National College of Ireland

Configuration Manual

MSc Research Project
Programme Name

Sowmya Bhumireddi
Student ID: 23373563

School of Computing
National College of Ireland

Supervisor: Jawad Salahuddin

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Sowmya Bhumireddi
Student ID:	23373563
Programme:	MSc in Cybersecurity **Year:** 2025
Module:	Research (Practicum) / Internship
Lecturer:	Jawad Salahuddin
Submission Due Date:	11th December 2025
Project Title:	Building a Lightweight Rule-Based Anomaly Intrusion Detection System for Internet of Medical Things Networks
Word Count:	……1553 **Page Count:** ……7…….…….……

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: 11th December 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):	

Configuration Manual

Sowmya Bhumireddi
Student ID: 23373563

1 Introduction

1.1 Purpose

The configuration manual provides a complete guide to installing, configuring, and using the lightweight rule-based Intrusion Detection System (IDS) created as an extension to the architecture of the anomaly-based IDS suggested in the research paper Zachos et al. (2022). This configuration manual provides detailed instructions for setting up the software, tools, and environments used in the research project.

1.2 Scope

This configuration guide discusses the entire installation of a virtualized testbed of a lightweight rule-based Intrusion Detection System (IDS) based on Zachos et al. (2022). It involves the installation of VirtualBox, the creation of VMs, the deployment of Python-based MDA agents and CD engine, the running of attack scenarios, and data logging. It is confined to the virtual environment that approximates resource-constrained IoMT devices and lacks physical deployment, and is not integrated with real clinical systems.

2 Purpose, Scope and Prerequisites

2.1 Purpose

The configuration manual provides a complete guide to installing, configuring, and using the lightweight rule-based Intrusion Detection System (IDS) created as an extension to the architecture of the anomaly-based IDS suggested in the research paper Zachos et al. (2022). This configuration manual provides detailed instructions for setting up the software, tools, and environments used in the research project.

2.2 Scope

This configuration manual is intended for researchers and cybersecurity professionals who aim to quantitatively analyze cybersecurity threats and the effectiveness of mitigation measures. It covers data collection methods, analysis techniques, and interpretation of results.

2.3 Prerequisites

16 GB RAM and 50 GB of free disk space to use in a laptop/desktop so that Oracle VirtualBox can run.

3 System Requirements and Research Setup

3.1 Hardware Requirements

- Host Machine: This should be any laptop or desktop with a minimum RAM of 8 GB, a processor of Intel i5/i7 or AMD equivalent, with 20 GB of free disk space.
- Virtual Machines: Oracle VirtualBox is used to simulate the IoMT devices and the gateway using the testbed. Minimum allocation per VM:
    - Gateway: 4GB RAM, 2 CPU Cores
    - IoMT Devices: 1-2 GB RAM, 1-2 CPU cores each.

3.2 Software Requirements

- Virtualization platform: Oracle VirtualBox 7.0 or above (free, open-source).
- OS: Ubuntu 22.04 LTS Desktop (64-bit) ISO, as all the VMs used this OS
- Programming Language: Python Version 3.8 or greater
- Python Libraries: psutil (data collection metrics), plotly (data visualization dashboard), pandas (data management).
- Attack tools: stress-ng (CPU/memory stress), hping3 (network flood) stress-ng (CPU/memory stress), hping3 (network flood)

3.3 Testbed Creation

- Install Oracle VirtualBox 7.0+.
- Make host-only vboxnet0 (IPv4: 192.168.56.1, mask 255.255.255.0, no DHCP).
- Download Ubuntu 22.04 LTS ISO.
- Make Gateway VM (4GB RAM, 2cores, host-only adapter).
- Install Ubuntu (minimal), 192.168.56.1 is the static IP, hostname is IoMT-Gateway.
- Copy the clone to 2 IoMT VMs, allocate the IPs 10-13, and hostnames.
- In each VM, run the following command "*sudo apt update && sudo apt install python3-pip stress-ng hping3 -y && pip3 install psutil pandas plotly.* "
- Test connectivity: ping 192.168.56.1.

3.4 Research Objectives

- Replicate the MDA element of Zachos et al. (2022).
- Install and configure a full-fledged rule-based CD engine.
- Establish a testbed to test against typical IoMT attacks.
- Measure accuracy, latency and overhead by collecting and analyzing data.
- Repeatable configuration of future extensions.

3.5 Research Questions

- What can be set in the incomplete CD part of Zachos et al. (2022) to be a lightweight rule-based detection?
- Which thresholds and rules are effective to real-time detect on sub-optimal settings?
- How accurately and how many overheads does the configured system withstand simulated attacks?

3.6 Research Scope and Limitations

- Scope: Complete set-up of a virtual IoMT IDS testbed, rule-based detection, visualization in the form of a dashboard, and attack simulation.
- Limitations All constraints of physical devices (e.g., battery life) might not be completely replicated in a virtualized environment. Attacks that cause self-stressing CPUs can also have a temporary impact on the responsiveness of agents, which can be alleviated by restricting stressed cores. None of the encryption or authentication is for data transmission (recommended future improvement).

4 Methodology

4.1 Research Design

The testbed was constructed manually with the help of VirtualBox in order to provide isolation and reproducibility. It is also based on Zachos et al. (2022) but with the exact (10-second sampling) data collection and custom rule-based detection since literature has indicated that ML overhead is inappropriate when it comes to IoMT (Hameed et al., 2021).

4.2 Data Collection Techniques

Primary data were gathered through psutil measures relayed as JSON packets on MDA agents to the gateway. All packets were logged for analysis.

4.3 Variables and Measurement Instruments

- Variables: CPU %, memory %, processes, active connections (independent); detection alerts (dependent).
- Measurement Tools: psutil for metrics, Python socket for transmission, Plotly for visualization.

4.4 Data Analysis Techniques

- Baseline and thresholds, descriptive statistics (mean, standard deviation).
- Detection rate = number of true positives/ number of attacks.
- Latency = difference between the timestamp of the first alert and the start of the attack.

4.5 Validity and Reliability

The validity of the direct reproduction of Zachos et al. architecture. Reliability: 3 repetitions per attack and therefore equal sampling.

4.6 Ethical Considerations

All testing must be conducted in a remote virtual network. No actual medical information or equipment was utilized. The attacks were ethically simulated research attacks. 4 Data Collection and Analysis.

5 Data Collection and Analysis

5.1 Data Collection Process

- Step 1: Configure host only network and VMs with fixed IPs.
- Step 2: Install mda_agent.py on the IoMT devices - initiation of the metrics collection time frame (interval of 10 seconds).
- Step 3: Run gateway_ids.py. It runs and starts logging by listening on port 9999.
- Step 4: Sequentially make an attack and give 60-90 seconds rest.
- Step 5: Store output using tee to the output of ids.log.

5.2 Data Analysis Procedure

- Baseline (15 min normal traffic): Mean/SD of adaptive rules.
- Attack stage: Log initial alert time and metric spikes.
- Post-processing: Calculate detection rate, latency (mean/standard deviation), FPR (false alerts/normal packets).

5.3 Interpretation of Results

It was found to obtain 100 % low latency and overhead. CPU/process/attacks (sustained alert); memory/network instant. The results affirm the fact that a rule-based approach is lightweight and effective.

5.4 Troubleshooting Problems

- If no information on the gateway, run the command "*sudo ufw open 9999"* and test it with the help of the command "*ss -tuln grep 9999"*.
- If the dashboard is blank: Kill all the browser windows, restart the gateway script.
- If agent is not sending: Check ps aux grep MDA agent,
- Restart nohup if the attack shows no alert:
- To prevent starvation of agents, increase the number of stressed cores (e.g. --cpu 2) to the minimum.
- If there is a high VM lag: Reduce to 2–3 devices.

6 Conclusions and Recommendations

6.1 Summary of study

The manual is capable of configuring a full IoMT IDS that is a continuation of Zachos et al. (2022) by accomplishing all goals with every step that can be replicated.

6.2 Recommendations

- To researchers: This installation may serve as a reference for comparing ML with rule-based IDS.
- To students: Extend with encryption or more rules.

6.3 Future Work

Connection to physical IoMT devices, incorporate TLS to transmit securely, combine federated learning to achieve privacy-preserving detection and compare against new threats such as ransomware attacks in medical devices.

References

Hameed, S.S., Selamat, A., Latiff, L.A., Razak, S.A., Ondrej Krejcar, Fujita, H., Nazir, M. and Sigeru Omatu (2021). A Hybrid Lightweight System for Early Attack Detection in the IoMT Fog. *Sensors*, 21(24), pp.8289–8289. doi:https://doi.org/10.3390/s21248289.

Zachos, G., Mantas, G., Essop, I., Kyriakos Porfyrakis, Ribeiro, J.C. and Rodriguez, J. (2022). Prototyping an Anomaly-Based Intrusion Detection System for Internet of Medical Things Networks. *Greenwich Academic Literature Archive (University of Greenwich)*, pp.179–183. doi:https://doi.org/10.1109/camad55695.2022.9966912.