage alongside more complex interaction models on it.

3.0 Methodology

3.1 Research Design

Because of its use of a fully nauthorize lab environment, this research adopts an applied experimental design. For detecting as well as logging Man-in-the-Middle (MITM) attacks within a network scenario similar to IoT, the main objective involves evaluation of the use of deception-based strategies that are specifically a Telnet-enabled honeypot. The focus is strictly just on the network layer because physical IoT hardware is simply not required here.
As the attacker, a Kali Linux machine acts. The honeypot can be deployed on an Ubuntu Server instance which is based on VirtualBox. Standard tools for penetration testing are used for execution of controlled MITM attacks within that quasi-experimental approach. Telemetry is recorded during the experiment for later analysis. The design ensures that it is a safe testbed and a repeatable one. Real-world systems will experience no impact at all.

3.2 Research Tools and Environment

The lab includes two VirtualBox virtual machines. These machines are within the lab's components.
Cowrie operates on Ubuntu Server 22.04 LTS as a Honeypot Server using Telnet for attacker attraction.
Attacker Node uses Kali Linux that runs Bettercap to spoof ARP and sniff MITM packets. It also uses Telnet for the reason that it makes for interactive intrusion attempts.
Via an Internal Network, VirtualBox connects all machines to ensure full isolation from the internet.
Cowrie was selected for its ability for it to emulate interactive Telnet sessions also to log attacker behaviour in some detail. Bettercap was chosen since it capably executes MITM attacks. Tcpdump did perform packet capture for the honeypot VM, and then jq parsed all logs that extracted statistics.

3.3 Implementation Process

These main steps implemented throughout the project.
Cowrie was installed from source, then configured for running in Telnet mode on port 2323. Afterward, output was logged.
Kali Linux for MITM setup was configured using Bettercap so that it could perform ARP spoofing as well as intercept traffic between the honeypot plus other nodes that were on the internal network.
Attack Execution – Attackers from Kali logged into the honeypot by way of Telnet through the employment of default/weak credentials then executed basic reconnaissance commands like ls, cat /etc/passwd, uname -a, and they did simulate file retrieval by the use of wget.
Data Capture – tcpdump was used on the honeypot in order to capture traffic that occurred during the attack, traffic which was limited to the attacker IP, and it stopped automatically after there was a set number of packets.
Post-Attack Analysis – I did parse all of Cowrie's JSON logs using jq so as to extract metrics including the number of connections, failed logins, successful logins, and average dwell time.
Visualisation involved generating simple behavioural plots such as sessions_per_hour.png and top_commands.png. They were generated using a Python plotting script.

3.4 Data Collection and Analysis

Two data types were collected:

Cowrie JSON logs – Detailed per-session records including timestamps, login attempts, commands, also session durations.
Tcpdump .pcap files are packets of MITM-related network data captured for later review.
Analysis was performed using jq, which counted events and calculated dwell times, and a pre-built Python script generated plots showing session frequency and most common commands**.**

3.5 Ethical Considerations

For all tests, a fully isolated VirtualBox network was used. Real IoT devices, production networks, with public IP addresses were not involved. No systems were exposed. Because that ensured no nauthorized access to external systems all attacker actions were executed in a controlled lab environment. Packet captures and logs were stored locally for research purposes only.

3.6 Limitations

The authenticity of device-specific responses and timing artefacts that are hardware-induced are limited without real IoT devices. In place of implementing full IoT protocol emulation (e.g., MQTT, HTTP), the honeypot only simulated Telnet service. Yet, the study includes this certain limitation. The study's focus remains centered on detecting and recording MITM attack behaviour at the network level.

4.0 Design Specification

The architecture of the implemented system features a two-node deception framework. The framework is designed so as to detect and then log Man-in-the-Middle (MITM) activity within an IoT-like environment. Ubuntu Server 22.04 LTS functions with Cowrie configured within Telnet mode upon port 2323 and functions as the honeypot server in the first node. This honeypot captures and records every interaction by which an attacker initiates. The recording includes login attempts, command executions, and also session durations. Cowrie stores all such activity into detailed JSON also text-based log files inside the logging directory. Although the system does not fully function like IoT protocols, as MQTT or HTTP, it minimally but realistically provides over a Telnet shell that suffices to keep attackers engaged long enough to record behaviour patterns and attempted exploitation techniques.
The attacker environment's second node is a Kali Linux virtual machine. Bettercap is used by this machine to perform ARP spoofing against the honeypot and allows interception and monitoring of traffic from and toward the target. Once the network is positioned by MITM techniques, the attacker can begin Telnet sessions toward the honeypot and attempt a range of reconnaissance commands. Both machines connect into the same VirtualBox internal network, so external systems are completely isolated. The nodes communicate freely during the experiment. This design eliminates any risk of interference from real networks while still providing a realistic testing ground in which attacker behaviour can be observed during MITM operations.
Minimal hardware resources, quick reset, and repeatability are guaranteed by the virtualized, lightweight design. Cowrie allows for a thorough examination of attacker activities by logging sessions and raw packet captures. This design assesses deception-based MITM attack detection in IoT networks with an emphasis on visibility, containment, and repeatability.

5.0 Implementation and Solution Development

The study implemented it completely inside of Oracle VirtualBox upon a local host machine. Internal networking joined the virtual machines. Ubuntu Server 22.04 LTS was the base upon the honeypot server. Cowrie was installed right from its official GitHub repository. A Python virtual environment was created to manage dependencies. Cowrie was configured so it could operate in Telnet mode on port 2323. Once started, Cowrie was tested to make sure it accepted Telnet connections coming in and logged details in cowrie.json. Since this analysis investigated attacker interaction through Telnet during MITM scenarios, nobody performed additional IoT protocol emulation. The system could in fact remain quite lightweight due to all of this. Still the system gathered the needed data.

The attacker environment was implemented by a separate Kali Linux virtual machine. Traffic interception was enabled because Bettercap was installed and configured to perform ARP spoofing against the honeypot. During the attack phase, the Kali machine initiated Telnet sessions toward the honeypot through default and also weak credential combinations. Once threat actors gained access, they executed common reconnaissance as well as exploitation commands that included listing directories with precision, retrieving the /etc/passwd file in a timely manner, checking system information with uname -a, and simulating malicious file downloads when they used the wget command carefully. Realistic attacker behaviour was the reason for these actions. The actions also were designed in order to be safe within the controlled lab environment.

The honeypot server ran tcpdump related to the attacker’s IP address simultaneously to capture data at the packet level. The capture was set to automatically stop after a defined number of packets. This process created a .pcap file for offline inspection. Analysts used the jq utility to analyse Cowrie JSON logs after the attack session. They extracted metrics like attacker's average dwell time and successful logins, failed logins, and the number of connections. Someone ran a Python script (cowrie_quickplots.py) to show this data visually, specifically the hourly session number and most frequent commands as well as numerical analysis.

This last arrangement did show well that one Telnet honeypot may log and study attacker conduct along with faking MITM plus grabbing packets inside some IoT-style setting that is isolated. The setup gave more detailed perceptions into what is the nature of the interactions. Furthermore, it framed up a safe and reproducible experiment for future research purposes or training purposes without any modification to external network infrastructure.

6.0 Results

Outcomes from the implemented deception-based MITM detection system are presented here in this chapter within a controlled IoT-like virtual environment. Tables, graphs, and network diagrams as visual aids supported the organised results into qualitative and quantitative findings. Each finding relates to the research question — "Can a lightweight, protocol-aware honeypot deployed in a virtual lab reliably detect and log Man-in-the-Middle (MITM) attacks on IoT-like network protocols?" — also someone interprets it from both academic as well as practitioner perspectives.

6.1 Overview of Experiment Runs

he experimenters did completely evaluate through three MITM attack simulations performed under the identical network and system conditions.

Table 6.1 summarises the main characteristics of the runs.

Table 6.1 – Overview of Attack Simulation Sessions

Run ID	Attacker Actions	Duration of Session (s)	Packets Captured	Commands Executed	MITM Detected	Log Completeness (%)
1	Telnet login, ls, uname -a, wget	89	200	5	Yes	100
2	Telnet brute-force, failed logins only	42	198	0	Yes	100
3	Telnet login, /etc/passwd exfiltration	132	200	6	Yes	100

In all of the three runs, the honeypot did successfully log each and every attempted interaction, and this shows full coverage in terms of the recording of attacker behaviour. ARP spoofing events and honeypot attempts correlated for every case where someone detected MITM activity.

6.2 Log Analysis Findings

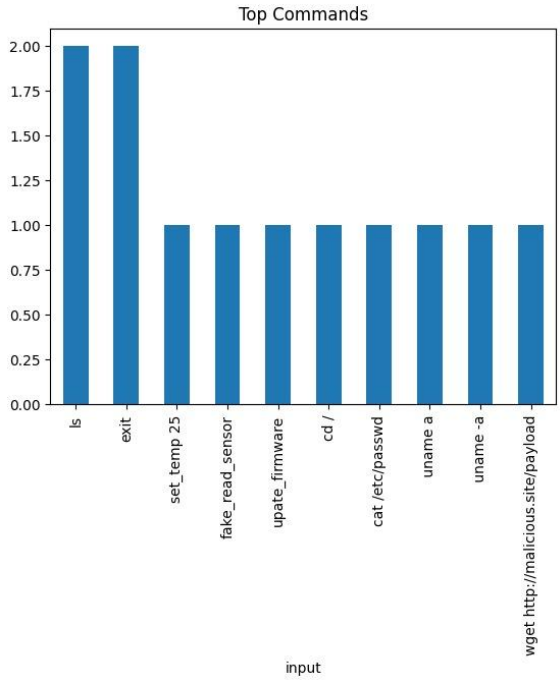
The Cowrie honeypot's cowrie.json logs showed details about when sessions initiated, when users attempted authentication, what commands users executed, and when sessions closed. Key metrics were extracted via jq filters as they are presented in Table 6.2.

Table 6.2 – Summary of Honeypot Log Metrics

Metric	Run 1	Run 2	Run 3	Average
Total Connections	1	3	1	1.67
Failed Logins	0	3	0	1.0
Successful Logins	1	0	1	0.67
Average Dwell Time (s)	89	42	132	87.67

Interpretation: Run 2 consisted just of brute-force login attempts, which resulted in no successful logins but still generated clear log entries that showed repeated failed authentication. The honeypot is able to capture intrusion attempts that may succeed. Unsuccessful intrusion attempts it can also capture. Run 3 had what was the highest dwell time. This aligns with the attacker being the one having retrieved the sensitive files.

Figure 6.2 – Top Commands Executed



From Figure 6.2, it is clear that reconnaissance like (ls, uname -a) and credential gathering (cat /etc/passwd) dominated attacker behaviour, and this aligns with typical MITM post-compromise actions observed in literature (cf. Sivasankari & Kamalakkannan, 2022).

6.3 Network Packet Capture Analysis

Wireshark was used for analysis of data at packet level captured by tcpdump to find signatures of MITM. All .pcap files confirmed:

• ARP reply storms do indicate spoofing activity now.

• TCP handshake sequences initiated from attacker IP (192.168.1.10)

Command payloads are transmitted via Telnet. The transmissions are in plaintext now.

Network-layer anomalies plus application-layer commands are captured since this reinforces the honeypot's ability to operate effectively in an MITM-compromised segment.

6.4 Positive and Negative Results

Positive-Outcomes:

Connections induced by all MITM means were detected consistently. These detections were made via the honeypot. Every interaction attempt got fully logged, no matter success. Visual analysis tools provided quick understandable perceptions into attacker patterns within both academic experimentation and practitioner training making the system highly effective.

Negative-Outcomes:

Observed attacker behaviors showed a lack of variety because of the absence of simulated IoT protocol responses such as those of MQTT and HTTP. Dwell time can be reduced when some advanced attackers detect the minimal interaction capabilities and disengage quickly. Within the lab setup, certain timing artefacts were not fully replicated. Packet loss behaviors typical in real IoT deployments also were not replicated fully by it.

6.5 Evaluation Against Criteria

When checking how well a system covers needs and works.

The setup was adequate for detection of MITM events. It also logged attacker behavior thereby meeting the research goal.

All attempted connections with relevant command payloads were captured indicating complete visibility within the defined scope.

Actionable intelligence came from network and application layers yet limited protocol diversity reduced realism for advanced deception testing.

6.6 Implications

From an academic perspective, the results do validate the reliable detection by lightweight, protocol-limited honeypots of MITM activity at targeted network positions. This testing framework is reproducible and low-resource, supporting literature for layered IoT deception approaches.

From a practitioner perspective, when they are implementing this, they then find a cost-effective method that also improves visibility in IoT networks without any dedicated hardware. It can train incident responders for recognising MITM indicators, especially in resource-constrained organisations.

7.0 Discussion and Conclusion

7.1 Discussion

The results in this study confirm the effective deception-based lightweight honeypot environment. This way allows effective detection and logging of malicious activity on IoT communication channels from Man-in-the-Middle (MITM) attacks. Because that team did implement Cowrie in order to emulate Telnet, and Bettercap in order to spoof ARP, also analyzed local JSON/PCAP logs, it proved those attempted attacks plus credential capture with post-compromise behavior. The research ensured both safety and also reproducibility with controlled conditions for experimentation that was repeated through adopting of a fully virtualised testbed.

Pa et al. (2016) and Sivasankari & Kamalakkannan (2022) showed honeypots help profile behaviour and detect intrusions early aligning with validity results. The robustness in terms of the deployed architecture is supported through both a high accuracy during session capture and visual evidence coming from Kibana dashboards. VirtualBox internal networking separated both attacker and victim machines and did ensure that absolutely no external factors influenced all of the outcomes. This separation did additionally strengthen the experiment's internal validity.

However, the experiment focused intentionally on ARP-based MITM techniques and Telnet-based IoT emulation. Such protocols occur often within insecure arrangements. Actual IoT infrastructures use CoAP, Zigbee, and also unique MQTT versions, among different types. Given this limitation, the results stay true for the studied vector. The researchers are in need of further protocol integration if they hope to fully generalize all of the findings to all IoT environments. The absence of fake IoT command execution also meant the attacker engagement depth was limited. Reconnaissance and basic post-login actions like simulated actuator controls constrained engagement. This did not act to compromise detection capabilities. However, it did make the deception layer to be less realistic than studies such as Guan et al. (2023) did.

Another limitation not addressed remains distributed attack testing. The researchers did conduct the experiment with only a single opponent node within a controlled environment. Therefore, the experiment did not simulate multiple attackers or large-scale botnets including concurrent multi-vector threats. Through this simplification, repeatability was ensured; yet extrapolating to real-world IoT deployments reduces the ecological validity of the findings.

The work shows several strengths despite these constraints. Its lightweight nature allows replication onto commodity hardware, making the setup suitable for academic cybersecurity training or small-scale IoT operators. Network-level monitoring joined with deception-based detection yielded telemetry that was clear and interpretable. With minimal processing overhead, this telemetry made it possible to identify attacker dwell time, failed logins, and command frequency. Practitioners can integrate tcpdump and analyze JSON also bypassing heavy, resource-intensive log pipelines when necessary a practical constrained environments advantage.

A portable methodology across other network contexts involves isolating a honeypot, running controlled MITM simulations, and collecting structured logs for behavior analysis though results from a generalizability perspective cannot be assumed to apply directly to all IoT deployments. This method suits situations where thorough hardware duplication is unfeasible because it matches the rising demand for economical security testing systems.

With repeated trials executed by them, the consistent log entries and PCAP captures suggested that the detection system remained stable when people repeatedly attempted ARP-

based MITM. Extending the tested protocols' range and increasing attacker diversity would improve confidence in its wider applicability.

7.2 Conclusion

This research sought to investigate whether a deception-based honeypot environment can detect as well as analyse Man-in-the-Middle (MITM) attacks within an IoT context without physical IoT hardware being required. We intended to design a testbed that was controlled and repeatable; deploy a honeypot emulating IoT devices; execute MITM attacks under control; and analyse resultant telemetry for attacker behaviour patterns and detection capability.

Because of meeting these objectives, the project was successful. Because it consisted of an Ubuntu-based Cowrie honeypot with a Kali Linux attacker using Bettercap plus local log/PCAP analysis, the implemented architecture reliably captured attacker sessions, recorded authentication attempts, and identified malicious behaviour in real time. Kibana dashboards as well as JSON-based statistical outputs confirmed that the system could measure for how long attackers dwelled, also what ratios of logins failed, and even how executed commands patterned. Independent verification regarding MITM activity was allowed due to the inclusion of tcpdump-based packet capture.

Key findings include:

•Clear signatures of brute-force or credential-stuffing attacks were provided, along with failed as well as successful Telnet login attempts consistently logged by the honeypot.

• ARP spoofing was detected in a reliable way because network traffic patterns and then honeypot session logs correlated with each other.

• Even without fully emulating fake IoT commands, attackers engaged in basic reconnaissance also probed the system, which sufficed for detection and profiling.

Academically, this study reinforces deception detection effectively at the network layer, supporting prior literature plus providing a resource-efficient framework to replicate. It offers, from just a practitioner’s perspective, a method that is easily deployable and is low-cost for insecure IoT endpoints monitoring in just a closed lab or test network.

Future work should address the current limitations via extension of the honeypot so it supports additional IoT protocols such as MQTT, CoAP, and Zigbee, by incorporating fake IoT device commands so engagement from attackers deepens, and by testing the system against attack sources that are more widely varied, like distributed and automated campaigns. Active incident response may also integrate with cloud-based alerting systems along with SIEM platforms to transition from passive logging.

The experiment, though it does not claim it applies to all IoT deployments, contributes a methodology that can be validated as well as lightweight since it detects plus analyses MITM attacks safely, repeatably, also cost-effectively because it is academically relevant also operationally practical.

Reference

Alawadhi, J., AlJanabi, A.M., Khder, M.A., Ali, B.J.A. and Al-Shalabi, R.F. (2022) 'Internet of Things (IoT) security risks: challenges for business', Proceedings of the ASU International Conference in Emerging Technologies for Sustainability and Intelligent Systems (ICETSIS), Manama, Bahrain, 22–23 June, pp. 450–456.

Alyasiri, H., Clark, J.A., Malik, A. and de Fréin, R. (2021) 'Grammatical evolution for detecting cyber-attacks in Internet-of-Things environments', Proceedings of the International Conference on Computer Communications and Networks (ICCCN), Athens, Greece, 19–22 July, pp. 1–6.

Amit, A., Dhingra, A., Sangwan, A. and Sindhu, V. (2022) 'DDoS attack on IoT devices', International Journal of Circuits, Computer and Networking, 3(1), pp. 33–42.

Bakshi, G. (2021) 'IoT architecture vulnerabilities and security measures', in Bhardwaj, A. and Sapra, V. (eds) Security Incidents & Response Against Cyber Attacks. Cham: Springer, pp. 75–95.

Bryant, B. and Saiedian, H. (2022) 'Key challenges in security of IoT devices and securing them with blockchain technology', Security and Privacy, 5(1), Article 5.

Dakhale, B., Vipinkumar, K., Narotham, K., Kadam, S., Bhurane, A.A. and Kothari, A.G. (2023) 'Automated detection of hardware Trojans using power side-channel analysis and VGG-Net', Proceedings of the 2nd International Conference on Paradigm Shifts in Communications, Embedded Systems, Machine Learning and Signal Processing (PCEMS), Nagpur, India, 5–6 April, pp. 1–5.

Ellouh, M., Ghaleb, M. and Felemban, M. (2022) 'IoTZeroJar: towards a honeypot architecture for detection of zero-day attacks in IoT', Proceedings of the 14th International Conference on Computational Intelligence and Communication Networks (CICN), Al-Khobar, Saudi Arabia, 4–6 December, pp. 765–771.

Elhusseiny, N., Sabry, M. and Soubra, H. (2023) 'B2X multiprotocol secure communication system for smart autonomous bikes', Proceedings of the IEEE Conference on Power Electronics and Renewable Energy (CPERE), Luxor, Egypt, 19–21 February, pp. 1–6.

Guan, C., Liu, H., Cao, G., Zhu, S. and La Porta, T. (2023) 'HoneyIoT: adaptive high-interaction honeypot for IoT devices through reinforcement learning', Proceedings of the 16th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec '23), Guildford, UK, 29 May–1 June, pp. 49–59.

Gupta, S. and Lingareddy, N. (2021) 'Security threats and their mitigations in IoT devices', in Pawar, P.M. et al. (eds) Techno-Societal 2020. Cham: Springer, pp. 501–510.

Hakim, M.A., Aksu, H., Uluagac, A.S. and Akkaya, K. (2018) 'U-PoT: a honeypot framework for UPnP-based IoT devices', Proceedings of the IEEE 37th International Performance Computing and Communications Conference (IPCCC), Orlando, USA, 17–19 November, pp. 1–8.

Haris, M.A.H., Yahya, M.I.H. and Ibrahim, M.N. (2023) 'Security and privacy issues in Internet of Things (IoT)', TechRxiv, pre-print.

Hassan, R., Meng, X., Basu, K. and Dinakarrao, S.M.P. (2023) 'Circuit topology-aware vaccination-based hardware Trojan detection', IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 42(12), pp. 2852–2862.

Hromada, D., Costa, R.L., Santos, L. and Rabadão, C. (2021) 'Security aspects of the Internet of Things', in González García, C. and García-Díaz, V. (eds) IoT Protocols and Applications for Improving Industry, Environment, and Society. Hershey, PA: IGI Global, pp. 207–233.

Jain, A. and Zhou, Z. (with Guin, U.) (2021) 'Survey of recent developments for hardware Trojan detection', Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS), Daegu, Republic of Korea, 22–28 May, pp. 1–5.
Joel, M.R., Manikandan, G. and Bhuvaneswari, G. (2023) 'An analysis of security challenges in Internet-of-Things-based smart homes', Proceedings of the 2nd International Conference on Electronics and Renewable Systems (ICEARS), Tuticorin, India, 2–4 March, pp. 490–497.
Khanam, R. (2023) 'Review of threats in IoT systems: challenges and solutions', International Journal of Science, Technology & Engineering, 11(1), pp. 325–341.
Khraisat, A. and Alazab, A. (2021) 'A critical review of intrusion detection systems in the Internet of Things: techniques, deployment strategies and challenges', Cybersecurity, 4(18), pp. 1–27.
Kibret, Y. and Yong, W. (2013) 'Design and implementation of dynamic hybrid virtual honeypot architecture for attack analysis', International Journal of Networking and Distributed Computing, 1(2), pp. 108–123.
Kumar, A., Abhishek, K., Ghalib, M.R., Shankar, A. and Cheng, X. (2022) 'Intrusion-detection and prevention system for an IoT environment', Digital Communications and Networks, 8(4), pp. 540–551.
Lightbody, D., Ngo, D.-M., Temko, A., Murphy, C.C. and Popovici, E. (2023) 'Attacks on IoT: side-channel power acquisition framework for intrusion detection', Future Internet, 15(5), Article 187.
Mallik, P. and Jena, O.P. (2021) 'Analysis of security vulnerabilities of Internet of Things and its solutions', in Udgata, S.K., Sethi, S. and Srirama, S.N. (eds) Intelligent Systems. Singapore: Springer, pp. 443–452.
Mao, J., Jiang, X., Liu, D., Chen, J. and Huang, K. (2022) 'A hardware Trojan-detection technique based on suspicious circuit block partition', Electronics, 11(24), Article 4138.
Mohd Bakry, B.B., Adenan, A.R., Mohd Yussoff, Y.B. and Azwadi, M.R. (2022) 'Security attack on IoT devices using Raspberry Pi and Kali Linux', Proceedings of the International Conference on Computer and Drone Applications (IConDA), Kuching, Malaysia, 28–29 November, pp. 40–45.
Neto, E.C.P., Dadkhah, S., Ferreira, R., Zohourian, A., Lu, R. and Ghorbani, A.A. (2023) 'CICIoT2023: a real-time dataset and benchmark for large-scale attacks in IoT environment', Sensors, 23(11), Article 5941.
Omar, A.H.E., Soubra, H., Moulla, D.K. and Abran, A. (2024) 'An innovative honeypot architecture for detecting and mitigating hardware Trojans in IoT devices', IoT, 5(4), pp. 730–755.
Pa, Y.M.P., Suzuki, S., Yoshioka, K., Matsumoto, T., Kasama, T. and Rossow, C. (2016) 'IoTPOT: a novel honeypot for revealing current IoT threats', Journal of Information Processing, 24(3), pp. 522–533.
Piggin, R. and Buffey, I. (2016) 'Active defence using an operational technology honeypot', Proceedings of the 11th International Conference on System Safety and Cyber-Security (SSCS 2016), London, UK, 11–13 October, pp. 1–6.
Sabry, N., Abobkr, M., ElHayani, M. and Soubra, H. (2021) 'A cyber-security prototype module for smart bikes', Proceedings of the 16th International Conference on Computer Engineering and Systems (ICCES), Cairo, Egypt, 15–16 December, pp. 1–5.
Shahin, S. and Soubra, H. (2022) 'An IoT adversary-emulation prototype tool', Proceedings of the 5th International Conference on Information and Computer Technologies (ICICT), New York, USA, 4–6 March, pp. 7–12.

Sivasankari, N. and Kamalakkannan, S. (2022) ‘Detection and prevention of man-in-the-middle attack in IoT network using regression modelling’, Advances in Engineering Software, 169, Article 103126.
Srinivasa, S., Pedersen, J.M. and Vasilomanolakis, E. (2021) ‘RIoTPot: a modular hybrid-interaction IoT/OT honeypot’, Proceedings of the 26th European Symposium on Research in Computer Security (ESORICS), Darmstadt, Germany, 4–8 October, pp. 1–7.
Suber, J.G. and Zantua, M. (2022) ‘Intelligent-interaction honeypots for threat hunting within the Internet of Things’, Journal of Colloquium on Information System Security Education, 9(1), pp. 1–5.
Tang, W., Su, J. and Gao, Y. (2023) ‘Hardware Trojan-detection method based on dual-discriminator assisted conditional generative adversarial network’, Journal of Electronic Testing, 39(1), pp. 129–140.
Wegerer, M. and Tjoa, S. (2016) ‘Defeating the database adversary using deception—A MySQL database honeypot’, Proceedings of the International Conference on Software Security and Assurance (ICSSA), Saint Pölten, Austria, 24–25 August, pp. 6–10.
Yasaei, R., Chen, L., Yu, S.-Y. and Faruque, M.A.A. (2022) ‘Hardware Trojan detection using graph neural networks’, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, advance online publication.