Generating Datasets for Anomaly-Based Intrusion Detection Systems in IoT Networks

Ismael Ahmad Essop

A thesis submitted in partial fulfilment of the
requirements of the University of Greenwich
for the degree of *Doctor of Philosophy*

School of Engineering
Faculty of Engineering and Science
University of Greenwich

December 2021

This page intentionally left blank.

DECLARATION

I certify that the work contained in this thesis, or any part of it, has not been accepted in substance for any previous degree awarded to me or any other person, and is not concurrently being submitted for any other degree other than that of Doctor of Philosophy which has been studied at the University of Greenwich, London, UK.

I also declare that the work contained in this thesis is the result of my own investigations, except where otherwise identified and acknowledged by references. I further declare that no aspects of the contents of this thesis are the outcome of any form of research misconduct.

I declare any personal, sensitive or confidential information/data has been removed or participants have been anonymised. I further declare that where any questionnaires, survey answers or other qualitative responses of participants are recorded/included in the appendices, all personal information has been removed or anonymised. Where University forms (such as those from the Research Ethics Committee) have been included in appendices, all handwritten/scanned signatures have been removed.

Student Name: Ismael Ahmad Essop

Student Signature:

Date: 21/12/2021

First Supervisor's Name: Dr Georgios Mantas

First Supervisor's Signature:

Date: 21/12/2021

This page intentionally left blank.

ACKNOWLEDGEMENTS

Throughout this journey and the writing of this dissertation I have received a great deal of support and help from so many people. Although I am allowed to assume the sole authorship of this dissertation, the journey to complete this work has been a richly collaborative effort. It is first and foremost the fruit of working alongside my supervisors, mentors, colleagues, friends, and family.

I want to begin by expressing my immense gratitude towards my Ph.D. supervisor, Dr Georgios Mantas, for his mentorship and scholarly guidance. He is not just a true leader in the field, but a wonderful human being who has had a significant impact on me as an individual and as a researcher. His continuous support and unwavering belief in me encouraged me to undertake an ambitious project with much less trepidation. His dedication to cultivating an impactful research effort and his sense of mission in pushing the boundary of knowledge acquisition has positively impacted me. I'd also like to thank Dr Hieu Chi Le and Professor James Gao for their generous feedback when called upon and allowing me to focus on my research throughout my Ph.D., which has been one of the best and most productive periods of my life. I would like to express my sincere thanks to a number of researchers for their ideas, interests, and constructive criticisms, namely Dr Jose Ribeiro, George Zachos, Maria Papaioannou, and Professor Jonathan Rodriguez.

I am also indebted to Professor Predrag Rapajic who inspired me to keep trudging the difficult road of research.

I am forever indebted to my mother for her sacrifices in my life, my brother Salim for helping with financing my undergraduate and master's studies, my sister Zaynah for her continuous encouragement and friendship.

I feel extremely blessed to have as my best friend and wife Aliyah, who always provides me with unconditional understanding, love, and support. I also wish to express my enduring love to my sons, Muhammad for his forbearance, Ali for his wits, and Hamza for his determined nature. You can now have your father back!

Last, but not least, I dedicate this work to my late father, Idris, who never managed to complete his primary education, but instilled in me such spiritual and moral values, that have inspired me to surmount the numerous challenges that life throws at you.

This page intentionally left blank.

ABSTRACT

Over the past few years, we have witnessed the emergence of Internet of Things (IoT) networks that bring significant benefits to citizens, society, and industry. However, their heterogeneous and resource-constrained nature makes them vulnerable to a wide range of threats and an attractive target to attackers with a wide spectrum of motivations ranging from criminal intents, aimed at financial gain, to industrial espionage and cyber-sabotage. Consequently, security solutions protecting IoT networks from attackers are critical for the acceptance and wide adoption of such networks in the coming years. Nevertheless, the high resource requirements of conventional security mechanisms cannot be afforded by (i) the resource-constrained IoT nodes and/or (ii) the constrained environment in which the IoT nodes are deployed. Therefore, there is an urgent need for developing novel security mechanisms to address the pressing security challenges of IoT networks in an effective and efficient manner, taking into consideration their resource-constrained inherent limitations, before they gain the trust of all involved stakeholders and reach their full potential in the IoT market. Toward this direction, considerable research efforts have recently been put into the design and development of novel Anomaly-based Intrusion Detection Systems (AIDSs), tailored to the resource-constrained characteristics of IoT networks, because of their ability to detect not only known but also new, zero-day attacks, in IoT networks. However, although the concept of IoT AIDSs is promising, it cannot be materialised before the significant gap of the scarcity of benchmark datasets for training and evaluating Machine Learning (ML) models for IoT AIDSs is addressed. In fact, the current scarcity of benchmark IoT datasets constitutes a significant research gap that should be addressed in order to enable the development of more accurate and efficient IoT AIDSs whose effectiveness is evaluated based on their performance to successfully detect IoT attacks that is a process reliant on up-to-date, representative and well-structured IoT-specific benchmark datasets that until now have been missing. Therefore, contribution to filling this research gap is the main target of this thesis. In particular, the focus of this thesis is on the generation of new labelled IoT datasets that will be publicly available to the research community and include the following required information so as to be considered as benchmark IoT datasets for training and evaluating ML models for IoT AIDSs: (a) information reflecting multiple benign and attack scenarios from current IoT network environments, (b) sensor measurement data, (c) network-related information (e.g., packet-level information) from IoT networks, and (d) information related to the behaviour of the IoT devices deployed within IoT networks.

This page intentionally left blank.

Table of Contents

List of Figures

This page intentionally left blank.

List of Tables

This page intentionally left blank.

Key Acronyms

6LoWPAN	IPv6 over Low-power Wireless Personal Area Networks
AdaBoost	Adaptive Boosting
AIDS	Anomaly-based Intrusion Detection System
ARP	Address Resolution Protocol
CIA	Confidentiality, Integrity and Authenticity
CPU	Central Processing Unit
CSS	Cross-site Scripting
CSV	Comma Separated Values
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name Server
DoS/DDoS	Denial of Service/Distributed Denial of Service
DT	Decision Tree
EL	Ensemble Learning
ENISA	European Union Agency for Cybersecurity
FN	False Negative
FP	False Positive
FTP	File Transfer Protocol
GPS	Global Positioning System
HTTPS	Hypertext Transfer Protocol Secure
ICMPv6	Internet Control Message Protocol version 6
IDS	Intrusion Detection System
IIoT	Industrial Internet of Things
IoT	Internet of Things
IPv6	Internet Protocol Version 6
IT	Information technology
ITU	International Telecommunication Union
KNN	K-nearest Neighbour
LPM	Low Power Mode
LR	Logistic Regression
MAP	Maximum A Posteriori
MI	Mutual Information
MITM	Man-In-The-Middle

mJ	Millijoules
ML	Machine Learning
MQTT	Message Queuing Telemetry Transport
MS	Milliseconds
NB	Naïve Bayes
NFV	Network function virtualization
NTP	Network Time Protocol
OS	Operating System
OT	Operational technology
OWASP	Open Web Application Security Project
PCAP	Packet Capture
RBD	Radial Basis Function
RF	Random Forest
RPL	Routing Protocol for Low-Power and Lossy Networks
RX	Receiving Feature
SDN	Software-Defined Network
SNMP	Simple Network Management Protocol
SSDP	Simple Service Discovery Protocol
SVM	Support Vector Machines
TB	Terabyte
TCP/IP	Transfer Control Protocol/Internet Protocol
TN	True Negative
TP	True Positive
TP-Link	Twisted Pair Link
TX	Transmission Feature
UDP	User Datagram Protocol
USB	Universal Serial Bus
VMs	Virtual Machines
WiFi	Wireless Fidelity
WLANs	Wireless Local Area Networks
WSNs	Wireless Sensor Networks

CHAPTER 1 Introduction

1.1 Motivation

Despite the significant benefits that the Internet of Things (IoT) networks bring to citizens, society, and industry, the fact that these networks incorporate a wide range of different communication technologies (e.g., WLANs, Bluetooth, and Zigbee) and types of nodes/devices (e.g., temperature/humidity sensors), which are vulnerable to various types of security threats, raises many security and privacy challenges in IoT-based systems [1], [2], [3], [4]. For instance, attackers may compromise IoT networks to manipulate sensing data (e.g., by injecting fake data) and cause malfunction to the IoT-based systems that rely on the compromised IoT networks. It is worthwhile mentioning that IoT networks can become an attractive target to attackers with a wide spectrum of motivations ranging from criminal intents, aimed at financial gain, to industrial espionage and cyber-sabotage [5], [6], [7]. Consequently, security solutions protecting IoT networks from attackers are critical for the acceptance and wide adoption of such networks in the coming years. Nevertheless, the high resource requirements of conventional security mechanisms cannot be afforded by (i) the resource-constrained IoT nodes (e.g., sensors) with limited processing power, storage capacity, and battery life; and/or (ii) the constrained environment in which the IoT nodes are deployed and interconnected using lightweight communication protocols [1], [8]. Therefore, there is an urgent need for developing novel security mechanisms to address the pressing security challenges of IoT networks with reasonable cost in terms of processing and energy, taking into consideration their resource-constrained inherent limitations, before they gain the trust of all involved stakeholders and reach their full potential in the IoT market [2], [3], [5], [9].

Towards this direction, considerable research efforts have recently been put into the design and development of novel Anomaly-based Intrusion Detection Systems (AIDSs), tailored to the resource-constrained characteristics of IoT networks, because of their ability to detect not only known but also new, zero-day attacks, in IoT networks [10], [11], [12], [13]. However, although the concept of IoT AIDSs is promising, it cannot be materialised before the **significant gap of the scarcity of benchmark datasets** for training and evaluating Machine Learning models for IoT AIDSs is addressed [14], [15]. In fact, the current scarcity of benchmark IoT datasets constitutes a significant research gap that should be addressed in order to enable the development of more accurate and efficient IoT AIDSs whose effectiveness is evaluated based on their performance to successfully detect IoT attacks that is a process reliant on **up-to-date**, **representative** and **well-structured IoT-specific benchmark datasets** that until now have been missing.

1.2 Research Challenges

Although several datasets, such as KDDCUP99 [16], NSL-KDD [17], UNSW-NB15 [18], and CICD2017 [19], have been created over the past two decades for evaluation purposes of network-based Intrusion Detection Systems (IDSs), they do not include any specific characteristics of IoT networks as these datasets do not contain sensors' reading data or any IoT network traffic [14], [13]. To respond to this major issue, few efforts focused on the generation of IoT-specific datasets have also been seen in the literature recently. Yet, they are characterised by some limitations in terms of the IoT-specific information they include. For instance the datasets proposed in [20] and [21] are IoT-specific datasets but they lack of events reflecting attack scenarios. To address this limitation, the

IoT-specific and network-related datasets proposed in [22] and [23] contain events reflecting attack scenarios, however, they do not cover a diverse set of attack scenarios and do not include sensors' reading data or information related to the behaviour of the IoT devices (e.g., sensors/actuators) within the network. Therefore, these IoT datasets can mainly be used for detecting only a limited number of network-based attacks against IoT networks as they do not contain adequate information for detecting a wide range of network-based attacks and/or attacks that manipulate sensor measurement data or compromise IoT devices within the IoT network.

Consequently, there is an urgent need for comprehensive IoT-specific datasets containing not only **network-related information** (e.g., packet-level information) but also **information reflecting multiple benign and attack scenarios** from current IoT network environments, **sensor measurement data**, and **information related to the behaviour of the IoT devices** deployed within the IoT network for efficient and effective training and evaluation of AIDSs suitable for IoT networks. Towards this direction, the recent work [14], has proposed, for the first time, to the best of our knowledge, a new dataset that includes events of a variety of IoT-related attacks and legitimate scenarios, IoT telemetry data collected from heterogeneous IoT data sources, network traffic of IoT network, and audit traces of operating systems [14].

Therefore, it is clear that more benchmark IoT datasets including the following required information: i) **information reflecting multiple benign and attack scenarios**, ii) **sensor measurement data**, iii) **network-related information**, and iv) **information related to the behaviour of the IoT devices** are essential to be generated and become publicly available to the research community so as to fill the significant research gap of the scarcity of benchmark IoT datasets that will enable the development of more accurate and efficient IoT AIDSs.

1.3 Scope of the Research

Contribution to filling the **significant gap of the scarcity of benchmark IoT datasets** for training and evaluating Machine Learning models for IoT AIDSs is the main target of this PhD research work. In particular, the scope of this PhD research work is the generation of new labelled IoT datasets that will be publicly available to the research community and include the following required information so as to be considered as benchmark IoT datasets:

a. information reflecting multiple benign and attack scenarios from current IoT network environments;

b. sensor measurement data;

c. network-related information (e.g., packet-level information) from IoT networks; and

d. information related to the behaviour of the IoT devices deployed within IoT networks.

It is worthwhile mentioning that the new labelled IoT datasets are generated by implementing various benign IoT network scenarios and IoT network attack scenarios in the Cooja simulator which is the companion network simulator of the open source Contiki Operating System (OS) that is one of the most popular OSs for resource constrained IoT devices [24], [25]. To the best of our knowledge, this is the first time that the Cooja simulator is used, in a systematic way, to generate benchmark IoT datasets. In addition, the implemented attack scenarios cover the following types of IoT network attacks which have not been considered in the datasets proposed in [14]: UDP flooding attack, blackhole attack, sinkhole attack, and sleep deprivation attack.

The new labelled IoT datasets generated by the Cooja simulator are not to be considered as a replacement of datasets captured from real IoT networks or real IoT testbeds, but instead to be considered as complementary datasets that will contribute to fill the current gap of the scarcity of benchmark datasets for training and evaluating Machine Learning models for IoT AIDSs. Furthermore, the generated datasets are analysed to select important raw features for the detection of anomalies as well as extract new features, more informative and non-redundant, based on the raw features. Finally, different Machine Learning (ML) algorithms for IoT AIDSs (e.g., Naïve Bayes, K-Nearest Neighbour, Random Forest, Logistic Regression, etc.) are applied to evaluate their performance on the generated malicious datasets and validate that the generated malicious datasets can be used for training and testing effectively ML algorithms for IoT AIDSs.

1.4 Thesis Contribution

The main contributions of this PhD research work lie in the following:

- Generation of a set of benign IoT datasets from a benign IoT network scenario implemented in the Cooja simulator. The generated IoT-specific information from the simulated scenario was captured from the Contiki plugin “powertrace” (i.e., features such as CPU consumption) and the Cooja tool “Radio messages” (i.e., network traffic features) to generate the “powertrace” dataset and the network traffic dataset, respectively, within csv files. The generated datasets constitute the benign IoT datasets for the simulated benign IoT network scenario. Furthermore, a detailed description of the approach proposed to generate the set of benign IoT datasets has also been provided and published in *Generating Datasets for Anomaly-based Intrusion Detection Systems in IoT and Industrial IoT Networks* [26]. This contribution is covered in Chapter 3. In addition, we generated a set of malicious datasets from the following attack scenarios implemented in the Cooja simulator: i) UDP flooding attack, ii) blackhole attack, iii) sinkhole attack, and iv) sleep deprivation attack. The generated IoT-specific information from the simulated attack scenarios was captured from the Contiki plugin “powertrace” and the Cooja tool “Radio messages” in order to generate the corresponding “powertrace” and network traffic datasets for each of the simulated attack scenarios within csv files. The generated datasets constitute the malicious IoT datasets for the simulated IoT attack scenarios. Moreover, a detailed description of the approach proposed to generate the set of the malicious IoT datasets has also been given. The description of the approach proposed to generate the set of the UDP flooding attack datasets has been published in *Generating Datasets for Anomaly-based Intrusion Detection Systems in IoT and Industrial IoT Networks* [26]. This contribution is covered in Chapter 4.
- Analysis of the malicious “powertrace” datasets to investigate whether their raw features can be important in the detection of anomalies in the network-level power profiling of low-power IoT devices (i.e., motes) due to UDP flooding attacks, blackhole attacks, sinkhole attacks, or sleep deprivation attacks. Based on the results and the observations in Section 5.2.1, the following 5 features have been identified as the most important for all malicious “powertrace” datasets: “transmit”, “cpu”, “lpm”, “listen”, and “idle_listen”. Furthermore, we extracted new features, more informative and non-redundant, based on the raw features of the generated benign and malicious “powertrace” datasets. To this end, the total energy consumption of each mote in an IoT

network was investigated in Section 5.2.2 as a valuable feature for training and evaluating IoT AIDSs. According to the observations and conclusions in Section 5.2.2, the total energy consumption of each mote in an IoT network can play a valuable role in anomaly-based intrusion detection for the following types of attacks in IoT networks: UDP flooding attack, blackhole attack, sinkhole attack, and sleep deprivation attack. Besides that, we extracted new features, more informative and non-redundant, based on the raw features of the generated benign and malicious network traffic datasets. The generated benign and malicious network traffic datasets were also analysed in Section 5.3.1 and the new feature that was extracted was the "RPL packets overhead". This new feature provides information about the number of RPL packets (per mote and total) transmitted over the total number of exchanged messages within the IoT network, indicating a blackhole or sinkhole attack when its value is high and a UDP flooding attack or sleep deprivation attack when its value is low. This contribution is covered in Chapter 5.

- Validation of the generated malicious "powertrace" datasets by applying the following most popular ML algorithms for IoT AIDS to evaluate their performance on the generated malicious datasets: naïve Bayes (NB), decision tree (DT), random forest (RF), logistic regression (LR), support vector machines (SVM), and k-nearest neighbour (KNN). Using five-fold cross validation, these algorithms were trained and tested over the same labelled dataset for each attack scenario. Furthermore, the traditional metrics of accuracy, precision, recall, and F1-score were used to evaluate the performance of the ML algorithms on the generated datasets. The evaluations results demonstrated that the RF, KNN, and DT algorithms presented very high values regarding accuracy (between 0.93 and 1.0) and outperform the other algorithms regarding precision, recall and F1-score for all malicious datasets. In particular, it is worthwhile mentioning that the RF, KNN, and DT algorithms achieved precision between 0.84 and 1.0 for the "udp-flood-pwrtrace_label.csv", "blackhole-pwrtrace_label.csv", and the "sleep_depr-pwrtrace_label.csv". In principle, the evaluations results demonstrated that the generated malicious datasets can be used for training and testing effectively ML algorithms for IoT AIDSs. This contribution is covered in Chapter 6.

This PhD research work has led to the following 5 peer reviewed publications: 3 journal papers, 1 book chapter, and 1 conference paper (https://www.gre.ac.uk/people/rep/faculty-of-engineering-and-science/ismael-essop).

Journal papers (3)

1. Zachos, Georgios, **Essop, Ismael**, Mantas, Georgios, Porfyrakis, Kyriakos, Ribeiro, Jose, Rodriguez, Jonathan (2021), An anomaly-based intrusion detection system for internet of medical things networks. Electronics, 10: 2562 (21) 2079-9292 (Online) (doi: https://doi.org/10.3390/electronics10212562).
2. **Essop, Ismael**, Ribeiro, José C., Papaioannou, Maria, Zachos, Georgios, Mantas, Georgios, Rodriguez, Jonathan (2021), Generating datasets for anomaly-based intrusion detection systems in IoT and industrial IoT networks. Sensors, 21: 1528 (4) 1424-8220 (Online) (doi: https://doi.org/10.3390/s21041528).
3. Papaioannou, Maria, Karageorgou, Marine, Mantas, Georgios, Sucasas, Victor, **Essop, Ismael**, Rodriguez, Jonathan, Lymberopoulos, Dimitrios (2020), A Survey on security threats and countermeasures in Internet of Medical Things (IoMT). Transactions on Emerging Telecommunications Technologies: e4049 ISSN: 2161-3915 (Print), (doi: https://doi.org/10.1002/ett.4049).

Book Chapters (1)

1. Karageorgou, Marina, Mantas, Georgios, **Essop, Ismael**, Rodriguez, J , Lymberopoulos, D (2020), Cybersecurity Attacks on Medical IoT Devices for Smart City Healthcare Services. In: Fadi Al-Turjman, Muhammad Imran (eds.), IOT Technologies in Smart-Cities: From Sensors to Big Data, Security and Trus. Institution of Engineering & Technology, pp. 171-187. ISBN: 9781785618697 (doi: https://doi.org/10.1049/PBCE128E).

Conference Papers (1)

1. Zachos, Georgios, **Essop, Ismael**, Mantas, Georgios, Kyriakos, Porfyrakis , Jose, Ribeiro , Jonathan, Rodriguez (2021), Generating IoT Edge Network Datasets based on the TON_IoT telemetry dataset. (1st) . pp. 1-6 . ISBN: 9781665417792ISSN: 2378-4865 (Print), 2378-4873 (Online) (doi: https://doi.org/10.1109/CAMAD52502.2021.9617799).

1.5 Organisation of the Thesis

The remaining Chapters of this thesis are organized as follows.

- **Chapter 2** gives a comprehensive overview of the four main pillars of this PhD research work: i) *Internet of Things (IoT)*, ii) *Machine Learning (ML) algorithms for anomaly-based intrusion detection in IoT networks*, iii) *evaluation metrics for the performance of ML algorithms*, and iv) *existing datasets for training and evaluation of anomaly-based intrusion detection in IoT networks*. Therefore, the Chapter starts with an overview of the IoT concept. Afterwards, the three-layer IoT architecture, which is the typical IoT architecture in the literature, is presented where the Perception Layer (i.e., IoT network), the focal point of this PhD research work, is discussed. Following this, an overview of the main security attacks against IoT networks is given along with security and privacy protection requirements for IoT and security considerations for developing secure IoT ecosystems. Next, the most popular ML algorithms used in IoT Anomaly-based Intrusion Detection Systems (AIDS) are reviewed and their main advantages and drawbacks are discussed, followed by the metrics based on which their performance is evaluated. Last but not least, five of the most well-known existing datasets for training and evaluation of IoT AIDSs are reviewed.

- **Chapter 3** provides a detailed description of the approach followed to generate a set of benign datasets from a benign IoT network scenario implemented in the Cooja simulator. The generated IoT-specific information from the simulated scenario was captured from the Contiki plugin “powertrace” (i.e., features such as CPU consumption) and the Cooja tool “Radio messages” (i.e., network traffic features) to generate the “powertrace” dataset and the network traffic dataset, respectively, within csv files. The generated datasets constitute the benign IoT datasets for the simulated benign IoT network scenario.

- **Chapter 4** is focused on the generation of a set of malicious datasets from the following attack scenarios implemented in the Cooja simulator: i) UDP flooding attack, ii) blackhole attack, iii) sinkhole attack, and iv) sleep deprivation attack. The generated IoT-specific information from the simulated attack scenarios was captured from the Contiki plugin “powertrace” (i.e., features such as CPU consumption) and the Cooja tool “Radio messages” (i.e., network traffic features) in order to generate the corresponding “powertrace” and network traffic datasets for each of the simulated attack scenarios within csv files. The generated datasets constitute the malicious IoT datasets for the simulated IoT attack scenarios.

- **Chapter 5** is focused on the analysis of the generated benign “powertrace” and network traffic datasets, presented in Chapter 3, and the generated malicious “powertrace” and network traffic datasets, demonstrated in Chapter 4. The Chapter starts with the analysis of the malicious “powertrace” datasets to investigate whether their raw features can be important in the detection of anomalies in the network-level power profiling of low-power IoT devices due to UDP flooding attacks, blackhole attacks, sinkhole attacks, or sleep deprivation attacks. Next, the Chapter continues with investigating the extraction of new features, more informative and non-redundant, based on the raw features of the generated benign and malicious datasets. The new features are intended to constitute valuable features for anomaly-based detection of UDP flooding attacks, blackhole attacks, sinkhole attacks and sleep deprivation attacks in

IoT networks. To this end, the total energy consumption of each mote is investigated as a valuable feature in Section 5.2.2. Last but not least, the generated benign and malicious network traffic datasets are also analysed in Section 5.3.1 to derive new features more informative in terms of the behaviour of the network traffic.

- **Chapter 6** is focused on the validation of the generated malicious “powertrace” datasets, presented in Chapter 4, by applying different Machine Learning (ML) algorithms for IoT AIDSs to evaluate their performance on the generated malicious datasets. In particular, the following most popular ML algorithms for IoT AIDSs, reviewed in Section 2.3, were applied: naïve Bayes (NB), decision tree (DT), random forest (RF), logistic regression (LR), support vector machines (SVM), and k-nearest neighbor (KNN). Using five-fold cross validation, these algorithms were trained and tested over the same labelled dataset for each attack scenario. Furthermore, the following four traditional metrics, reviewed in Section 2.4, were used to evaluate the performance of the ML algorithms on the generated datasets when these algorithms are used for anomaly detection in IoT AIDSs: accuracy, precision, recall, and F1-score. In all experiments, the Python language (version 3.8.2) was used, along with the Scikit-Learn library [27] and a Python script created, utilizing specific functions of the Scikit-Learn library, to perform training and testing of the ML algorithms.

- **Chapter 7** concludes this PhD thesis and provides future research objectives.

This page intentionally left blank.

Chapter 2 Related Work

2.1 Introduction

This Chapter is focused on giving a comprehensive overview of the four main pillars of this PhD research work: i) *Internet of Things (IoT)*, ii) *Machine Learning (ML) algorithms for anomaly-based intrusion detection in IoT networks*, iii) *evaluation metrics for the performance of ML algorithms*, and iv) *existing datasets for training and evaluation of anomaly-based intrusion detection in IoT networks*. Therefore, the Chapter starts with an overview of the IoT concept. Afterwards, the three-layer IoT architecture, which is the typical IoT architecture in the literature, is presented where the Perception Layer (i.e., IoT network), the focal point of this PhD research work, is discussed. Following this, an overview of the main security attacks against IoT networks is given along with security and privacy protection requirements for IoT and security considerations for developing secure IoT ecosystems. Next, the most popular ML algorithms used in IoT Anomaly-based Intrusion Detection Systems (AIDS) are reviewed and their main advantages and drawbacks are discussed, followed by the metrics based on which their performance is evaluated. Last but not least, five of the most well-known existing datasets for training and evaluation of IoT AIDSs are reviewed.

2.2 Internet of Things (IoT)

In this Section, an overview of the IoT concept along with its fundamental characteristics and high-level requirements is given. Then, the three-layer IoT architecture, which is the typical IoT architecture in the literature, is presented and an overview of the main security attacks against IoT networks is provided. Furthermore, the security and privacy protection requirements for IoT, according to ITU-T Recommendation Y.2066 [28], are presented. Concluding this Section, concerns that limit the consolidation of secure IoT ecosystems, according to ENISA in [29], are discussed.

2.2.1 An Overview

The Internet of Things (IoT) is the latest development in the long and continuing revolution of computing and communications [30]. Its size, ubiquity, and influence on everyday lives, business, and government dwarf any technical advance that has gone before.

The Internet of Things (IoT) is a term that refers to the expanding interconnection of smart devices, ranging from appliances to tiny sensors [30]. A dominant theme is the embedding of short-range mobile transceivers into a wide array of gadgets and everyday items, enabling new forms of communication between people and things, and between things themselves. The Internet now supports the interconnection of billions of industrial and personal objects, usually through cloud systems. The objects deliver sensor information, act on their environment, and in some cases modify themselves, to create overall management of a larger system, such as a factory or city [30].

The IoT is primarily driven by deeply embedded devices [30]. These devices are low-bandwidth, low-repetition data capture, and low-bandwidth data-usage appliances that communicate with each other and provide data via user interfaces. Embedded appliances, such as high-resolution video security cameras, video VoIP phones, and a handful of others, require high bandwidth streaming capabilities. Yet countless products simply require packets of data to be intermittently delivered.

2.2.1.1 Evolution

With reference to the end systems supported, the Internet has gone through roughly four generations of deployment culminating in the IoT [30]:

1. **Information technology (IT):** PCs, servers, routers, firewalls, and so on, bought as IT devices by enterprise IT people, primarily using wired connectivity.
2. **Operational technology (OT):** Machines/appliances with embedded IT built by non-IT companies, such as medical machinery, SCADA (supervisory control and data acquisition), process control, and kiosks, bought as appliances by enterprise OT people, primarily using wired connectivity.
3. **Personal technology:** Smartphones, tablets, and eBook readers bought as IT devices by consumers (employees) exclusively using wireless connectivity and often multiple forms of wireless connectivity.
4. **Sensor/actuator technology:** Single-purpose devices bought by consumers, IT, and OT people exclusively using wireless connectivity, generally of a single form, as part of larger systems. The fourth generation is usually thought of as the IoT, and which is marked by using billions of embedded devices.

2.2.1.2 Useful Definitions

Before providing the fundamental characteristics, high-level requirements and ITU reference of the IoT, attention must be drawn to some basic IoT-related definitions provided by ITU-T Y.2060 (06/2012) [31] in order to establish good understanding:

- ***Device***: With regard to the IoT, this comprises a piece of equipment enabled with obligatory communication capabilities and other optional capabilities such as sensing, actuation, data capture, data storage and data processing capabilities.
- ***Internet of Things***: A global information infrastructure that supports advanced services by interconnecting physical and/or virtual Things based on existing and/or evolving interoperable technologies. In particular, an IoT enables services for identification, data capture, processing, and communication to a wide variety of applications whilst ensuring security and privacy requirements.
- ***Thing***: An object inside the IoT system enabled with capabilities of being identified and integrated into communication systems. The thing might be physical or virtual. ***Physical Thing*** is an object of the physical world enabled with capabilities of being sensed, actuated, and connected to other Things and /or systems (e.g., industrial robots, electrical equipment etc.). On the other hand, ***Virtual Thing*** is an object in the information world enabled with capabilities of being stored, processed and accessed (e.g., multimedia content, application software etc.).

2.2.1.3 Concept of IoT

The Internet of things (IoT) can be perceived as a far-reaching vision with technological and societal implications [31]. IoT has become a very popular topic of Research and Innovation mainly due to the ubiquitous transformation of computing. Devices have come to be "smart" enabled with capabilities to sense, communicate in a pervasive way and interact with their environment making possible a wide range of useful applications and solutions to the humanity such as Health, Transportation, Agriculture, Home and Industrial Automation, Retail and many more. The 2005 ITU Internet Report

[32] was the first to add a 3rd dimension to the legacy "ANY PLACE" and "ANY TIME" communication: the "ANY THING" communication, as illustrated in Figure 2.1. This addition changed the way we used to perceive the word "telecommunication" to communication between everything rather than communication between individuals only. This implied an expected exponential growth of "smart" interconnected devices leading to network connections which should be facilitated by powerful networks [33].

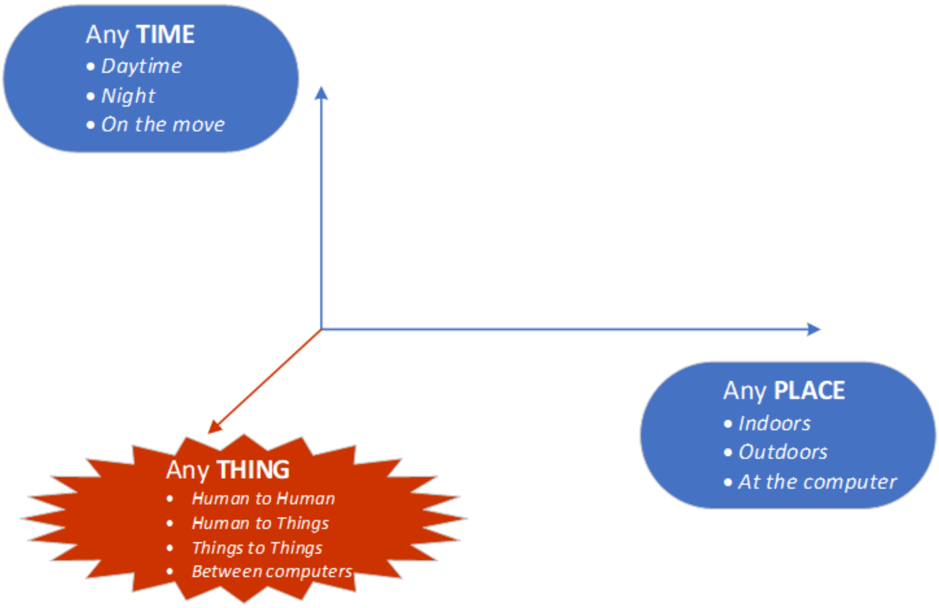


Figure 2.1. The new Dimension of IoT. (Source: [32]**)**

To ensure connectivity and interoperability, it important to establish a common accepted reference IoT architecture upon which all IoT applications would be based. Towards this direction, the International Telecommunications Union (ITU) has started an effort to standardize the functional architecture model for IoT providing a 3-layer architecture in 2012 [31]. In the following, based on [31], we present the fundamental characteristics of IoT, its high-level requirements, the IoT reference model proposed by ITU, fundamentals on IoT security, and finally some baseline security recommendations provided by ENISA [34].

2.2.2 Fundamental Characteristics and High-Level Requirements of the IoT

2.2.2.1 Fundamental Characteristics of the IoT

In [31], ITU-T has also identified the fundamental characteristics of an IoT system. According to their findings, those characteristics are the following:

- ***Heterogeneity***: refers to the various heterogeneous IoT devices that comprise an IoT network. These devices although they have very different hardware and networking characteristics, they get connected to each other and interact with other IoT devices and/or platforms on various types of networks.

- ***Interconnectivity***: refers to the fact that any IoT device is enabled with capabilities to interconnect/be interconnected with the global Information and Communication Infrastructure.

- ***Enormous scale***: refers to the number of devices required to be interconnected and managed is significantly larger in an IoT environment. This practically means that the communication initialized by devices is much higher than the one that is initialized by humans. Even more important is the management and the analysis of the data generated. This relates to semantics of data, as well as efficient data handling.

- ***Things-related services***: The IoT provisions services related to the connected "things" within their constraints such as privacy protections and semantic consistency between physical things and their associated virtual things. To provide these thing-related services within the constraints of things requires that both the underlying technologies and the physical and information world change.

- ***Dynamic Changes***: While roaming and interacting in an IoT system, the state of devices dynamically change (e.g., get connected or disconnected, sleeping and waking up). Besides, the context of devices dynamically changes (e.g., location speed). Additionally, the number of interconnected devices changes dynamically as well within IoT systems.

2.2.2.2 High-level Requirements

In [31], ITU-T has provided a set of high-level IoT System Requirements for the development of an IoT Reference Model based on the fundamental characteristics of an IoT system identified above. According to their findings, those requirements are the following:

- ***Identification-based connectivity***: refers to capabilities that enable the smart "Things" to be connected to the IoT networks based on their identifiers. This includes a unified processing of identifiers which might be heterogeneous.

- ***Interoperability***: needs to be ensured within IoT networks to support a variety of information and services given the fact that IoT networks are highly heterogeneous and distributed systems.

- ***Autonomic services provisioning***: refers to specific operations of the IoT network infrastructure that will enable IoT services to be provided by automatically capturing, communicating and processing of the data of the "Things" according to the rules configured by the operators and/or configured by the subscribers. Autonomic services may depend on automatic data fusion and data mining techniques.

- ***Automatic Networking:*** refers to specific operations of the IoT network infrastructure that will enable automatic networking including self-management, self-configuration, self-healing, self-optimization, and self-protection for supporting and facilitating adaptation in different application domains, different communication environments and large number and types of devices.

- ***Location-based capabilities***: Localization is a key enabling technology in IoT considering that location-based services must be supported. Towards this direction, smart Things should be enabled with capabilities to track their position to facilitate the provision of services which depend on their location. Attention must be drawn to the fact that, nowadays, location-based communication and services are highly restricted by Regulations and Laws and thus, when addressing this requirement, we should keep in mind that we need to comply with them.

- ***Privacy protection***: Data acquired by "Things" may contain sensitive private information of the consequent users. It is very important that privacy concerns should comply with the relevant established privacy Regulations and Laws and privacy protection to be taken into consideration during all processes related to data such as data transmission, data aggregation, data storage, data mining and data processing while, at the same time, not setting a barrier to data source authentication.

- ***Security***: refers to the necessity to integrate security policy and measures related to the things and their communication in an IoT framework. This is mainly because the capabilities of Things to connect at any time, any place and any (other) thing introduces significant security threats against CIA (Confidentiality, Integrity and Authenticity) for both data and services within IoT networks. Therefore, security comprises an important requirement that needs be addressed in advance in order for the emerging IoT applications to gain the trust of all involved stakeholders and reach their full potential in the 5G market.
- ***High quality and highly secure human body related services***: refers to the requirement of guaranteeing high data quality, data accuracy and data security for data derived automatically or through human intervention for particular services that are based on the capturing, communicating, and processing of data related to human static features and dynamic behaviour with or without human intervention.
- ***Manageability***: generally, the applications in an IoT system have to work automatically and without or insignificant human intervention or participation. Towards this direction, there is the necessity for the whole operation process to be easily manageable by the relevant entities in order to ensure normal network operations without significant delays.
- ***Plug and Play***: refers to plug and play capabilities of IoT systems in order to enable or facilitate on-the-fly generation, composition and acquisition of semantic-based configurations to seamlessly integrate an internetwork of things with the respective applications and efficiently respond to these applications' requirements.

2.2.3 IoT Architecture

The three-layer IoT architecture, shown in Figure 2.2, is the typical IoT architecture consisting of three main layers [1], [35]: 1) perception layer; 2) network layer; and 3) application layer, which are further described below. In this PhD work, the simulated scenarios in the Cooja simulator are restricted only to the perception layer of the 3-layer IoT architecture.

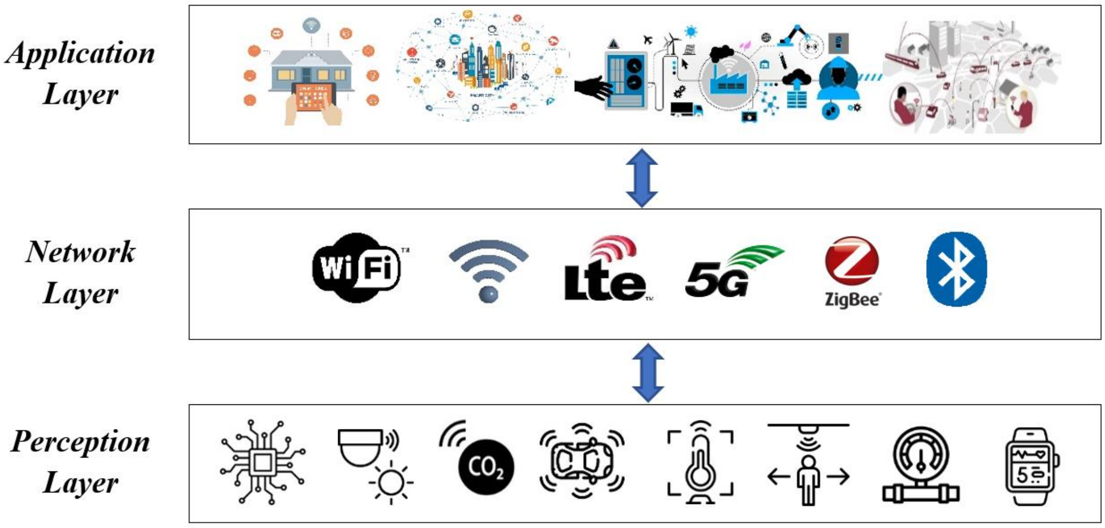


Figure 2.2. Three-layer IoT Architecture.

Perception Layer: This layer consists of devices (i.e., sensors) that enable the perception of their environment and thus, it can also be perceived as the Device Layer in the ITU-T reference model [31]. The Perception Layer can be considered as an analogue to the senses or nerve endings of a human being such as the eyes, ears, nose, skin, etc. In particular, the perception layer includes sensing devices such as thermometers, humidity sensors, and medical sensors [36], [37], [38], [39] that measure and gather information about different parameters or conditions in their surrounding environment at a Gateway, and send it, through the Network Layer, to the Application Layer where it is processed and stored. In addition to its sensing capabilities, this layer also includes devices (i.e., actuators) which are responsible to perform actions (e.g., control commands) based on the decisions taken at the Application Layer.

Network Layer: It is the transmission layer and its main function is to to receive the data, gathered by the Perception Layer, through a Gateway, and determine the routes so as to transmit them to the Application Layer through integrated networks. On the other hand, the Network Layer is responsible to transmit the required actions (e.g., control commands) determined at the Application Layer to the actuators in the Perception Layer, through a Gateway. The Network Layer might be implemented using the current or the evolving network and mobile technologies such as IEEE802.11 standards, 4G, 5G, Bluetooth, Zigbee, and also numerous types of networking and data collection protocols such as MQTT, TCP/IP etc [39], [40], [41], [42]. In addition to its capabilities for connectivity and networking, this layer includes management operation for the seamless and flawless operation of the integrated IoT systems.

Application Layer: This layer is in charge of delivering IoT application services to the users/subscribers. To do this, it utilizes the gathered context from the layers below to deliver intelligent applications such as smart-home, e-health, smart-transport etc. to the end users [43], [44], [45], [46], [47]. This layer comprises the final goal of the IoT system consolidating inputs from the underlying technologies to offer useful and user-friendly applications to the end users. It therefore mostly includes intelligent software development functions. It can be seen as the means to converge between the social IoT needs and the industrial technology in such a way as to have a broad impact on the global or local economic or social development.

2.2.3.1 Device and Gateway Capabilities of IoT networks

In general, the leading purpose of IoT is to connect objects (e.g., physical things, virtual things etc.) into the IoT network, and to measure, gather and handle the information provided by these objects through IoT devices of the IoT edge network (i.e., perception layer) that transmit the gathered information to the next layer (i.e., network layer) of the IoT-based smart system via domain interfaces [48]. To achieve that, IoT networks are enabled with capabilities that logically can be classified into two main categories [31]: i) *device capabilities* that mainly include the direct interaction with the communication network, indirect interaction with the communication network, ad-hoc networking, and sleeping and waking-up capabilities, and (ii) *gateway capabilities* that include multiple interfaces support and protocol conversion as there are generally two situations where protocol conversion is required. The first situation is when communications at the device layer use different device layer protocols, such as Bluetooth technology protocols and ZigBee technology protocols. While the second one is when communications involve two different layers (i.e., perception/device layer and network layer) and different protocols are utilized at each layer (e.g., a Bluetooth technology protocol at the perception/device layer and a 4G/5G technology protocol at the network layer) [31].

2.2.4 Security Attacks in the IoT Network – Perception Layer Environment

Security on IoT network – Perception Layer is a significant challenge due to the heterogeneity and vast number of its IoT devices and connections [3], [6], [49], [50], [51]. As the main purpose of the IoT network is to gather data, attackers mainly target to forge/steal transmitted/collected IoT data, damage perception IoT devices, and make the whole IoT network or specific IoT nodes unavailable, as presented below.

2.2.4.1 Sinkhole attacks

In this type of attacks, a compromised IoT node (i.e., IoT gateway) in the Perception Layer proclaims very appealing false capabilities of power, computation and communication (e.g., shortest route) [1] so that nearby nodes (i.e., adjacent IoT sensors) will choose it as the forwarding node in the routing process due to its very attractive capabilities. As a consequence, the compromised IoT node can increase the amount of obtained IoT data that in turn are dropped or modified before they are delivered to the Application Layer system via the Network Layer. Therefore, a sinkhole attack can not only compromise the confidentiality of the IoT data but also can constitute an initial step to launch additional attacks such as DoS/DDoS attacks [1], [51], [52].

2.2.4.2 Node capture attacks

In this type of attack, the adversary is able to extract important information about the captured node, such as the group communication key, radio key, etc. Additionally, the adversary can copy the important information related to the captured node to a malicious node, and afterwards fake the malicious node as a legitimate node to connect to the IoT network (i.e., Perception Layer). This type of attack is also known as node cloning/replication attack. This attack may lead to compromising the security of the complete IoT-based system [1], [53].

2.2.4.3 Malicious code injection attacks

An attacker can take control of an IoT node or device in the Perception Layer by exploiting its security vulnerabilities in software and hardware and injecting malicious code into its memory. Afterwards, using the malicious code, the attacker can force the node or device to perform unintended operations. For example, the infected IoT node(s) or device(s) can be used as a bot(s) to launch further attacks (e.g., DoS, DDoS) against other devices or nodes within the Perception Layer or even against the other Layers. In addition, the attacker can use the injected malicious code in the infected device or node to get access into the IoT-based system and/or get full control of the system [1], [6], [54].

2.2.4.4 False data injection attacks

After capturing an IoT node or device in the Perception Layer, the adversary can inject false data in place of benign data measured by the captured IoT node or device and transmit the false data to the Application Layer via the Network Layer. Thereafter, receiving the false data, the Application Layer may provide wrong services, which further negatively impacts the effectiveness of IoT-based system relying on the Perception Layer [1], [55].

2.2.4.5 Replay attacks

In the Perception Layer, the attacker can use a malicious IoT node or device to transmit to the destination host (i.e., IoT gateway) with legitimate identification information, already received by the destination host, so that the malicious node or device can become a trusted node/device to the destination host. Replay attacks are commonly launched in authentication process to destroy the validity of certification [1].

2.2.4.6 Eavesdropping

As the IoT nodes and devices in Perception Layer communicate via wireless networks, an attacker (i.e., eavesdropper) can retrieve sensitive IoT data by overhearing the wireless transmission. For instance, an adversary within the Perception Layer can eavesdrop exchanged information by tracking wireless communications and reading the contents of the transmitted packages. The eavesdropper can passively intercept the wireless communication between a sensor (e.g., environment industrial sensors or sensors on the machine resources) and the IoT gateway, and extract confidential data (e.g., through traffic analysis) in order to maliciously use them [1], [56], [57].

2.2.4.7 Sleep deprivation attacks or Denial of Sleep attacks

These attacks target to drain the battery of the resource constrained IoT nodes of the Perception Layer. In principle, the IoT nodes in the Perception Layer are usually programmed to follow a sleep routine when they are inactive in order to reduce the power consumption and extend their life cycle. However, an adversary may break the programmed sleep routines and keep the IoT nodes continuously active until they are shut down due to a drained battery. Attackers can achieve this by running infinite loops in these resources using malicious code or by artificially increasing their power consumption [1], [3], [58].

2.2.4.8 Sybil attacks

In a sybil attack, a malicious or sybil node or device can illegitimately claim multiple identities, allowing it to impersonate them within the Perception Layer. For instance, the malicious node can achieve to connect with several other devices in order to maximize its influence and even deceive the complete system to draw incorrect conclusions[1], [59], [60].

2.2.4.9 Blackhole attacks

In a blackhole attack, the intention of the attacker is to create an artificial packet loss in the Perception Layer. To achieve that, a compromised IoT node drops the received packets that have to be routed to other IoT nodes [61]. This attack can be very damaging when combined with a sinkhole attack causing the loss of a large part of the traffic. If the compromised node is located at a strategic position in the network it can isolate several nodes [62], [63].

2.2.4.10 Denial of Service (DoS) attacks

The main target of these attacks is to deplete resources of the Perception Layer in order to make the whole IoT network or specific nodes or devices (e.g., IoT gateway) unavailable. For instance, jamming attacks are a type of DoS attacks where an attacker transmits a high-range signal to overload the communication channel between two communicating entities and disrupt their communication. Within the Perception Layer, jamming attacks can disrupt the communication between the IoT sensors and the Gateway in order to prevent IoT data from being transmitted to the Gateway, leading to malfunctions in the provided services to the authorized users. Jamming attacks can be performed by passively listening to the wireless medium so as to broadcast on the same frequency band as the legitimate transmitting signal. Moreover, a DoS attack can be carried out within the Perception Layer by a compromised IoT node flooding the Gateway with a lot of transmitted data/requests (e.g., UDP packets) and render it unavailable or disrupt its normal operations [1], [35], [64], [65].

2.2.5 IoT Security and Privacy Requirements

According to ITU-T Recommendation Y.2066 [66], a list of security and privacy protection requirements for IoT is provided. The requirements refer to the functional requirements during capturing, storing, transferring, aggregating, and processing the data of things, as well as to the provision of services which involve things. These requirements are related to all the IoT actors. The requirements are the following:

- **Communication security:** Secure, trusted, and privacy protected communication capability is required so that unauthorized access to the content of data can be prohibited, data integrity can be guaranteed, and privacy-related content of data can be protected during data transmission or transfer in IoT.

- **Data management security:** Secure, trusted, and privacy protected data management capability is required so that unauthorized access to the content of data can be forbidden, data integrity can be guaranteed, and privacy-related content of data can be secured when storing or processing data in IoT.

- **Service provision security:** Secure, trusted, and privacy protected service provision capability is required so that unauthorized access to service and illicit service provision can be forbidden and privacy information related to IoT users can be protected.

- **Integration of security policies and techniques:** The ability to integrate different security policies and techniques is required in order to ensure a consistent security control over the variety of devices and user networks in IoT.

- **Mutual authentication and authorization:** Before a device (or an IoT user) can access the IoT, mutual authentication and authorisation between the device (or the IoT user) and IoT is essential to be performed based on predefined security policies.

- **Security audit:** Security audit is necessary to be supported in IoT. Any data access or attempt to access IoT applications are required to be fully transparent, traceable and reproducible based on appropriate regulation and laws. In particular, IoT is required to support security audit for data transmission, storage, processing, and application access.

2.2.6 Security Requirements of the Gateway

A key element in achieving security in an IoT deployment is the Gateway. ITU-T Recommendation Y.2067 in [28] provides specific security requirements that the Gateway should implement, some of which are illustrated in Figure 2.3.

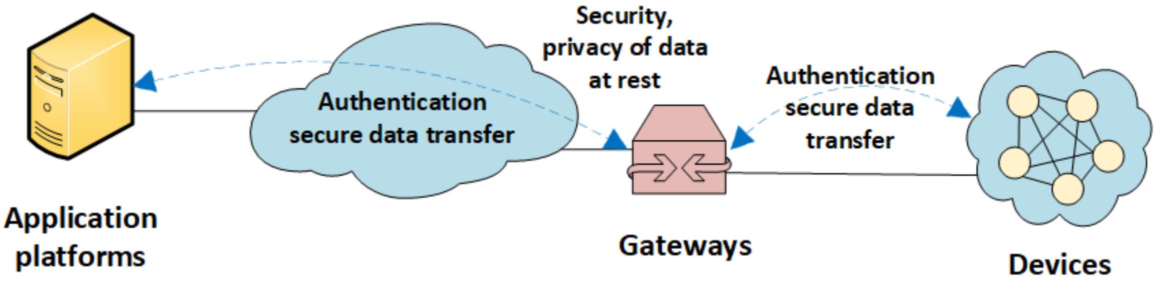

Security, privacy of data at rest
Authentication secure data transfer
Authentication secure data transfer
Application platforms
Gateways
Devices

Figure 2.3. IoT Gateway Security Functions.

In particular, according to [28], the Gateway is required to:

- support identification of each access to the connected devices.
- support authentication with devices. Based on application requirements and device capabilities, the Gateway is required to support mutual or one-way authentication with devices.
- support mutual authentication with applications.
- support the security of the data that are stored in devices and the Gateway, or transferred between the Gateway and devices, or transferred between the Gateway and applications – the Gateway is required to support the security of these data based on security levels.
- support mechanisms to protect confidentiality for devices and the Gateway.

2.2.6 Security Considerations

As time passes, we are becoming increasingly dependent on smart, interconnected devices for a lot of tasks in our everyday lives. Nevertheless, the same devices or “things” can be the target of attacks and intrusions that can cause malfunction of devices and endanger our personal privacy and public safety. Thus, it is evident that security is one of the main challenges that should be seriously considered together with safety in IoT. These two matters are always closely connected with the physical world. Furthermore, one more issue concerns the administration of IoT devices, meaning who will be the supervisor and manage the devices. The difficulty of the administration task can be better understood, considering the inherent complexity and diversity of the IoT ecosystem and its scalability issues [29].

There are a lot of different concerns that limit the consolidation of secure IoT ecosystems. Below, some of these concerns are presented [29]:

- **Very large attack surface:** IoT-related risks and threats are many in number and are constantly changing. Also, IoT devices and services affect citizens’ health, safety and privacy since devices gather, exchange and process data from various sources sometimes including sensitive data. Because of the aforementioned, the attack range related to IoT is extremely wide.

- **Limited device resources:** Technical constraints in IoT means that conventional security practices cannot be applied as they are, but significant reengineering will be required. A characteristic of a majority of IoT devices is their inherent limited capabilities as far as processing, storage and power are concerned. Therefore, advanced security controls cannot be effectively implemented.

- **Complex ecosystem:** One more reason that security concerns regarding IoT are enhanced is that IoT is often depicted as a collection of independent devices. In reality, it should be considered as a large and diverse ecosystem including devices, communications, interfaces and people.

- **Fragmentation of standards and regulations:** IoT security concerns are additionally complicated due to the fact that standards and regulations about IoT security measures are slowly adopted, and simultaneously new technologies are constantly emerging.

- **Widespread deployment:** Not only commercial IoT applications, but also Critical Infrastructures (Cis) have recently started to migrate toward Smart ones. This is achieved by implementing IoT on top of legacy infrastructures.

- **Security integration:** The potentially opposing viewpoints and requirements from all involved stakeholders complicate matters relating to security integration. An instance of that would be IoT systems with different authentication methods, which should be able to communicate and operate with each other seamlessly.

- **Safety aspects:** The presence of actuators or other devices which operate on the physical world turns security threats into safety threats in the IoT context.

- **Low cost:** As IoT and its advanced functionalities are employed in several sectors, the potential for considerable cost savings is further highlighted. The reduced costs can be achieved by implementing features such as data flows, advanced monitoring, and integration. However, the low cost of IoT devices and systems can become an important obstacle in implementing security solutions. Manufacturers tend to care more about decreasing production costs. As a result, security features become more limited and product security possibly cannot protect against specific IoT attacks.

- **Lack of expertise:** Since the IoT domain is a comparatively new one, not a lot of people possess the suitable skillset and experience in IoT cybersecurity.

- **Security updates:** It is extremely challenging to apply security updates to IoT systems. IoT User interfaces, in their majority, do not allow traditional update mechanisms. Securing of those mechanisms, especially considering Over-The-Air updates, is in itself a really difficult task.

- **Insecure programming:** The “time to market” pressure for products of the IoT domain is higher compared to other domains and thus, limitations are imposed on the efforts to develop security and privacy by design. For this reason, and sometimes also due to budget issues, more emphasis is directed towards the functionality of the IoT products rather than their integrated security.

- **Unclear liabilities:** The assignment of liabilities is unclear. Therefore, in case of security incidents, many ambiguities and conflicts can be raised, especially considering the large and complex supply chain involved in IoT. On top of that, the challenge of how to manage security if one single component was shared by several parties remains open. Last but not least, enforcing liability is another major challenge.

2.3 Machine Learning Algorithms for Anomaly-based Intrusion Detection in IoT Networks

In this Section, we review the most popular ML algorithms used in IoT Anomaly-Based Intrusion Detection Systems (AIDS). In particular, the most commonly used algorithms in the literature are the following: naïve Bayes (NB), decision tree (DT), random forest (RF), linear regression (LR), logistic regression (LR), support vector machines (SVM), and k-nearest neighbour (KNN). In [67], the authors stated that the aforementioned ML algorithms have been commonly used in the design and development of various efficient and effective AIDS for IoT. On top of that, in [14], the authors also highlighted that k-nearest neighbor (KNN), logistics regression (LR), support vector machines (SVM), decision tree (DT), random forest (RF), and naïve Bayes (NB) constitute suitable ML algorithms for the design and development of efficient and effective AIDS for IoT. At the end of the section, we provide Table 2.1 with an overview of all ML algorithms presented in this section, along with their main advantages and drawbacks when applied in the design and development of anomaly detection systems for IoT.

2.3.1 Naïve Bayes (NB)

Naive Bayes (NB) is a supervised ML algorithm that operates by applying Bayes' theorem to calculate the probability of occurrence of an event (i.e., normal or abnormal [68]) based on previous observations of similar events with the "naive" assumption of conditional independence between every pair of features given the value of the class variable in order to simplify the process of modelling [69]. Regardless this controversial assumption, it is anticipated that Naïve Bayes is a fast classifier and has a great performance in practice for many domains. The NB classifier is a commonly employed supervised classifier with main advantages the ease of implementation and its simplicity.

Given events *Y* and *X* with $P(X) \neq 0$, Bayes' theorem states the following:

$$P(Y|X) = \frac{P(Y)P(X|Y)}{P(X)}$$

where,

$P(Y|X)$ represents the conditional probability of *Y* occurring given that *X* is true,

$P(X|Y)$ represents the conditional probability of *X* occurring given that *Y* is true,

$P(Y)$ represents the probability of *Y* occurring without any condition, and

$P(X)$ represents the probability of *X* occurring without any condition.

Nevertheless, in a real case classification problem, there can be multiple X variables depending on the features of the training data. Hence, in the situation, Bayes Theorem is extended to Naïve Bayes considering that features are independent:

$$P(Y|X_1, \cdots, X_n) = \frac{P(Y)P(X_1, \cdots, X_n|Y)}{P(X_1, \cdots, X_n)} \tag{2.1}$$

Based on the "naive" assumption of class-conditional independence, the features are conditionally independent of one another given the class, thus:

$$P(X_1, \cdots, X_n|Y) = P(X_1|Y) \cdots P(X_n|Y) = \prod_{i=1}^{n} P(X_i|Y) \tag{2.2}$$

Based on (2.1) and (2.2), we have:

$$P(Y|X_1,\cdots,X_n) = \frac{P(Y)\prod_{i=1}^{n} P(X_i|Y)}{P(X_1,\cdots,X_n)} \tag{2.3}$$

Since $P(X_1,\cdots,X_n)$ is constant given the input, we can use the following classification rule:

$$P(Y|X_1,\cdots,X_n) \propto P(Y)\prod_{i=1}^{n} P(X_i|Y)$$

$$\hat{Y} = arg\max_{Y} P(Y)\prod_{i=1}^{n} P(X_i|Y)$$

In addition, we can use Maximum A Posteriori (MAP) estimation to estimate $P(Y)$ and $P(X_i|Y)$; the former is then the relative frequency of class Y in the training set. This was, computing posterior probability, the algorithm classifies new unlabeled instances as normal or abnormal. Another advantage of NB is that in both binary and multi-label classification problems it does not require many samples for its proper running during its training phase. However, its feature independence assumption might negatively impact its accuracy as the NB classifier fails to perceive interdependencies among the features of a dataset [67].

It is worthwhile to highlight that there are different types of NB classifiers mainly based on the assumptions they make regarding the distribution of $P(X_i|Y)$. In general, these assumptions to define the likelihood of the features are strongly depending on the type of the data (e.g., categorical data, multinomially distributed data etc.), as well as on the application (e.g., text classification, binary classification, large scale classification etc.). For instance, it implements Bernoulli NB for data that are distributed based on multivariate Bernoulli distributions; i.e., there may be multiple features on a given training dataset, however each one is assumed to be a binary-valued (i.e., Bernoulli, Boolean) variable.

2.3.2 Decision Tree (DT)

A decision tree (DT) is a supervised ML algorithm used for classification. The main target of DTs classifier is to extract features of the training dataset and then organize an ordered tree based on the value of these features [69]. In a DT, a node corresponds to a feature of the training dataset and the branches of that node correspond to the values of that feature. The construction of the ordered tree starts from the origin node of the tree which is known as the root node. The main challenge of DTs algorithm is to select the feature, which will be the root node of the tree, in order to optimally split up the training dataset into subsets, one for every value of the selected feature. Afterwards, the process might be repeated recursively for each branch, using only those training instances that actually reach the branch (i.e., they have the feature value of the particular branch). If at any time all training instances at a node have the same classification, then the development of that part of the tree is stopped, and this class is considered the terminal node (detect or not an anomaly in the system). In order to determine which feature to split on in order to create the ordered tree, various metrics, such as Gini Index, Entropy and Information Gain, are utilized for identification of the feature that will be considered the root node, which will optimally divide the training dataset [67], [69], and for identification of which feature to split on. An example of a DT classifier is illustrated in Figure 2.4.

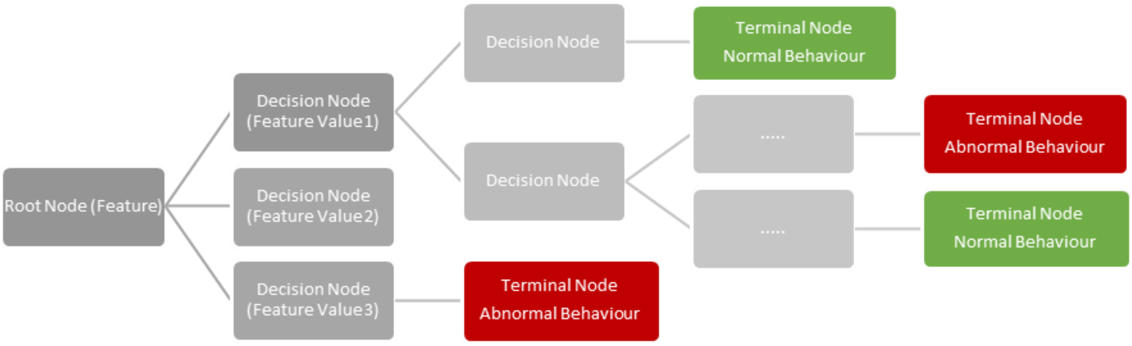

Root Node (Feature)
Decision Node (Feature Value 1)
Decision Node (Feature Value 2)
Decision Node (Feature Value 3)
Decision Node
Decision Node
Terminal Node Normal Behaviour
.....
.....
Terminal Node Abnormal Behaviour
Terminal Node Normal Behaviour
Terminal Node Abnormal Behaviour

Figure 2.4. Generic Structure of Decision Tree Model.

In [67], the authors discuss that DTs algorithms carry out two different processes: the induction process and the inference process [70]. During the induction process, the algorithm combines unoccupied nodes and branches to construct the DT. Initially, the optimal feature is selected as the root node of the DT based on the Gini Index, Entropy and Information Gain or other measures. Then, in each subsequent step, the induction process continues by selecting more features as tree nodes constructing that way the ordered tree. The main idea during the selection of the features is to keep to the minimum the overlapping among the different classes of the training dataset. In the end, the ordered tree is constructed by identifying and classifying the leaves of each sub-DT according to their corresponding classes.

On the other hand, the inference process involves the classification of new unknown instances and thus, occurs in a constructed DT. During the inference process, the algorithm, through an iterative comparison with the created DT, classifies unknown instances. This process is completed when a matching leaf node is found, and under this node the unknown instance is classified [67]. The authors in [71] performed experiments using the Gini index as a measure to select both the root node of the DT and the rest of the tree nodes. In addition, they set to 10 the minimum number of samples per leaf node in order to avoid overfitting and to end up with a pruned tree [71], [72].

2.3.3 Linear Regression (LR)

Linear Regression (LR) is a statistical supervised ML algorithm that functions by predicting the quantitative value of a variable forming a linear relationship with one or more independent features [69], as it is illustrated in Figure 2.5. In order to build a LR model, it is required to take into consideration the following assumptions [69]:

- Every independent feature in the data should be Normally Distributed. This can be examined using visualization techniques such as histogram, Q-Q plot, etc.
- The independent variables should have a linear relationship with the dependent variables. This can be also examined using visualization techniques such as Scatter plot, pairplot, Heatmap etc. in order to visualize each feature of the data in one particular plot.
- The variance of the residual should remain consistent throughout the data. This property is also referred to as homoscedasticity and can be confirmed with the residual vs fitted plot.
- The mean of the residual should be zero. Residual is the difference between the observed and predicted y-values and thus, residuals virtually zero show that the model is working well.

- Finally, there should be little or no autocorrelation in the data. Autocorrelation appears when the residuals are not independent with each other. This typically can be examined in time series analysis plotting the ACF plot or performing Durbin-Watson test. Generally, when performing Durbin-Watson test:
    - ○ if the output is 2, there is no autocorrelation;
    - ○ if the output is a value less than 2, the autocorrelation is positive; and
    - ○ if the output is a value greater than 2 and less than 4, the autocorrelation is negative.

There are 2 different types of linear regression models. The very simplest type of linear regression is when there is a single predictor variable x and a single response variable y, also referred to as simple linear regression. The extension to multiple predictor variables (i.e., X1, X2,.. Xi,) is known as multiple linear regression. In fact, multiple linear regression is a generalization of simple linear regression when there are more than one independent variables. The basic models for simple and multiple linear regression are following:

$$y = b_0 + b_1 x_1 \tag{2.4}$$

$$y = b_0 + b_1 x_1 + b_2 x_2 + .. + b_i x_i + \epsilon \tag{2.5}$$

where:

y: dependent variable

b_0: constant

$b_1, b_2, \dots, b_i$: coefficients

$x_1, x_2, \dots, x_i$: independent variables

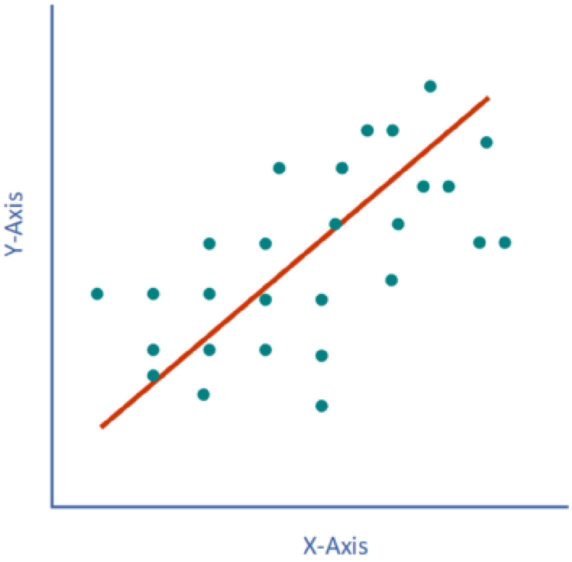


Figure 2.5. Simple Linear Regression Model.

2.3.4 Logistic Regression (LR)

A logistic regression (LR) algorithm functions by estimating the probability of a particular instance to belong in a specific class and is commonly used in an effective and efficient manner in classification problems for spam filtering (e.g., in [73]) and intrusion detection (e.g., in [74]), as illustrated in Figure

2.6. Additionally, the authors, in [75], designed and implemented a security solution based on a LR algorithm and discussed that it is possible to secure an IoT-based production line against DDoS attacks by using ML algorithms and commonly available tools for network traffic analysis.

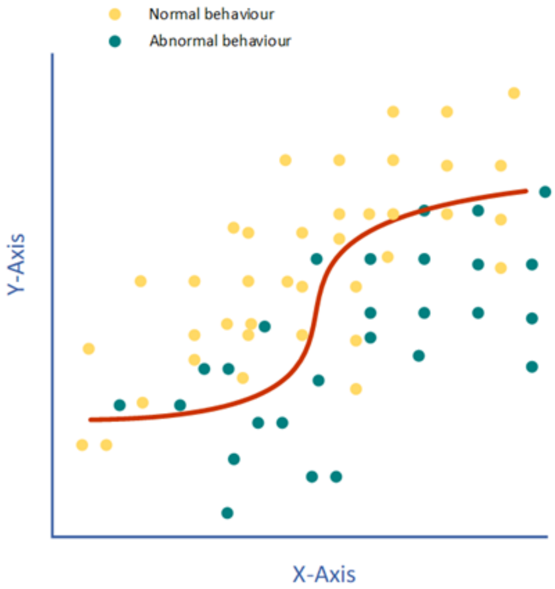


Figure 2.6. Logistic Regression Model.

The LR algorithm classifies new unknown instances utilizing a predetermined probability threshold. For example, in the case of binary classification problem (i.e., normal or abnormal activity), a threshold of 50% would mean that an instance is normal if its estimated probability is less than 50%. If the estimated probability is greater than 50%, then the LR classifier will output that this is an attack instance. The LR algorithm operates estimating this probability utilizing the following equation:

$$h_\theta(x) = \sigma(\theta^T \times x) \tag{2.6}$$

where:

h_θ is the hypothesis function, which outputs the estimated probability,

x is the feature vector of the instance,

θ is the model's parameter,

θ^T is the transpose of θ, and

σ(.) is a sigmoid (i.e., logistic) function that defines the threshold.

The equation of the sigmoid function σ(.) is the following:

$$\sigma(z) = \frac{1}{(1 + e^{(-z)})} \tag{2.7}$$

$$z = (\theta^T \times x) \tag{2.8}$$

It is worthwhile mentioning that the output of the sigmoid function is a value between 0 and 1. In particular, a number closer to 0 indicates an observation of a normal behaviour, whereas a number closer to 1 indicates an observation of an abnormal behaviour, or in other words an attack observation. During the training phase, the LR model calculates the parameter θ.

2.3.5 Support Vector Machine (SVM)

The SVM classification algorithm operates by creating an optimal hyperplane in the feature space which accurately demarks the two or more different classes. Optimal hyperplane is considered the separating hyperplane which maximizes the distance – also referred to as 'Margin' - between the nearest training instances (i.e., from both classes, meaning from both sides of the hyperplane) and the hyperplane. In particular, a margin is considered to be good if the separation is larger for both classes, and points belonging to one class should not cross to another class. In the initialization of SVM algorithm, the algorithm plots x random hyperplanes along with the training data, as for instance it is shown in Figure 2.7 (i.e., 7a) where three hyperplanes, namely 'A', 'B' and 'C', have been considered. After that, SVM attempts to adjust the orientation of the hyperplanes in such a way that it homogeneously divides the given classes. In Figure 2.7 (i.e., 7b), we can observe that all three hyperplanes, namely 'A', 'B' and 'C', segregate the two classes (i.e., yellow and green circles that represent sample of normal and abnormal observations) well. The main challenge then is to decide which of all created hyperplanes is the most appropriate (i.e., optimal) hyperplane for the particular application with the given training instances. The answer to this is to select the hyperplane with the higher margin from the nearest training instances. In this way, SVM achieves higher degree of robustness as the chance of misclassification is lower. In the example in Figure 2.7b, the hyperplane 'B' is selected as the optimal hyperplane given that the margin for hyperplane 'B' is comparatively higher than both hyperplanes 'A' and 'C'. Therefore, the hyperplane 'B' is considered as the optimal hyperplane [69].

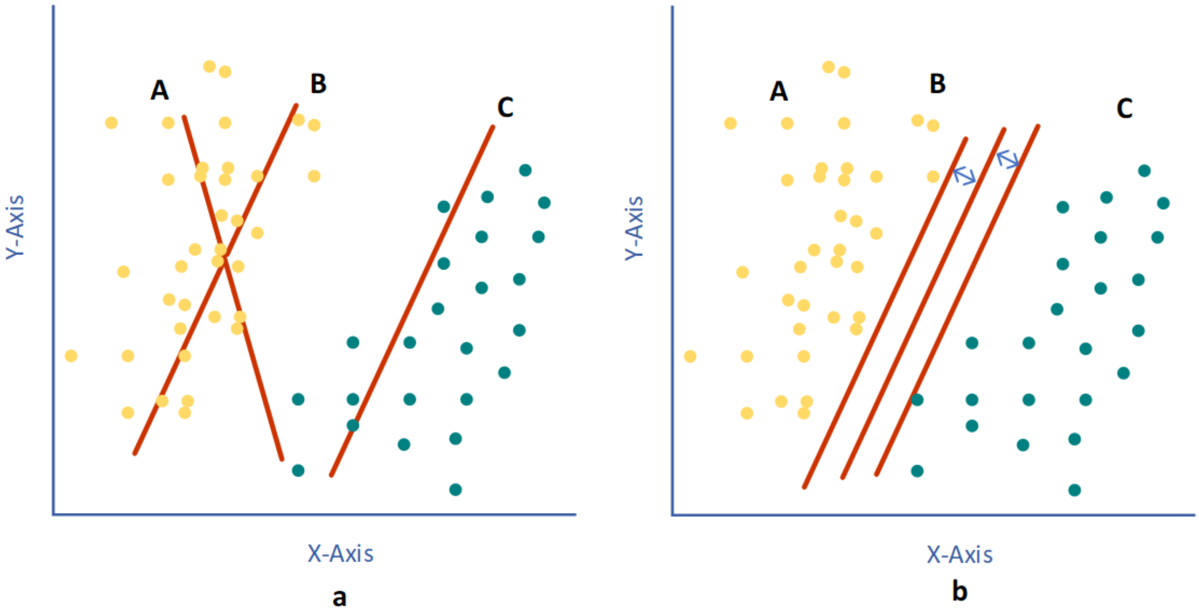


Figure 2.7. SVM model.

The best use case for SVMs is when the classification problem relates to classes with large feature sets and fewer data instances [67]. In these cases, SVM appears to have many advantages. First of all, a SVM classifier is considered to be highly scalable, due to its simplicity during both training and operating phases. Furthermore, its main advantage in intrusion detections classification problems is that SMV classifier is able to efficiently operate tasks such as anomaly-based intrusion detection in real-time, including real-time learning. In addition, a SVM classifier does not require much storage or memory to implement and does not requires many initialization parameters for each proper running [67]. As a result, due to their scalability and low requirements, SVMs appear to be suitable for use in IDSs that are implemented in a resource-constrained IoT system, and thus they require more lightweight solutions in order to operate in an effective and efficient manner. However, it is crucial to carefully consider and select the kernel function that the SVM algorithm will apply to optimally split the training data in the case that the data are not linearly separable. After finding the best kernel function to achieve a specific classification, its performance speed has always been a challenge [67]. The authors in [71], tested several functions and parameters in their SVM model for performing anomaly-based intrusion detection and in their experiments they selected an SVM classifier with a Gaussian radial basis function (RBF) kernel.

2.3.6 K-Nearest Neighbor (KNN)

The k-Nearest Neighbor (k-NN) classifier serves as an illustration of a non-parametric statistical approach and does not require any initial parameter for its proper working. The main idea of k-NN classifier is that it predicts the label of a new unclassified instance after observing the labels of the k closest training instances to this new instance (i.e., the k-nearest neighbors), and the majority class of the k closest training instances is assigned to the new instance. To achieve this, it determines the k closest training instances using a distance metric, and selects the dominant class label among them as the relevant class [69]. Generally, the Euclidean distance is typically used, while other options include Chebyshev, Manhattan, and Minkowski distances [69].

It is noteworthy that the choice of *k* - which defines the number of closest training instances (i.e., nearest neighbors) required to accurately classify the new instance - constitutes an important parameter that affects the overall performance of the classifier [69], [76], [77]. Nevertheless, the *k* can be determined experimentally, i.e., starting with *k*=1, we estimate the accuracy of the classifier, and the process is repeated increasing the number of the *k*-nearest neighbors used to predict the label of the new unclassified instance. Then, the *k*-value that achieves the higher accuracy may be selected. In general, the larger the number of training instances is, the larger the value of k will be.

In Figure 2.8, we can observe that the yellow circles depict the instances of observations of normal behavior, the green squares depict the instances of observations of abnormal behavior, or in other words an attack observation, while the new unclassified instance is represented by a dark red square. This new unclassified instance will be classified under a known class (i.e., normal or abnormal behaviour) based on the majority class of the k closest training instances. As mentioned previously, k is the number of nearest neighbors used for the classification of the new instance and it is worthwhile to highlight that the classification might be different depending on the chosen value of k [69].

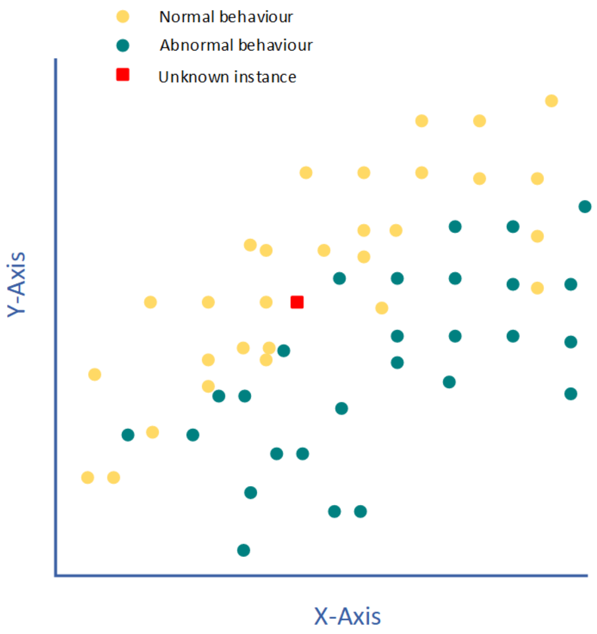


Figure 2.8. k-NN Classifier.

2.3.7 Ensemble Learning (EL)

Ensemble Learning (EL) functions by combining the classification results of several classification algorithms and then generating a majority vote out for the final classification [67], as shown in Figure 2.9. This way, EL builds on the strong points of the utilized classifiers and through this combination of various homogeneous/heterogeneous classifiers' outputs significantly improves classification accuracy [78], [79]. In [80], the author showed that the accuracy of every ML classification algorithm strongly depends on the application as well as the associated data (i.e., training and testing data). Hence, there is not a single ML algorithm that can be described as "one size fits all solution" with high accuracy for various generalized applications. On the contrary, EL schemes which combine a variety of classification results derived from several classification algorithms might comprise an optimal solution for generalized applications as they appear to be best suited for maximizing accuracy through a reduction in variance and avoiding overfitting [67].

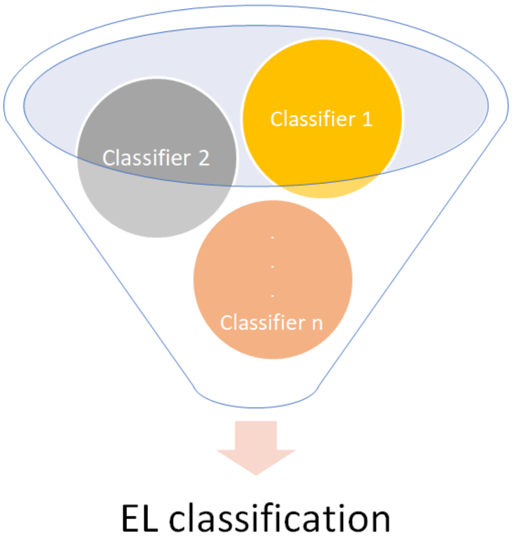


Figure 2.9. EL Classification.

Nevertheless, the high accuracy of EL classifiers has a result of high cost in terms of increased time and complexity, due to the use of multiple classifiers at the same time [81]. There are various studies in the literature that have examined the efficiency of EL for the intrusion detection problem [82], [83], [84]. Furthermore, there are research works on the feasibility of EL in resource-constrained environments such as IoT networks. For instance, in [85], the authors proposed a generalized application lightweight EL framework being proposed for online anomaly detection in IoT networks. On top of that, the authors in this study demonstrated that the proposed EL framework outputted better and more accurate results than each member classifier individually [85].

2.3.7.1 Random Forest (RF)

A random forest (RF) is a supervised ensemble ML algorithm used for classification, regression and other tasks that functions by constructing a multitude of DTs at training time, as it can be seen in Figure 2.10. This way, it achieves error resistant classifications, while it is proved to be more accurate than simple DTs [67], [71], [72]. To do this, during the training phase, the algorithm constructs random DTs from the features of the training dataset and afterwards the model is trained to classify new unknown instances based on to majority voting of those DTs [67], [71], [72]. The DTs that constitute an RF classifier are trained in a different way compared to the simple DTs described in Section 2.3.2. In particular, the difference relies on the fact that the ruleset of a simple DT is created based on the given training dataset during the training phase, while in a RF ensemble ML model the various DTs are generated using randomly picked instances from the training dataset as an input [86].

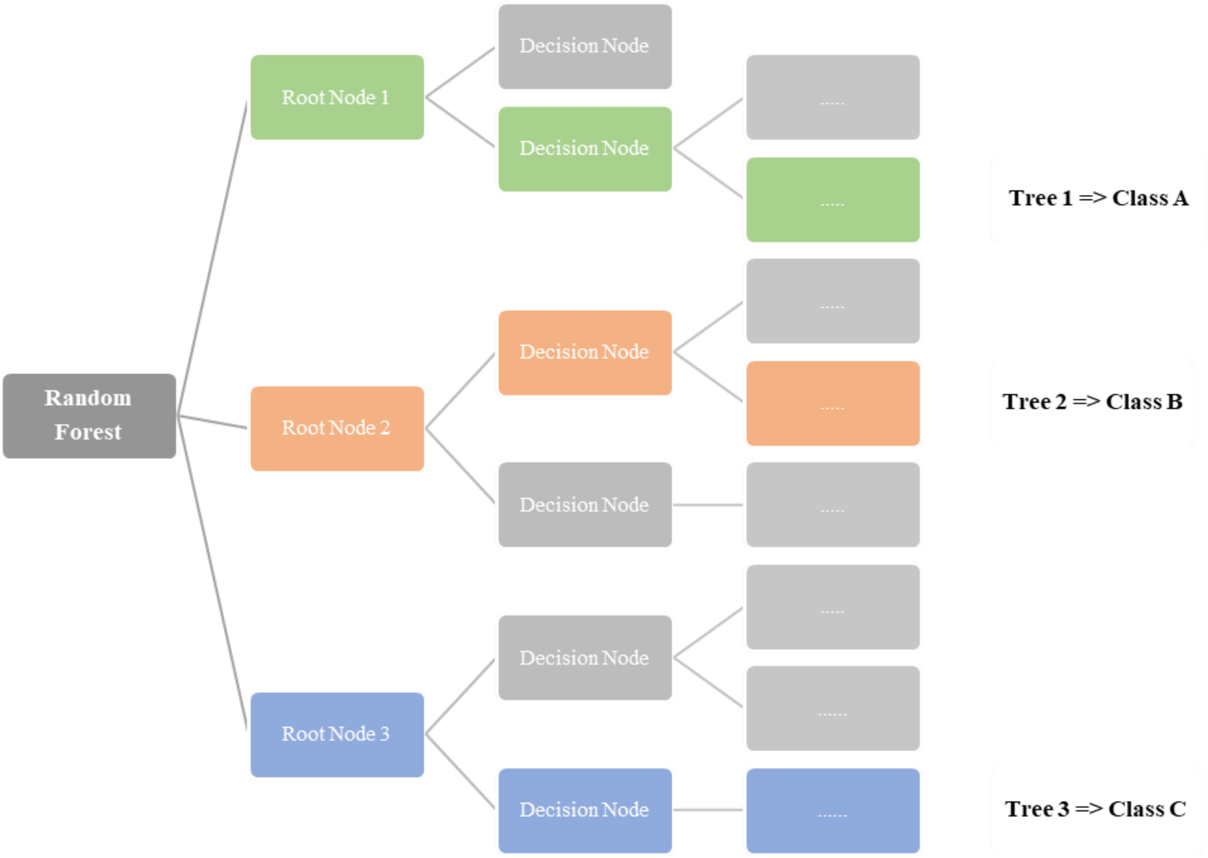


Figure 2.10. Generic Structure of Random Forest Model.

According to [67], [71], [72], the inherent randomness during the training of a RF model outputs a more robust and accurate model, and on top of that, the output RF model appears to be more resistant to overfitting. Apart from that, it does not require proper feature selection, and thus it needs significantly less inputs for each proper running. In [87], the authors showed that a RF classifier performs better, more accurately and efficiently detection of DDoS attacks in IoT networks rather than other classifications algorithms including the SVM, the KNN, and an artificial neural network (ANN) classifiers. The authors in [71] performed their experiments using the Gini index to construct the various DT components, setting to 10 the minimum number of samples per leaf node in order to avoid over fitting, as suggested in [72], demonstrating significant classification results.

2.3.7.2 AdaBoost

Adaptive Boosting or AdaBoost is a statistical classification meta-algorithm (i.e., it is not an ML algorithm by itself, but rather uses other (basic) algorithms to build a stronger one) and is the most widely used and studied for EL, with applications in numerous fields [69], [88]. AdaBoost can be applied in conjunction with many other types of learning algorithms in order to improve performance [89], [90]. The final output of the boosted classifier is represented by the weighted sum of the output of the several other learning algorithms/classifiers, as shown in Figure 2.11, also referred to as “weak learners”. AdaBoost is considered adaptive as the subsequent “weak learners” are tweaked in favor of those instances misclassified by previous classifiers [69], [88]. AdaBoost appears to be less susceptible to the overfitting problem than other learning algorithms, in particular, in classification problems [91]. It is important to highlight that although the individual learners might be weak in terms of performance, as long as their individual performance is slightly better than random guessing, then, the final model can be proven to converge to a strong learner. Attention must be drawn to the fact that every ML algorithm tends to suit better to particular problem types [88], [89], [90], [91]. On top of that, each ML algorithm typically has various parameters and configurations that need to be adjusted in order to achieve optimal performance on a certain dataset.

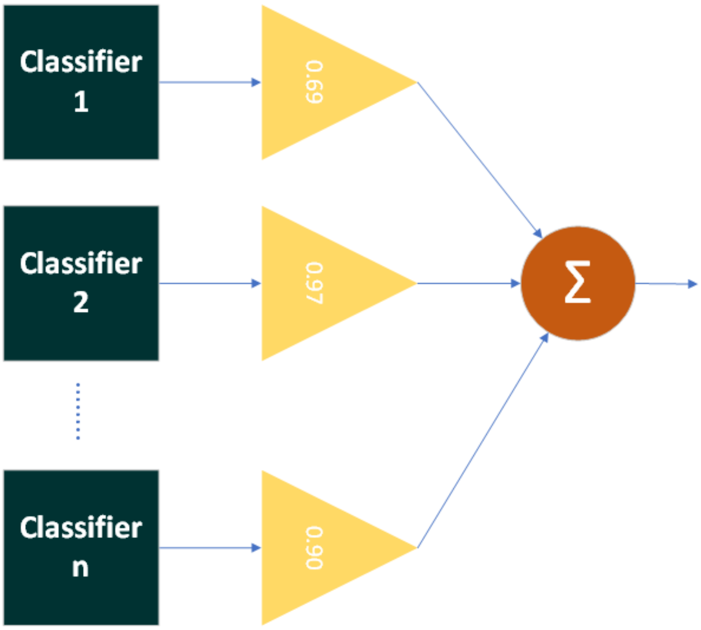


Figure 2.11. An Example of AdaBoost Classifier.

2.3.8 Conclusions

A summary of the main advantages and drawbacks of the reviewed ML algorithms is given below in Table 2.1.

ML Algorithm	Advantages	Drawbacks
Naïve Bayes	▪ Can be used in both binary and multi-class classification. ▪ Simple to use. ▪ Few samples required to train.	▪ The assumption about features independence can lead to low classification accuracy. ▪ “Zero frequency” problem. In the case where a class does not appear during training, it will be assigned a probability of zero.
Decision Tree	▪ Simple to use. ▪ Performance is not different for linearly and non-linearly separated parameters.	▪ Vulnerable to overfitting. ▪ Unstable (i.e., small data variation may result in the construction of extremely different DTs).
Linear Regression	▪ Simple to use. ▪ Computationally efficient. ▪ Overfitting can be reduced by regularization.	▪ Prone to underfitting. ▪ Prone to noise and overfitting. ▪ Sensitive to outliers. ▪ Limited use due to several assumptions that LR takes into consideration for its running.
Logistic Regression	▪ Simple to use. ▪ Easy to implement.	▪ Difficult to perform classification in case of non-linearly separable classes.
Support Vector Machine	▪ Better performance in datasets with few classes and many instances per class. ▪ Scalable. ▪ Reduced storage requirements.	▪ Finding the most appropriate kernel function is a challenge.
K-Nearest Neighbor	▪ Simple to use. ▪ Easy to implement.	▪ Difficult to find the optimal k. ▪ The computational speed decreases as the number of the k variable, the number of data points, or the number of classes increases.
Random Forest	▪ Resistant to overfitting. ▪ Feature selection is performed inherently. ▪ Fewer inputs required.	▪ Fast only in the case of a small number of trees. ▪ May require large datasets.
AdaBoost	▪ Robust to overfitting. ▪ Low computational complexity and error rates.	Sensitive to noisy data and outliers.

Table 2.1. Main advantages and drawbacks of the reviewed ML algorithms.

2.4 Evaluation Metrics

Various metrics are used to evaluate the performance of ML algorithms based on testing datasets. In order to calculate the evaluation metrics, the first step is the calculation of the values of the confusion matrix. The confusion matrix is generated when a trained ML model is used to classify the instances of a testing dataset. The confusion matrix compares values regarding the actual labels of the instances of the testing dataset and the corresponding labels predicted by the ML model. Table 2 shows the 2-by-2 confusion matrix regarding a classification problem with two classes (i.e., normal and attack).

		Predicted Label	
		Positive (Attack)	**Negative (Normal)**
Actual Label	**Positive (Attack)**	True Positive (TP)	False Negative (FN)
	Negative (Normal)	False Positive (FP)	True Negative (TN)

Table 2.2 Confusion Matrix for Binary Classification Problems.

The true positive (TP) and true negative (TN) relate to the correctly classified attack instances and normal instances, respectively. The false positive (FP) and false negative (FN) refer to the incorrectly classified normal instances and attacks instances, respectively. Based on these values, it is possible to compute several evaluation metrics, as shown in [67], [92], [93], [94]. In our case, the metrics of accuracy, precision, recall, and F1-score were used, and each metric is shortly presented below, along with its equation.

- **Accuracy**: shows the overall success of the model by comparing the amount of the correctly classified attack and normal instances to the total amount of instances.

$$\text{Accuracy} = (\text{TP} + \text{TN})/(\text{TP} + \text{TN} + \text{FP} + \text{FN}) \qquad (9)$$

- **Precision**: estimates the overall effectiveness of the model by calculating the percentage that an observation recognized as an attack is actually an attack observation.

$$\text{Precision} = \text{TP}/(\text{TP} + \text{FP}) \qquad (10)$$

- **Recall:** shows the overall success of the model by computing the percentage that an actual attack observation is correctly classified.

$$\text{Recall} = \text{TP}/(\text{TP} + \text{FN}) \qquad (11)$$

- **F1-score**: is calculated by the precision and recall metrics as their harmonious mean. It is a statistical function for estimating the accuracy of the model. As the precision and recall of a model approach the value of 100%, the F1-score and accuracy are maximized, and every instance is classified correctly.

$$\text{F1-score} = 2 \times (\text{Recall} \times \text{Precision})/(\text{Recall} + \text{Precision}) \qquad (12)$$

2.5 Datasets for Anomaly-based Intrusion Detection in IoT Networks

In this Section, the following five of the most well-known existing datasets for training and evaluation of IoT AIDSs are reviewed: (i) the LWSNDR dataset [20], (ii) the dataset presented in [21] for classifying IoT devices using network traffic characteristics, (iii) the "Bot-IoT" dataset [22], (iv) the dataset presented in [23] for detecting DoS attacks on IoT devices using network traffic traces, and (v) the "TON_IoT Telemetry" dataset [14], which is the most recent and representative data-driven IoT/IIoT-based dataset [95].

2.5.1 LWSNDR Dataset

The authors in [20] created two wireless sensor networks (WSNs) in order to serve as testbeds for the simulation of a single-hop sensor-data collection scenario and a multi-hop sensor-data collection scenario, respectively. In both scenarios, Crossbow TelosB motes were used as sensor nodes, and real humidity–temperature sensor data were collected.

In the single-hop scenario, four motes are used as sensor nodes and one mote as the base station node. The four sensor nodes were split into two sets of two nodes, and the first set of nodes collected indoor data, whereas the other set of nodes collected outdoor data. Both sets of sensor nodes transmitted the gathered data to the base station node. In addition, anomalies were introduced to one sensor node in each set (i.e., indoor and outdoor) by utilizing a hot water kettle that alters both the temperature and the humidity simultaneously.

In the multi-hop scenario, four motes are used as sensor nodes, two motes as router nodes, and one mote as the base station node. The router nodes exist in the testbed because the sensor nodes are placed at a distance from where they cannot directly transmit their data to the base station node. The sensor nodes and the router nodes are split in two sets. In each set, two sensor nodes are connected to one router node, whereas the router node connects to the base station node. The two sensor nodes collect humidity–temperature data and send these data to the router node, which then transmits the data to the base station node. The sensor nodes of the first set are responsible for gathering indoor sensor readings, whereas the sensor nodes of the other set collect outdoor sensor readings. Similar to the single-hop scenario, in the multi-hop scenario, anomalies were also introduced to one sensor node in each set (i.e., indoor and outdoor) using a hot water kettle, which leads to an increase in both the temperature and the humidity simultaneously.

In both the single-hop and multi-hop scenarios, real labeled data were generated and were organized in a labelled dataset in order to be used for the purpose of evaluating anomaly detection algorithms. However, the produced dataset (i.e., "LWSNDR" dataset) contains only pure sensor telemetry data, and no information related to either the sensor behavior (e.g., energy consumption) or the network traffic flowing through the WSN is included. In addition, the given dataset does not include any specific attack scenarios, as also mentioned in [14]. Finally, the "LWSNDR" dataset was created in 2010 and cannot be easily considered as recent and representative regarding the current IoT devices or the attacks targeting them.

2.5.2 A Dataset for Classifying IoT Devices Using Network Traffic Characteristics

The authors in [21], designed and developed a robust framework that performs the classification of IoT devices separately, in addition to one class of non-IoT devices, with high accuracy, utilizing statistical attributes derived from network traffic characteristics. One of the authors' contributions was the creation of a smart environment infrastructure that served as a testbed in order to gather and synthesize traffic traces from several IoT devices. The smart environment contains a wide range of IoT devices (i.e., 28 unique IoT devices), non-IoT devices (e.g., smart phones, laptops) and a Wi-Fi access point (i.e., TP-Link access point). The Wi-Fi access point enables the IoT devices and non-IoT devices to communicate with the Internet servers via a gateway [21]. The authors considered the following types of IoT devices: cameras, controllers/hubs, energy management devices (e.g., lights, plugs, motion sensors), appliances, and health-monitors.

Using the created smart environment, traffic traces were collected and synthesized for a period of six months. The traffic traces were collected using the "tcpdump" tool and were stored as "pcap" files on an external USB hard drive of 1 terabyte (TB) storage attached to the gateway. The captured IoT traffic traces comprise (a) traffic produced by the IoT devices without any human interaction (e.g., DNS, NTP), and (b) traffic produced because of the users' interaction with the IoT devices (e.g., motion sensors, lightbulb color change upon user request). Next, the traffic traces were analyzed to gain insight on how to utilize them in order to perform classification of the IoT devices. The analysis of the authors showed that network traffic characteristics, such as activity cycles, port numbers, signaling patterns, and cipher suites, can be exploited in order to properly classify each IoT device.

A subset of these traffic traces was made publicly available as a dataset in order to be used by the scientific community. However, these traffic traces were not generated based on a specific type of attack scenario, and, as a result, they are not representative regarding the behavior of IoT devices or the traffic of IoT networks when under attack.

2.5.3 Bot-IoT Dataset

The authors in [22] generated a dataset, named as the "Bot-IoT" dataset, by incorporating simulated legitimate IoT network traffic, as well as IoT network traffic related to several different types of attacks. In order to generate the "Bot-IoT" dataset, a realistic testbed was developed, with the aim of being representative of an IoT network, and it comprises three components: (i) the network platforms, (ii) the simulated IoT services, and (iii) the extracting features and forensics analytics. Initially, as far as the network platforms of the testbed are concerned, both normal and attacking virtual machines (VMs) with additional network devices (i.e., firewall, tap) were included. Furthermore, the Node-RED tool [96] was employed in order to simulate certain IoT services (e.g., weather station, smart fridge). Finally, regarding the extracting features and forensics analytics, after the authors gathered the normal and attack traffic of the testbed in "pcap" files, they employed the Argus tool in order to extract the flow data and used a MySQL database in order to further process the extracted flow data. Then, statistical models were used in order to identify the most important features for discriminating normal and abnormal instances, and ML techniques were trained and evaluated so as to assess the value of the dataset in comparison to other benchmark datasets [22]. The produced dataset contains both normal and attack network traffic based on benign scenarios and botnet scenarios, respectively. The botnet scenarios include probing, DoS, DDoS, data theft, and keylogging attacks.

The "Bot-IoT" dataset contains over 72 million records of network traffic, and a scaled-down version of the dataset with roughly 3.6 million records is also provided by the authors for evaluation purposes. However, the "Bot-IoT" dataset does not include a variety of attack types (e.g., ransomware and XSS cross-site scripting), as mentioned in [14]. Additionally, the "Bot-IoT" dataset was made available to the scientific community in 2018 and, thus, cannot be easily considered as the most recent and representative dataset containing information about normal or attack traffic of a current IoT network and information about the behavior of IoT devices when they function under normal operation conditions, as well as when they function under attack.

2.5.4 A Dataset for Detecting DoS Attacks on IoT Devices Using Network Traffic Traces

The authors in [23] created an IoT-based dataset by collecting both normal traffic and traffic generated when various types of DoS attacks (e.g., TCP SYN flooding, Ping of Death) were carried out. A testbed was designed and comprises (i) a TPLink gateway with OpenWrt firmware, (ii) several IoT devices (e.g., WeMo motion sensor, Samsung smart-camera, Philips Hue bulb), (iii) two attackers, and (iv) two victims. One attacker was placed locally (inside the LAN) and the other attacker existed remotely (on the Internet). Moreover, both attackers were capable of attacking both victims. In order to store the network packet traces of all of the network traffic, a 1 TB external hard disk was attached to the gateway. The packet traces were stored as “pcap” files using the “tcpdump” tool.

In addition, two types of attacks were implemented: (a) direct attacks (i.e., ARP spoofing, TCP SYN flooding, UDP flooding, and Ping of Death), and (b) reflection attacks (i.e., SNMP, SSDP, TCP SYN, and Smurf). All of the types of DoS attacks were performed using different traffic rates (i.e., how many packets were sent to the victim). Furthermore, the attacks originated from either one of the attackers or both of them and targeted either one of the victims or both of them.

The authors made their dataset available to the community. The released dataset refers to a one-month period of benign and attack traffic relating to ten IoT devices, and annotations of those attacks are included. The dataset consists of 30 “pcap” files, and each file corresponds to a trace collected over a day [23]. Nevertheless, this dataset does not have a variety of attack types (e.g., ransomware and XSS cross-site scripting), as mentioned in [14]. In addition, similarly to the “Bot-IoT” dataset mentioned in Section 2.5.3, this dataset was made available to the community in 2018 and, therefore, cannot be easily considered as the most recent and representative dataset containing information about normal or attack traffic of a current IoT network and information about the behavior of IoT devices when they function under normal operation conditions, as well as when they function under attack.

2.5.5 ToN_IoT Telemetry Dataset

The “TON_IoT Telemetry” dataset includes events of a variety of IoT-related attacks and legitimate scenarios, IoT telemetry data collected from heterogeneous IoT/IIoT data sources, network traffic of the IoT/IIoT network, and audit traces of operating systems. Each of the classes of the “TON_IoT Telemetry” dataset describes either a normal record or the related type of attack in the case of an attack record. In [14], the authors presented the testbed that they developed in order to generate the “TON_IoT Telemetry” dataset [97]. The authors developed a testbed integrating IoT sensors (e.g., weather and modbus sensors), physical network components (e.g., switches, routers), several virtual machines (e.g., VMs of offensive Kali systems, VMs of Windows client systems), hacking platforms, cloud platforms, and fog platforms, and the testbed components were organized into the three layers of “Edge”, “Fog”, and “Cloud”. In addition, the testbed employed a software-defined network (SDN) and network function virtualization (NFV) through the NSX-VMware platform [98]. The NSX-VMware platform enabled: a) the establishment of a virtualized “Fog” layer and a virtualized “Cloud” layer that simultaneously operated to offer the IoT/IIoT and network services; b) the emulation and control of multiple virtual machines (VMs) in the testbed for both hacking and normal operations, and c) the management of the interaction between the three layers.

2.5.5.1 Testbed "Edge" Layer

The "Edge" layer is fundamental in IoT/IIoT applications because its devices measure real-world physical conditions and transmit the collected information to the "Fog" or "Cloud" for further analysis [99]. The "Edge" layer of the testbed contains various IoT/IIoT devices (e.g., weather and light bulb sensors) and physical gateways (i.e., routers and switches) to the Internet, as well as host systems. Besides, the "Edge" layer includes the physical host systems "NSX-VMware Server" and "vSphere System" used to deploy the "Fog" layer and the "Cloud" layer, respectively, by means of virtualization through the NSX-VMware platform [98] and the NSX-VMware hypervisor platform, respectively. The "Edge" layer of the testbed is linked to the "Fog" layer through the "vSwitch".

2.5.5.2 Testbed "Fog" Layer

The purpose of the "Fog" layer is to extend the Cloud computing and services to the "Edge" layer of the network in order to provide limited computing capacity and storage near to the data sources [99]. The "Fog" layer of the testbed consists of the VMs and the virtualization technology that manages the VMs and their services using the NSX-VMware platform [14]. The included VMs and their roles are as follows:

- VMs where the Offensive Kali systems [100] are installed and include the scripts to simulate various attack scenarios;
- VMs (i.e., Metaspoitable3, OWASP security Shepherd, and Damn Vulnerable Web App (DVWA)) which offer vulnerabilities that can be exploited by the Offensive Kali systems [100] ;
- VMs of client systems (i.e., Windows 7 and 10);
- an Ubuntu 18.04 Middleware server where the Node-Red [96] and Mosquitee MQTT broker tools were deployed to manage the IoT/IIoT services and to operate seven IoT/IIoT sensors: weather, smart garage door, smart fridge, smart TCP/IP Modbus, GPS tracker, motion-enabled light, and smart thermostat;
- an Ubuntu 14.04 LTS orchestrated server that offered network services, including DNS (i.e., mydns.com), HTTP(s), DHCP, email server (i.e., Zimbra), Kerberos, and FTP, and generated network traffic between VMs; and
- a VM with the Security Onion tool that is used to log the network data of all the active systems in the testbed.

2.5.5.3 Testbed "Cloud" Layer

The general purpose of the "Cloud" layer is to host large-size data centers with a significant capacity for both computation power and storage in order to support IoT/IIoT applications and satisfy the resource requirements for big data analysis. The "Cloud" layer of the testbed includes:

- a Hive-MQTT broker [101] that is used to publish and subscribe the sensing data of the IoT/IIoT services using the Node-Red tool;
- a vulnerable PHP website [102] used to execute injection attacking events; and
- Cloud centers services (e.g., Microsoft Azure IoT Hub [103] and Amazon Web Services Lambda [104]) that were configured to subscribe and publish IoT/IIoT topics between them and the VMs of the "Fog" layer through the MQTT protocol.

2.5.5.4 ToN_IoT Datasets

The authors in [14] simulated several different types of attack scenarios (i.e., scanning, DoS, DDoS, ransomware, backdoor, data injection, cross-site scripting (XSS), password cracking, and man-in-the-middle (MITM)) on their testbed, and collected data from the different components of their testbed in dataset files. All of the datasets are provided in files that follow the "csv" (comma separated values) format. The datasets files are split into two main folders: (i) the "Processed" datasets folder, and (ii) the "Train_Test" datasets folder.

The "Processed" datasets contain a processed and filtered version of the datasets with: (a) their standard features, (b) a label feature indicating whether an observation is normal or malicious, and (c) a type feature indicating the attacks' sub-classes for multi-class classification problems [14]. On the other hand, the "Train_Test" datasets contain selected records of the "Processed" datasets that were used by the authors in [14] as training and testing datasets for training and evaluating the accuracy and efficiency of various ML algorithms.

Both the "Processed" datasets and the "Train_Test" datasets consist of four types of dataset files (i.e., "Network", "IoT", "Linux", "Windows"), with each referring to either the network traffic or a specific type of device (e.g., sensor, server, desktop) of the testbed, as also demonstrated in Figure 2.12. In particular, the "Network" datasets contain the traffic data that passed through the entire testbed and were captured during the simulations, whereas the "IoT" datasets contain the data related to each of the seven IoT/IIoT sensors that were simulated in the testbed. Finally, the "Linux" datasets and the "Windows" datasets contain the data relating to the two Ubuntu systems and the two Windows systems in the testbed, respectively.

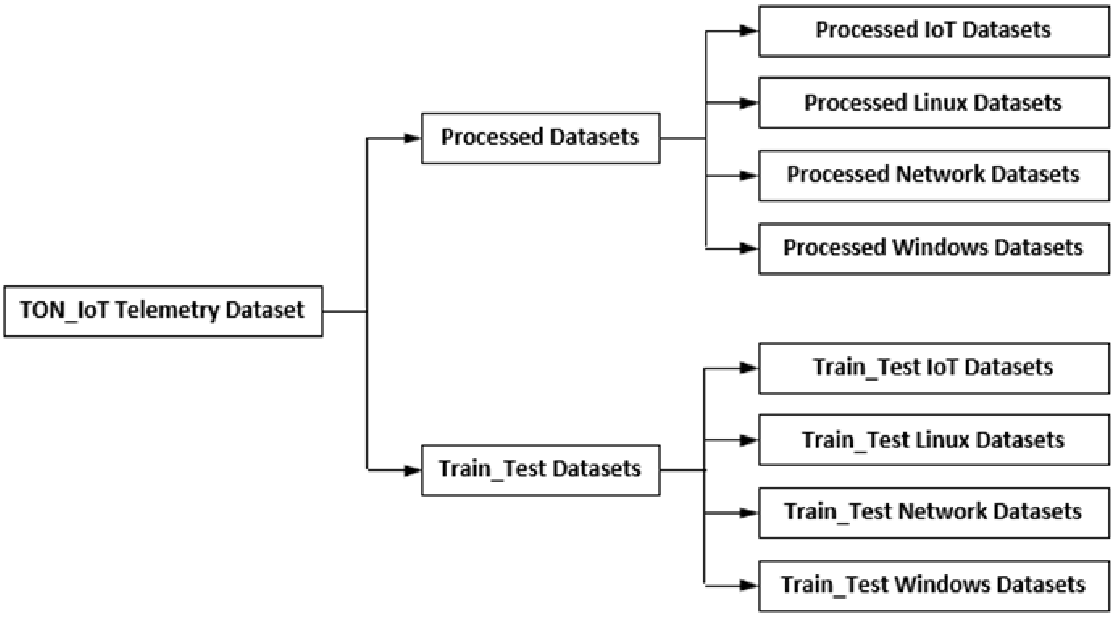

TON_IoT Telemetry Dataset
Processed Datasets
Train_Test Datasets
Processed IoT Datasets
Processed Linux Datasets
Processed Network Datasets
Processed Windows Datasets
Train_Test IoT Datasets
Train_Test Linux Datasets
Train_Test Network Datasets
Train_Test Windows Datasets

Figure 2.12. ToN_IoT Telemetry datasets hierarchy.

2.6 An Overview of Cooja Simulator

Open Source simulators like Cooja have only emerged within the last few years to reflect a new class of tools for simulating/hosting and managing IoT/IIoT based on cloud or remote deployment and an array of features to allow system level deployment. These platforms can be run as simulators in ways that could be considered more representative of deployed systems. Accurate simulation of IoT network nodes is nowadays often coupled to the operating system running on top of the node. Most of the specialised IoT operating systems provide a rather complex simulation environment for researchers and developers. Cooja for Contiki OS is one of the most popular representatives of this class of embedded IoT operating system simulators. COOJA is a flexible Java-based simulator designed for simulating networks of sensors running the Contiki operating system . COOJA is flexible in that many parts of the simulator can be easily replaced or extended with additional functionality [25]. Example parts that can be expanded include the simulated radio medium, simulated node hardware, and plug-ins for simulated input/output. A simulated node in COOJA has three basic properties: its data memory, the node type, and its hardware peripherals. The node type may be shared between several nodes and determines features common to all these nodes. For example, nodes of the same type run the same program code on the same simulated hardware peripherals. And nodes of the same type are initialized with the same data memory. During execution, however, nodes' data memories will come to differ due to for example different external inputs.

COOJA is now able to execute Contiki programs in two different ways. This can be done either by running the program code as compiled native code directly on the host CPU, or by running compiled program code in an instruction-level TI MSP430 emulator. COOJA is also capable of simulating non-ontiki nodes, such as nodes implemented in Java or even nodes running another operating system. All different approaches have advantages as well as shortcomings. Java-based nodes enable much speedier simulations but cannot run deployable code. Hence, they are useful for the development of distributed algorithms. Emulating nodes provides more detailed execution details compared to Java-based nodes or nodes running native code. Finally, native code simulations are more efficient than node emulations and is still able to simulate deployable code. Since the need of abstraction in a heterogeneous simulated network may differ between the different simulated nodes, there are advantages in combining several different abstraction levels in one simulation. For example, in a large, simulated network a few nodes may be simulated at the hardware level while the rest are implemented at the pure Java level. Using this method, it combines the advantages of the different levels. The simulation is faster than when emulating all nodes, but at the same time enables a user to receive fine-grained execution details from the few emulated nodes.

Java-based nodes enable much faster simulations but do not run deployable code. Finally, native code simulations are more efficient than node emulations, and COOJA executes native code by making Java Native Interface calls (JNI) from the Java environment to a compiled Contiki system. The Contiki system comprises of the entire Contiki core, pre-selected user processes, and a set of special simulation glue drivers. Another interesting consequence of using JNI is the ability to debug Contiki code using any regular debugger, such as gdb, by attaching it to the entire Java simulator and breaking when the JNI call is performed. Also, entire simulation states may be saved and later restored, skipping back simulations over time. The hardware peripherals of simulated nodes are called interfaces, and enable the Java simulator to detect and trigger events such as incoming radio traffic or a LED being lit. Interfaces also represent properties of simulated nodes such as positions that the actual node is not aware of. All interactions with simulations and simulated nodes are performed via plugins.

2.7 Summary

In this Chapter, a comprehensive overview of the four main pillars of this PhD research work was given: i) *Internet of Things (IoT)*, ii) *Machine Learning (ML) algorithms for anomaly-based intrusion detection in IoT networks*, iii) *evaluation metrics for the performance of ML algorithms*, and iv) *existing datasets for training and evaluation of anomaly-based intrusion detection in IoT networks*. The Chapter started with an overview of the IoT concept along with its fundamental characteristics and high-level requirements. Afterwards, the three-layer IoT architecture, which is the typical IoT architecture in the literature, was presented where the Perception Layer (i.e., IoT network), the focal point of this PhD research work, was discussed. Following this, an overview of the main security attacks against IoT networks was given. Furthermore, the security and privacy protection requirements for IoT, according to ITU-T Recommendation Y.2066 [28], were presented. Concluding the overview on IoT, concerns that limit the consolidation of secure IoT ecosystems, according to ENISA in [29], were discussed. Next, the most popular ML algorithms used in IoT Anomaly-based Intrusion Detection Systems (AIDS) were reviewed and their main advantages and drawbacks were discussed, followed by the metrics based on which their performance is evaluated. Moreover, five of the most well-known existing datasets for training and evaluation of IoT AIDSs were reviewed. Finally, an overview of Cooja simulator was provided.

This page intentionally left blank.

Chapter 3 Generating Benign IoT Datasets

3.1 Introduction

This Chapter provides a detailed description of the approach followed to generate a set of benign datasets by implementing a benign IoT network scenario in the Cooja simulator [25], as shown in Figure 3.1. The implemented scenario is an example scenario of a benign IoT network, and Cooja has been configured properly to simulate it as described in sections 3.3. The generated IoT-specific information from the simulated scenario was captured from the Contiki plugin “powertrace” (i.e., features such as CPU consumption) and the Cooja tool “Radio messages” (i.e., network traffic features) to generate the “powertrace” dataset and the network traffic dataset, respectively, which constitute the benign datasets for the simulated benign IoT network scenario.

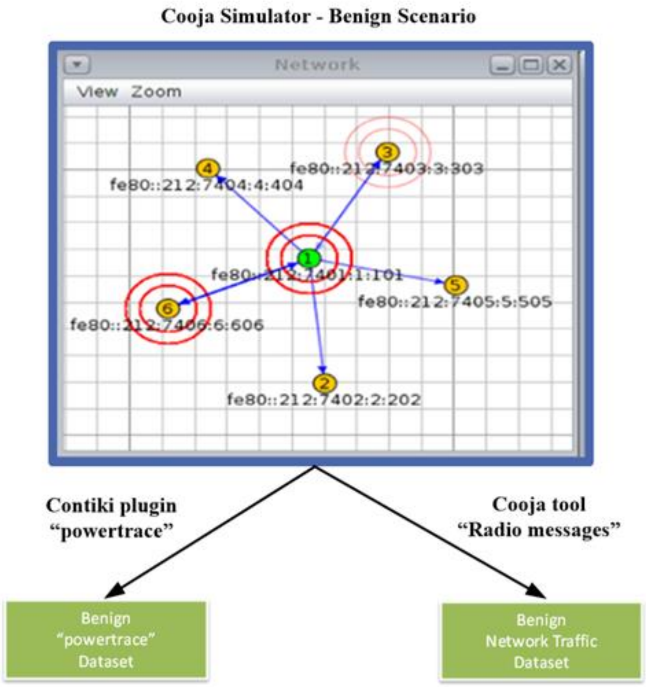
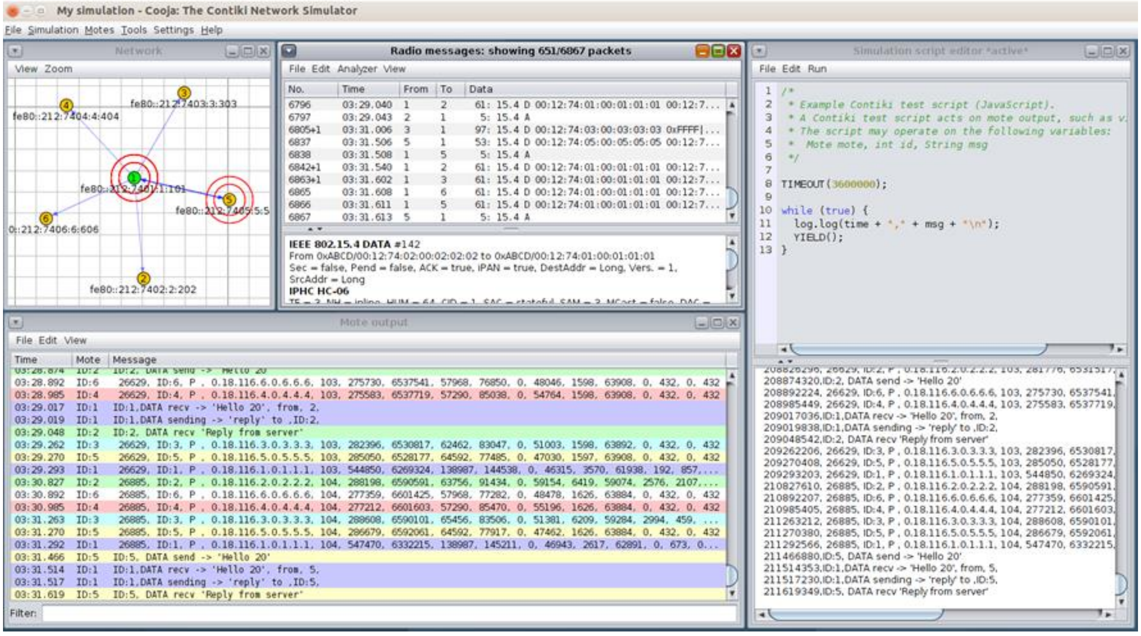


Figure 3.1. Benign IoT datasets generation by utilising the Cooja simulator.

3.2 Benign IoT network scenario – an example

The network topology of the simulated example benign IoT network scenario in the Cooja simulator environment consists of 5 yellow UDP-client motes (i.e., motes 2, 3, 4, 5, and 6) and the green UDP-server mote (i.e., mote 1), as depicted in Figure 3.1. The simulation duration was set to 60 mins and the motes’ outputs were printed out in the respective window (e.g., Mote output) while simulations run, as shown in Figure 3.2. In addition, the yellow UDP-client motes were configured to send text messages every 10 seconds, approximately, to the green UDP-server mote that was configured to provide a corresponding response. The UDP protocol was used at the Transport Layer and the IPv6 at the network layer. Moreover, the type of motes used in this scenario was the Tmote Sky that is an ultra-low power wireless module for use in sensor networks, monitoring applications, and rapid application prototyping. In addition, Tmote Sky motes leverage industry standards such as USB and IEEE 802.15.4 to interoperate seamlessly with other devices. By using industry standards, integrating humidity, temperature, and light sensors, and providing flexible interconnection with peripherals, Tmote Sky motes enable several mesh network applications [105].



```
No.     Time        From  To   Data
6796    03:29.040   1     2    61: 15.4 D 00:12:74:01:00:01:01:01 00:12:7...
6797    03:29.043   2     1     5: 15.4 A
6805+1  03:31.006   3     1    97: 15.4 D 00:12:74:03:00:03:03:03 0xFFFF|...
6837    03:31.506   5     1    53: 15.4 D 00:12:74:05:00:05:05:05 00:12:7...
6838    03:31.508   1     5     5: 15.4 A
6842+1  03:31.540   1     2    61: 15.4 D 00:12:74:01:00:01:01:01 00:12:7...
6863+1  03:31.602   1     3    61: 15.4 D 00:12:74:01:00:01:01:01 00:12:7...
6865    03:31.608   1     6    61: 15.4 D 00:12:74:01:00:01:01:01 00:12:7...
6866    03:31.611   1     5    61: 15.4 D 00:12:74:01:00:01:01:01 00:12:7...
6867    03:31.613   5     1     5: 15.4 A
```

```
IEEE 802.15.4 DATA #142
From 0xABCD/00:12:74:02:00:02:02:02 to 0xABCD/00:12:74:01:00:01:01:01
Sec = false, Pend = false, ACK = true, iPAN = true, DestAddr = Long, Vers. = 1,
SrcAddr = Long
IPHC HC-06
```

```
/*
 * Example Contiki test script (JavaScript).
 * A Contiki test script acts on mote output, such as v
 * The script may operate on the following variables:
 *  Mote mote, int id, String msg
 */

TIMEOUT(3600000);

while (true) {
  log.log(time + "," + msg + "\n");
  YIELD();
}
```

```
Time       Mote  Message
03:28.892  ID:6   26629, ID:6, P , 0.18.116.6.0.6.6.6, 103, 275730, 6537541, 57968, 76850, 0, 48046, 1598, 63908, 0, 432, 0, 432
03:28.985  ID:4   26629, ID:4, P , 0.18.116.4.0.4.4.4, 103, 275583, 6537719, 57290, 85038, 0, 54764, 1598, 63908, 0, 432, 0, 432
03:29.017  ID:1  ID:1,DATA recv -> 'Hello 20', from, 2,
03:29.019  ID:1  ID:1,DATA sending -> 'reply' to ,ID:2,
03:29.048  ID:2  ID:2, DATA recv 'Reply from server'
03:29.262  ID:3   26629, ID:3, P , 0.18.116.3.0.3.3.3, 103, 282396, 6530817, 62462, 83047, 0, 51003, 1598, 63892, 0, 432, 0, 432
03:29.270  ID:5   26629, ID:5, P , 0.18.116.5.0.5.5.5, 103, 285050, 6528177, 64592, 77485, 0, 47030, 1597, 63908, 0, 432, 0, 432
03:29.293  ID:1   26629, ID:1, P , 0.18.116.1.0.1.1.1, 103, 544850, 6269324, 138987, 144538, 0, 46315, 3570, 61938, 192, 857,...
03:30.827  ID:2   26885, ID:2, P , 0.18.116.2.0.2.2.2, 104, 288198, 6590591, 63756, 91434, 0, 59154, 6419, 59074, 2576, 2107,...
03:30.892  ID:6   26885, ID:6, P , 0.18.116.6.0.6.6.6, 104, 277359, 6601425, 57968, 77282, 0, 48478, 1626, 63884, 0, 432, 0, 432
03:30.985  ID:4   26885, ID:4, P , 0.18.116.4.0.4.4.4, 104, 277212, 6601603, 57290, 85470, 0, 55196, 1626, 63884, 0, 432, 0, 432
03:31.263  ID:3   26885, ID:3, P , 0.18.116.3.0.3.3.3, 104, 288608, 6590101, 65456, 83506, 0, 51381, 6209, 59284, 2994, 459, ...
03:31.270  ID:5   26885, ID:5, P , 0.18.116.5.0.5.5.5, 104, 286679, 6592061, 64592, 77917, 0, 47462, 1626, 63884, 0, 432, 0, 432
03:31.292  ID:1   26885, ID:1, P , 0.18.116.1.0.1.1.1, 104, 547470, 6332215, 138987, 145211, 0, 46943, 2617, 62891, 0, 673, 0...
03:31.466  ID:5  ID:5, DATA send -> 'Hello 20'
03:31.514  ID:1  ID:1,DATA recv -> 'Hello 20', from, 5,
03:31.517  ID:1  ID:1,DATA sending -> 'reply' to ,ID:5,
03:31.619  ID:5  ID:5, DATA recv 'Reply from server'
```

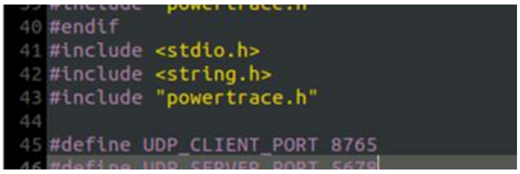
```
208874320,ID:2, DATA send -> 'Hello 20'
208892224, 26629, ID:6, P , 0.18.116.6.0.6.6.6, 103, 275730, 6537541,
208985449, 26629, ID:4, P , 0.18.116.4.0.4.4.4, 103, 275583, 6537719,
209017036,ID:1,DATA recv -> 'Hello 20', from, 2,
209019838,ID:1,DATA sending -> 'reply' to ,ID:2,
209048542,ID:2, DATA recv 'Reply from server'
209262206, 26629, ID:3, P , 0.18.116.3.0.3.3.3, 103, 282396, 6530817,
209270408, 26629, ID:5, P , 0.18.116.5.0.5.5.5, 103, 285050, 6528177,
209293203, 26629, ID:1, P , 0.18.116.1.0.1.1.1, 103, 544850, 6269324,
210827610, 26885, ID:2, P , 0.18.116.2.0.2.2.2, 104, 288198, 6590591,
210892207, 26885, ID:6, P , 0.18.116.6.0.6.6.6, 104, 277359, 6601425,
210985405, 26885, ID:4, P , 0.18.116.4.0.4.4.4, 104, 277212, 6601603,
211263212, 26885, ID:3, P , 0.18.116.3.0.3.3.3, 104, 288608, 6590101,
211270380, 26885, ID:5, P , 0.18.116.5.0.5.5.5, 104, 286679, 6592061,
211292566, 26885, ID:1, P , 0.18.116.1.0.1.1.1, 104, 547470, 6332215,
211466880,ID:5, DATA send -> 'Hello 20'
211514353,ID:1,DATA recv -> 'Hello 20', from, 5,
211517230,ID:1,DATA sending -> 'reply' to ,ID:5,
211619349,ID:5, DATA recv 'Reply from server'
```

Figure 3.2 Cooja Simulator – motes' outputs

3.3 Benign "powertrace" Dataset

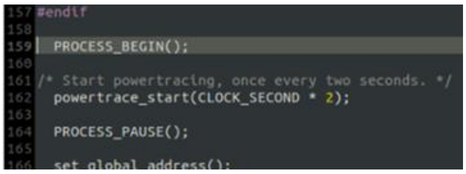
3.3.1 Benign "powertrace" Dataset – Generation Process

The "powertrace" dataset includes information about features such as such as total CPU energy consumption and low power mode (LPM) energy consumption. In fact, it is the dataset of the simulated benign IoT network scenario that includes records about information related to the energy consumption of the IoT devices (i.e., motes) deployed within the simulated IoT network. To enable the "powertrace" plugin and generate the "powertrace" dataset, the motes of the benign IoT network were programmed to make use of the "powertrace" plugin for collecting "powertrace" related features every 2 seconds. In particular, we included the "powertrace.h" library into the code of each mote (i.e. #include "powertrace.h"), as shown in Figure 3.3, and defined to start powertracing, once every 2 seconds, in the code of each mote as shown in Figure 3.4.



```
#endif
#include <stdio.h>
#include <string.h>
#include "powertrace.h"

#define UDP_CLIENT_PORT 8765
```

Figure 3.3 "powertrace.h" library in the mote code.



```
#endif

  PROCESS_BEGIN();

/* Start powertracing, once every two seconds. */
  powertrace_start(CLOCK_SECOND * 2);

  PROCESS_PAUSE();

  set_global_address();
```

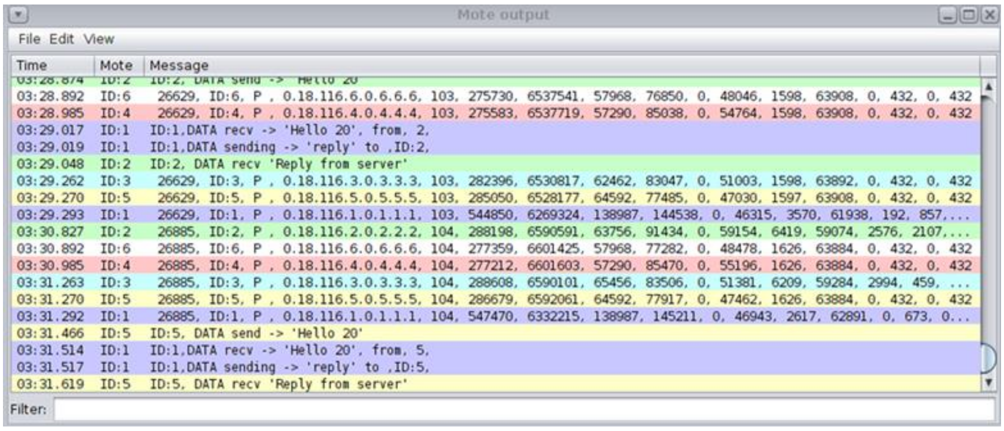
Figure 3.4 Powertracing Begin.

More precisely, the "powertrace" plugin captured raw information, every 2 seconds, about the set of features summarised in Table 3.1. In particular, the "powertrace" plugin tracks the duration (i.e., number of cpu ticks) of activities of a mote being in each power state. Particularly, the outputs demonstrate the fraction of time in which a mote remains in a given power state. There are the following six power states: i) cpu; ii) lpm; iii) transmit; iv) listen; v) idle_transmit; and vi) idle_listen, as shown in Table 3.1. These are measured with a hardware timer (i.e., clock frequency is defined in RTIMER_SECOND or 32,768 Hz for XM1000). In addition, it is worthwhile mentioning that in our simulated scenarios the value range for the following features was between 0 and 65535: cpu, lpm, transmit, listen, idle_transmit, idle_listen. This is because our acquisition time was 2 seconds and the hardware_timer is 32,768. Besides that, the value ranges for rimeaddr and seqno are dependent on the number of motes included in each simulated scenario, and the number of acquired samples during the monitoring time.

Index	Feature	Description
1	sim time	simulation time
2	clock_time()	clock time (i.e., by default, 128 ticks/second)
3	ID	Mote ID
4	P	label
5	rimeaddr	rime address
6	seqno	sequence number
7	all_cpu	accumulated CPU energy consumption
8	all_lpm	accumulated Low Power Mode energy consumption
9	all_transmit	accumulated transmission energy consumption
10	all_listen	accumulated listen energy consumption
11	all_idle_transmit	accumulated idle transmission energy consumption
12	all_idle_listen	accumulated idle listen energy consumption
13	cpu	CPU energy consumption for this cycle
14	lpm	LPM energy consumption for this cycle
15	transmit	transmission energy consumption for this cycle
16	listen	listen energy consumption for this cycle
17	idle_transmit	idle transmission energy consumption for this cycle
18	idle_listen	idle listen energy consumption for this cycle

Table 3.1 Set of Captured Features by “powertrace” plugin.

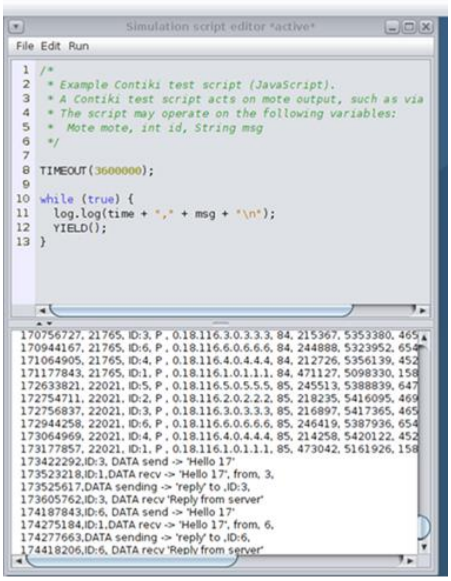
In Figure 3.1, the depicted Mote output window displays the captured “powertrace” information every 2 seconds and also the messages sent/received by each mote (printouts/printf from each mote).



```
Time       Mote   Message
03:28.874  ID:2   ID:2, DATA send -> 'Hello 20'
03:28.892  ID:6    26629, ID:6, P , 0.18.116.6.0.6.6.6, 103, 275730, 6537541, 57968, 76850, 0, 48046, 1598, 63908, 0, 432, 0, 432
03:28.985  ID:4    26629, ID:4, P , 0.18.116.4.0.4.4.4, 103, 275583, 6537719, 57290, 85038, 0, 54764, 1598, 63908, 0, 432, 0, 432
03:29.017  ID:1   ID:1,DATA recv -> 'Hello 20', from, 2,
03:29.019  ID:1   ID:1,DATA sending -> 'reply' to ,ID:2,
03:29.048  ID:2   ID:2, DATA recv 'Reply from server'
03:29.262  ID:3    26629, ID:3, P , 0.18.116.3.0.3.3.3, 103, 282396, 6530817, 62462, 83047, 0, 51003, 1598, 63892, 0, 432, 0, 432
03:29.270  ID:5    26629, ID:5, P , 0.18.116.5.0.5.5.5, 103, 285050, 6528177, 64592, 77485, 0, 47030, 1597, 63908, 0, 432, 0, 432
03:29.293  ID:1    26629, ID:1, P , 0.18.116.1.0.1.1.1, 103, 544850, 6269324, 138987, 144538, 0, 46315, 3570, 61938, 192, 857,...
03:30.827  ID:2    26885, ID:2, P , 0.18.116.2.0.2.2.2, 104, 288198, 6590591, 63756, 91434, 0, 59154, 6419, 59074, 2576, 2107,...
03:30.892  ID:6    26885, ID:6, P , 0.18.116.6.0.6.6.6, 104, 277359, 6601425, 57968, 77282, 0, 48478, 1626, 63884, 0, 432, 0, 432
03:30.985  ID:4    26885, ID:4, P , 0.18.116.4.0.4.4.4, 104, 277212, 6601603, 57290, 85470, 0, 55196, 1626, 63884, 0, 432, 0, 432
03:31.263  ID:3    26885, ID:3, P , 0.18.116.3.0.3.3.3, 104, 288608, 6590101, 65456, 83506, 0, 51381, 6209, 59284, 2994, 459, ...
03:31.270  ID:5    26885, ID:5, P , 0.18.116.5.0.5.5.5, 104, 286679, 6592061, 64592, 77917, 0, 47462, 1626, 63884, 0, 432, 0, 432
03:31.292  ID:1    26885, ID:1, P , 0.18.116.1.0.1.1.1, 104, 547470, 6332215, 138987, 145211, 0, 46943, 2617, 62891, 0, 673, 0...
03:31.466  ID:5   ID:5, DATA send -> 'Hello 20'
03:31.514  ID:1   ID:1,DATA recv -> 'Hello 20', from, 5,
03:31.517  ID:1   ID:1,DATA sending -> 'reply' to ,ID:5,
03:31.619  ID:5   ID:5, DATA recv 'Reply from server'
```

Figure 3.5 Cooja Simulator—Mote output window.

Furthermore, the Simulation script editor, shown in Figure 3.6, is a Cooja tool used to display messages and set a timer on the simulation. As shown in Figure 3.6, the upper part of the Simulation script editor was used to create scripts and the lower part to show the captured “powertrace” information and the printouts (i.e., printf messages) from the motes until the timeout occurs. In our implementation, we considered the simulation duration to be 60 mins and thus, the timeout was set at 3,600,000 ms. When the timeout occurred, the simulation stopped, and all the captured information and prints were stored in the log file named “COOJA.testlog”.

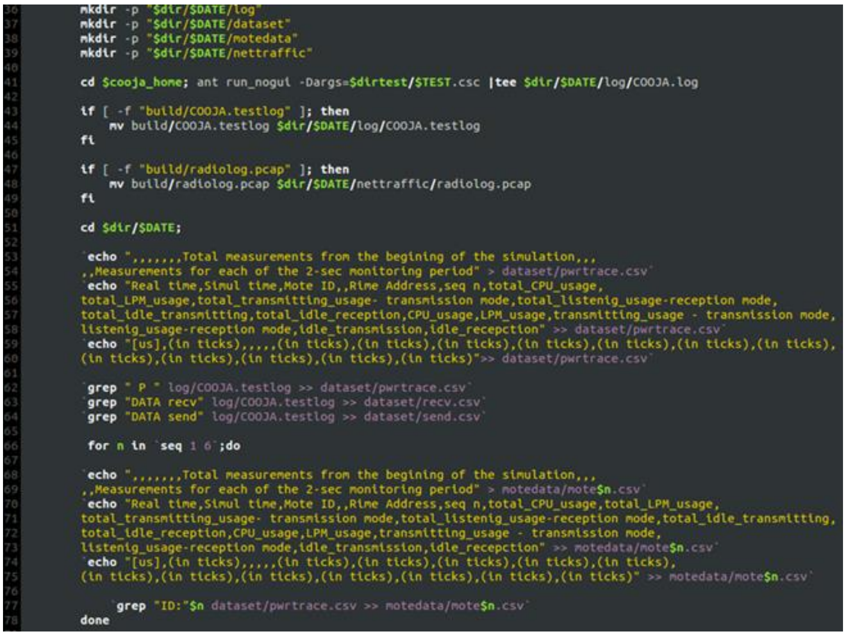


```
/*
 * Example Contiki test script (JavaScript).
 * A Contiki test script acts on mote output, such as via
 * The script may operate on the following variables:
 *  Mote mote, int id, String msg
 */

TIMEOUT(3600000);

while (true) {
  log.log(time + "," + msg + "\n");
  YIELD();
}
```

```
170756727, 21765, ID:3, P , 0.18.116.3.0.3.3.3, 84, 215367, 5353380, 465
170944167, 21765, ID:6, P , 0.18.116.6.0.6.6.6, 84, 244888, 5323952, 654
171064905, 21765, ID:4, P , 0.18.116.4.0.4.4.4, 84, 212726, 5356139, 452
171177843, 21765, ID:1, P , 0.18.116.1.0.1.1.1, 84, 471127, 5098330, 158
172633821, 22021, ID:5, P , 0.18.116.5.0.5.5.5, 85, 245513, 5388839, 647
172754711, 22021, ID:2, P , 0.18.116.2.0.2.2.2, 85, 218235, 5416095, 469
172756837, 22021, ID:3, P , 0.18.116.3.0.3.3.3, 85, 216897, 5417365, 465
172944258, 22021, ID:6, P , 0.18.116.6.0.6.6.6, 85, 246419, 5387936, 654
173064969, 22021, ID:4, P , 0.18.116.4.0.4.4.4, 85, 214258, 5420122, 452
173177857, 22021, ID:1, P , 0.18.116.1.0.1.1.1, 85, 473042, 5161926, 158
173422292,ID:3, DATA send -> 'Hello 17'
173523218,ID:1,DATA recv -> 'Hello 17', from, 3,
173525617,DATA sending -> 'reply' to ,ID:3,
173605762,ID:3, DATA recv 'Reply from server'
174187843,ID:6, DATA send -> 'Hello 17'
174275184,ID:1,DATA recv -> 'Hello 17', from, 6,
174277663,DATA sending -> 'reply' to ,ID:6,
174418206,ID:6, DATA recv 'Reply from server'
```

Figure 3.6 Simulation script editor.

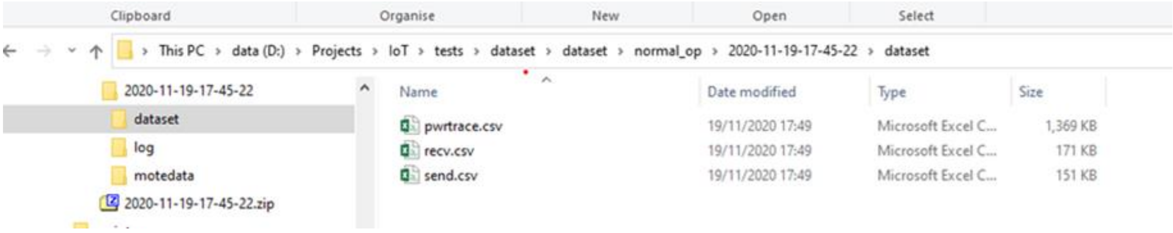
Having collected all the captured raw information from the “powertrace” plugin in the “COOJA.testlog” file, the challenging task was to extract this information from the “COOJA.testlog” file to a csv file that would be the “powertrace” dataset of the simulated benign IoT network scenario including records about the energy consumption of the motes. To address this challenge, the “IoT_Simul.sh” bash file was developed to extract all the required “powertrace” information from the “COOJA.testlog” file to the “pwrtrace.csv” file. An extract of the “IoT_Simul.sh” bash file is shown in Figure 3.7.



```
mkdir -p "$dir/$DATE/log"
mkdir -p "$dir/$DATE/dataset"
mkdir -p "$dir/$DATE/motedata"
mkdir -p "$dir/$DATE/nettraffic"

cd $cooja_home; ant run_nogui -Dargs=$dirtest/$TEST.csc |tee $dir/$DATE/log/COOJA.log

if [ -f "build/COOJA.testlog" ]; then
    mv build/COOJA.testlog $dir/$DATE/log/COOJA.testlog
fi

if [ -f "build/radiolog.pcap" ]; then
    mv build/radiolog.pcap $dir/$DATE/nettraffic/radiolog.pcap
fi

cd $dir/$DATE;

 `echo ",,,,,,,Total measurements from the begining of the simulation,,,
,,Measurements for each of the 2-sec monitoring period" > dataset/pwrtrace.csv`
 `echo "Real time,Simul time,Mote ID,,Rime Address,seq n,total_CPU_usage,
total_LPM_usage,total_transmitting_usage- transmission mode,total_listenig_usage-reception mode,
total_idle_transmitting,total_idle_reception,CPU_usage,LPM_usage,transmitting_usage - transmission mode,
listenig_usage-reception mode,idle_transmission,idle_recepction" >> dataset/pwrtrace.csv`
 `echo "[us],(in ticks),,,,,(in ticks),(in ticks),(in ticks),(in ticks),(in ticks),(in ticks),(in ticks),
(in ticks),(in ticks),(in ticks),(in ticks),(in ticks)">> dataset/pwrtrace.csv`

 `grep " P " log/COOJA.testlog >> dataset/pwrtrace.csv`
 `grep "DATA recv" log/COOJA.testlog >> dataset/recv.csv`
 `grep "DATA send" log/COOJA.testlog >> dataset/send.csv`

 for n in `seq 1 6`;do

 `echo ",,,,,,,Total measurements from the begining of the simulation,,,
,,Measurements for each of the 2-sec monitoring period" > motedata/mote$n.csv`
 `echo "Real time,Simul time,Mote ID,,Rime Address,seq n,total_CPU_usage,total_LPM_usage,
total_transmitting_usage- transmission mode,total_listenig_usage-reception mode,total_idle_transmitting,
total_idle_reception,CPU_usage,LPM_usage,transmitting_usage - transmission mode,
listenig_usage-reception mode,idle_transmission,idle_recepction" >> motedata/mote$n.csv`
 `echo "[us],(in ticks),,,,,(in ticks),(in ticks),(in ticks),(in ticks),(in ticks),
(in ticks),(in ticks),(in ticks),(in ticks),(in ticks),(in ticks),(in ticks)" >> motedata/mote$n.csv`

     `grep "ID:"$n dataset/pwrtrace.csv >> motedata/mote$n.csv`
done
```

Figure 3.7 Extract of the “IoT_Simul.sh” file.

Initially, the “IoT_Simul.sh” file created the root folder named with the simulation date and time (i.e., “2020-11-19-17-45-22” folder), as shown below in the left part of Figure 3.8. Afterwards, the bash file created the “log” folder, inside the “2020-11-19-17-45-22” folder, where the “COOJA.testlog” file was copied from the “…/cooja/build” folder located in the Cooja Simulator environment.


Clipboard
Organise
New
Open
Select
This PC › data (D:) › Projects › IoT › tests › dataset › dataset › normal_op › 2020-11-19-17-45-22 › dataset
2020-11-19-17-45-22
dataset
log
motedata
2020-11-19-17-45-22.zip

Name	Date modified	Type	Size
pwrtrace.csv	19/11/2020 17:49	Microsoft Excel C...	1,369 KB
recv.csv	19/11/2020 17:49	Microsoft Excel C...	171 KB
send.csv	19/11/2020 17:49	Microsoft Excel C...	151 KB

Figure 3.8 Location of the generated “pwrtrace.csv”, “recv.csv”, and “send.csv” files by the “IoT_Simul.sh” file.

In addition, in the “IoT_Simul.sh” file, the Linux tool “grep” was used to extract the required “powertrace” information by selecting the label “P” in each “powertrace” row from the “COOJA.testlog” file and save it in the “pwrtrace.csv” file in the “dataset” folder that was also created by the batch file inside the “2020-11-19-17-45-22” folder, as shown in the left part of Figure 3.8. In particular, it was implemented with the following command:

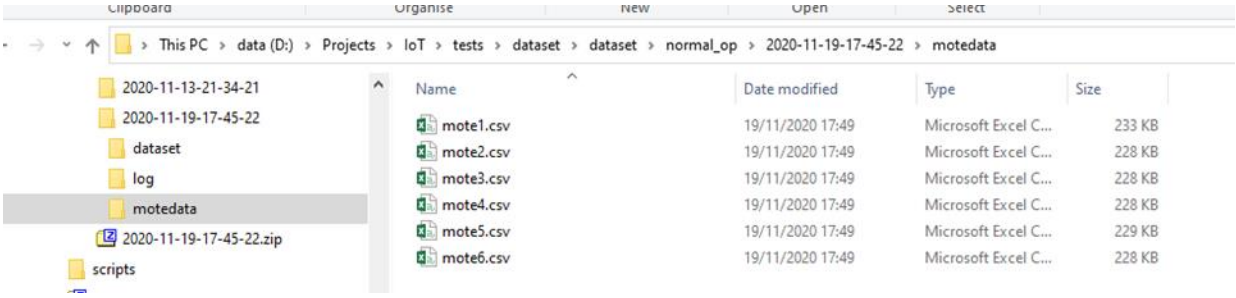
```
grep "P" log/COOJA.testlog >> dataset/pwrtrace.csv
```

However, in the “dataset” folder, apart from the “pwrtrace.csv” file, the “IoT_Simul.sh” file generated two more files, based on the information included in the “COOJA.testlog” file, as shown in Figure 3.8; the “recv.csv” file and the “send.csv” file that include the “received” and “sent”messages printed by the motes, respectively.

Finally, the “IoT_Simul.sh” file extracted the information related to each mote, from the “pwrtrace.csv” file, and generated one csv file for each mote with the corresponding information from “pwrtrace.csv” file. It was implemented with the following command, where “n” is the mote number (i.e., 1 to 6):

```
grep "ID:"$n dataset/pwrtrace.csv >> motedata/mote$n.csv
```

The generated 6 csv files (i.e., mote1.csv, mote2.csv, mote3.csv, mote4.csv, mote5.csv, mote6.csv) were stored in the “motedata” folder, as shown in Figure . The “motedata” folder was also created by the “IoT_Simul.sh” file inside the “2020-11-19-17-45-22” folder.



Name	Date modified	Type	Size
mote1.csv	19/11/2020 17:49	Microsoft Excel C...	233 KB
mote2.csv	19/11/2020 17:49	Microsoft Excel C...	228 KB
mote3.csv	19/11/2020 17:49	Microsoft Excel C...	228 KB
mote4.csv	19/11/2020 17:49	Microsoft Excel C...	228 KB
mote5.csv	19/11/2020 17:49	Microsoft Excel C...	229 KB
mote6.csv	19/11/2020 17:49	Microsoft Excel C...	228 KB

Figure 3.9 Location of the generated “mote1.csv”, “mote2.csv”, “mote3.csv”, “mote4.csv”, “mote5.csv”, “mote6.csv files” by the “IoT_Simul.sh” bash file.

An overview of the described process followed to extract the required information from the "COOJA.testlog" file to the "pwrtrace.csv", "recv.csv", "send.csv", "mote1.csv", "mote2.csv", "mote3.csv", "mote4.csv", "mote5.csv", and "mote6.csv" files are depicted in Figure .

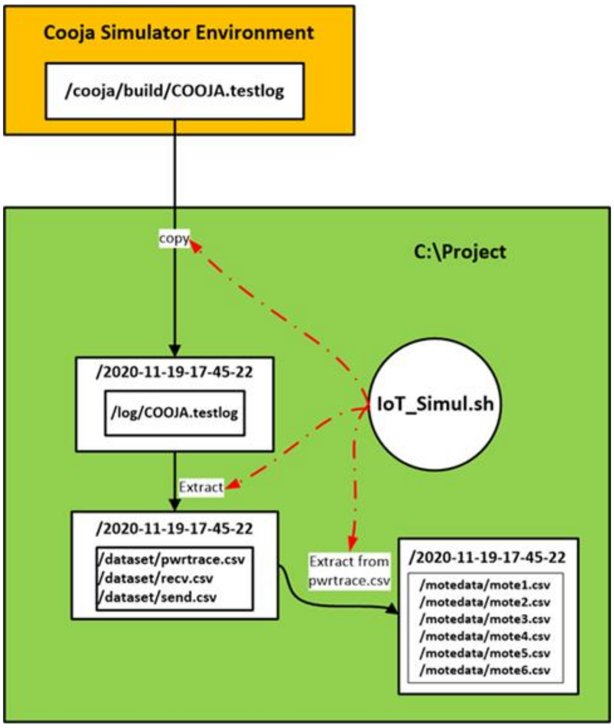


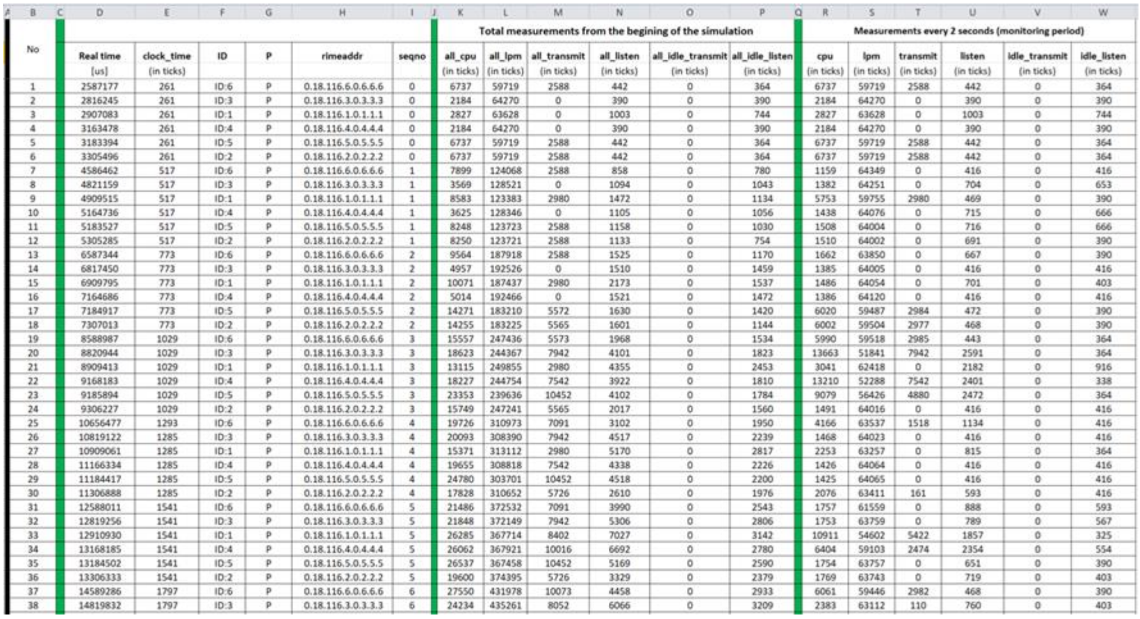
Figure 3.10 An overview of the process followed by the "IoT_Simul.sh" file to extract all the required "powertrace" information from the "COOJA.testlog" file.

3.3.2 Benign "powertrace" Dataset – Generated Results

The "powertrace" dataset consists of the following csv files: "pwrtrace.csv", "mote1.csv", "mote2.csv", "mote3.csv", "mote4.csv", "mote5.csv", and "mote6.csv" files. In this Section, we present sets of records from the "pwrtrace.csv", and in Appendix 1 we present sets of records from "mote1.csv", and "mote3.csv" files.

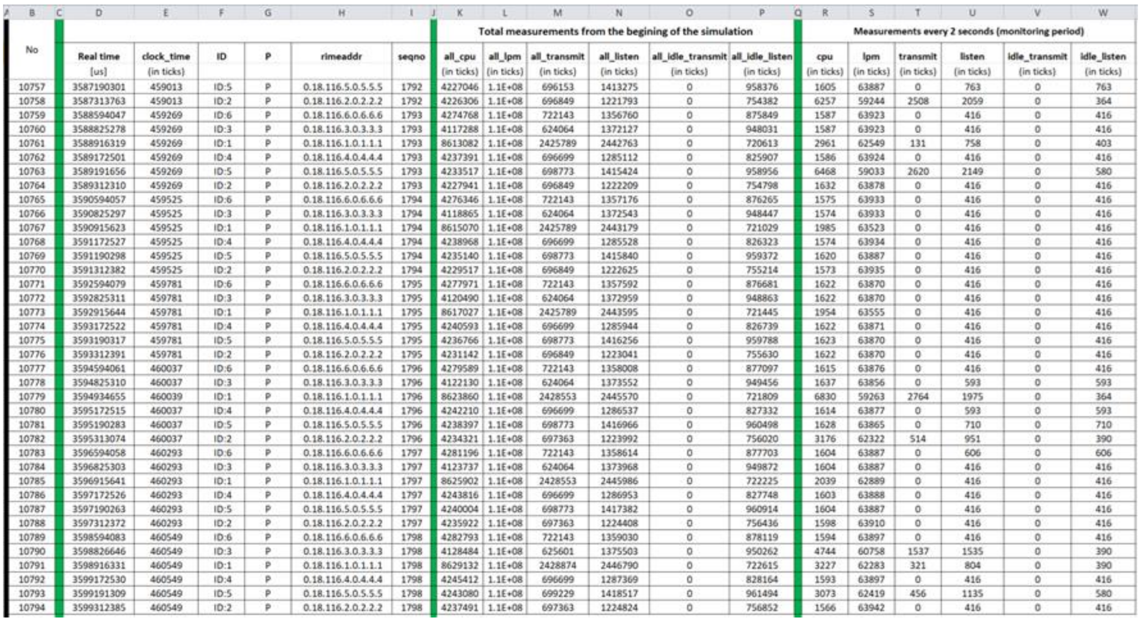
3.3.2.1 Benign "pwrtrace.csv"

The generated benign "pwrtrace.csv" file consists of 10,794 records and its first 38 records (i.e., 1–38) and its last 38 records (10,757–10,794) are depicted in Figure 3.11 and Figure 3.12, respectively.



B	D	E	F	G	H	I	K	L	M	N	O	P	R	S	T	U	V	W
No							Total measurements from the begining of the simulation						Measurements every 2 seconds (monitoring period)					
	Real time	clock_time	ID	P	rimeaddr	seqno	all_cpu	all_lpm	all_transmit	all_listen	all_idle_transmit	all_idle_listen	cpu	lpm	transmit	listen	idle_transmit	idle_listen
	[us]	(in ticks)					(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)
1	2587177	261	ID:6	P	0.18.116.6.0.6.6.6	0	6737	59719	2588	442	0	364	6737	59719	2588	442	0	364
2	2816245	261	ID:3	P	0.18.116.3.0.3.3.3	0	2184	64270	0	390	0	390	2184	64270	0	390	0	390
3	2907083	261	ID:1	P	0.18.116.1.0.1.1.1	0	2827	63628	0	1003	0	744	2827	63628	0	1003	0	744
4	3163478	261	ID:4	P	0.18.116.4.0.4.4.4	0	2184	64270	0	390	0	390	2184	64270	0	390	0	390
5	3183394	261	ID:5	P	0.18.116.5.0.5.5.5	0	6737	59719	2588	442	0	364	6737	59719	2588	442	0	364
6	3305496	261	ID:2	P	0.18.116.2.0.2.2.2	0	6737	59719	2588	442	0	364	6737	59719	2588	442	0	364
7	4586462	517	ID:6	P	0.18.116.6.0.6.6.6	1	7899	124068	2588	858	0	780	1159	64349	0	416	0	416
8	4821159	517	ID:3	P	0.18.116.3.0.3.3.3	1	3569	128521	0	1094	0	1043	1382	64251	0	704	0	653
9	4909515	517	ID:1	P	0.18.116.1.0.1.1.1	1	8583	123383	2980	1472	0	1134	5753	59755	2980	469	0	390
10	5164736	517	ID:4	P	0.18.116.4.0.4.4.4	1	3625	128346	0	1105	0	1056	1438	64076	0	715	0	666
11	5183527	517	ID:5	P	0.18.116.5.0.5.5.5	1	8248	123723	2588	1158	0	1030	1508	64004	0	716	0	666
12	5305285	517	ID:2	P	0.18.116.2.0.2.2.2	1	8250	123721	2588	1133	0	754	1510	64002	0	691	0	390
13	6587344	773	ID:6	P	0.18.116.6.0.6.6.6	2	9564	187918	2588	1525	0	1170	1662	63850	0	667	0	390
14	6817450	773	ID:3	P	0.18.116.3.0.3.3.3	2	4957	192526	0	1510	0	1459	1385	64005	0	416	0	416
15	6909795	773	ID:1	P	0.18.116.1.0.1.1.1	2	10071	187437	2980	2173	0	1537	1486	64054	0	701	0	403
16	7164686	773	ID:4	P	0.18.116.4.0.4.4.4	2	5014	192466	0	1521	0	1472	1386	64120	0	416	0	416
17	7184917	773	ID:5	P	0.18.116.5.0.5.5.5	2	14271	183210	5572	1630	0	1420	6020	59487	2984	472	0	390
18	7307013	773	ID:2	P	0.18.116.2.0.2.2.2	2	14255	183225	5565	1601	0	1144	6002	59504	2977	468	0	390
19	8588987	1029	ID:6	P	0.18.116.6.0.6.6.6	3	15557	247436	5573	1968	0	1534	5990	59518	2985	443	0	364
20	8820944	1029	ID:3	P	0.18.116.3.0.3.3.3	3	18623	244367	7942	4101	0	1823	13663	51841	7942	2591	0	364
21	8909413	1029	ID:1	P	0.18.116.1.0.1.1.1	3	13115	249855	2980	4355	0	2453	3041	62418	0	2182	0	916
22	9168183	1029	ID:4	P	0.18.116.4.0.4.4.4	3	18227	244754	7542	3922	0	1810	13210	52288	7542	2401	0	338
23	9185894	1029	ID:5	P	0.18.116.5.0.5.5.5	3	23353	239636	10452	4102	0	1784	9079	56426	4880	2472	0	364
24	9306227	1029	ID:2	P	0.18.116.2.0.2.2.2	3	15749	247241	5565	2017	0	1560	1491	64016	0	416	0	416
25	10656477	1293	ID:6	P	0.18.116.6.0.6.6.6	4	19726	310973	7091	3102	0	1950	4166	63537	1518	1134	0	416
26	10819122	1285	ID:3	P	0.18.116.3.0.3.3.3	4	20093	308390	7942	4517	0	2239	1468	64023	0	416	0	416
27	10909061	1285	ID:1	P	0.18.116.1.0.1.1.1	4	15371	313112	2980	5170	0	2817	2253	63257	0	815	0	364
28	11166334	1285	ID:4	P	0.18.116.4.0.4.4.4	4	19655	308818	7542	4338	0	2226	1426	64064	0	416	0	416
29	11184417	1285	ID:5	P	0.18.116.5.0.5.5.5	4	24780	303701	10452	4518	0	2200	1425	64065	0	416	0	416
30	11306888	1285	ID:2	P	0.18.116.2.0.2.2.2	4	17828	310652	5726	2610	0	1976	2076	63411	161	593	0	416
31	12588011	1541	ID:6	P	0.18.116.6.0.6.6.6	5	21486	372532	7091	3990	0	2543	1757	61559	0	888	0	593
32	12819256	1541	ID:3	P	0.18.116.3.0.3.3.3	5	21848	372149	7942	5306	0	2806	1753	63759	0	789	0	567
33	12910930	1541	ID:1	P	0.18.116.1.0.1.1.1	5	26285	367714	8402	7027	0	3142	10911	54602	5422	1857	0	325
34	13168185	1541	ID:4	P	0.18.116.4.0.4.4.4	5	26062	367921	10016	6692	0	2780	6404	59103	2474	2354	0	554
35	13184502	1541	ID:5	P	0.18.116.5.0.5.5.5	5	26537	367458	10452	5169	0	2590	1754	63757	0	651	0	390
36	13306333	1541	ID:2	P	0.18.116.2.0.2.2.2	5	19600	374395	5726	3329	0	2379	1769	63743	0	719	0	403
37	14589286	1797	ID:6	P	0.18.116.6.0.6.6.6	6	27550	431978	10073	4458	0	2933	6061	59446	2982	468	0	390
38	14819832	1797	ID:3	P	0.18.116.3.0.3.3.3	6	24234	435261	8052	6066	0	3209	2383	63112	110	760	0	403

Figure 3.11 Benign "pwrtrace.csv"—1 to 38 records.



B	D	E	F	G	H	I	K	L	M	N	O	P	R	S	T	U	V	W
No							Total measurements from the begining of the simulation						Measurements every 2 seconds (monitoring period)					
	Real time	clock_time	ID	P	rimeaddr	seqno	all_cpu	all_lpm	all_transmit	all_listen	all_idle_transmit	all_idle_listen	cpu	lpm	transmit	listen	idle_transmit	idle_listen
	[us]	(in ticks)					(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)
10757	3587190301	459013	ID:5	P	0.18.116.5.0.5.5.5	1792	4227046	1.1E+08	696153	1413275	0	958376	1605	63887	0	763	0	763
10758	3587313763	459013	ID:2	P	0.18.116.2.0.2.2.2	1792	4226306	1.1E+08	696849	1221793	0	754382	6257	59244	2508	2059	0	364
10759	3588594047	459269	ID:6	P	0.18.116.6.0.6.6.6	1793	4274768	1.1E+08	722143	1356760	0	875849	1587	63923	0	416	0	416
10760	3588825278	459269	ID:3	P	0.18.116.3.0.3.3.3	1793	4117288	1.1E+08	624064	1372127	0	948031	1587	63923	0	416	0	416
10761	3588916319	459269	ID:1	P	0.18.116.1.0.1.1.1	1793	8613082	1.1E+08	2425789	2442763	0	720613	2961	62549	131	758	0	403
10762	3589172501	459269	ID:4	P	0.18.116.4.0.4.4.4	1793	4237391	1.1E+08	696699	1285112	0	825907	1586	63924	0	416	0	416
10763	3589191656	459269	ID:5	P	0.18.116.5.0.5.5.5	1793	4233517	1.1E+08	698773	1415424	0	958956	6468	59033	2620	2149	0	580
10764	3589312310	459269	ID:2	P	0.18.116.2.0.2.2.2	1793	4227941	1.1E+08	696849	1222209	0	754798	1632	63878	0	416	0	416
10765	3590594057	459525	ID:6	P	0.18.116.6.0.6.6.6	1794	4276346	1.1E+08	722143	1357176	0	876265	1575	63933	0	416	0	416
10766	3590825297	459525	ID:3	P	0.18.116.3.0.3.3.3	1794	4118865	1.1E+08	624064	1372543	0	948447	1574	63933	0	416	0	416
10767	3590915623	459525	ID:1	P	0.18.116.1.0.1.1.1	1794	8615070	1.1E+08	2425789	2443179	0	721029	1985	63523	0	416	0	416
10768	3591172527	459525	ID:4	P	0.18.116.4.0.4.4.4	1794	4238968	1.1E+08	696699	1285528	0	826323	1574	63934	0	416	0	416
10769	3591190298	459525	ID:5	P	0.18.116.5.0.5.5.5	1794	4235140	1.1E+08	698773	1415840	0	959372	1620	63887	0	416	0	416
10770	3591312382	459525	ID:2	P	0.18.116.2.0.2.2.2	1794	4229517	1.1E+08	696849	1222625	0	755214	1573	63935	0	416	0	416
10771	3592594079	459781	ID:6	P	0.18.116.6.0.6.6.6	1795	4277971	1.1E+08	722143	1357592	0	876681	1622	63870	0	416	0	416
10772	3592825311	459781	ID:3	P	0.18.116.3.0.3.3.3	1795	4120490	1.1E+08	624064	1372959	0	948863	1622	63870	0	416	0	416
10773	3592915644	459781	ID:1	P	0.18.116.1.0.1.1.1	1795	8617027	1.1E+08	2425789	2443595	0	721445	1954	63555	0	416	0	416
10774	3593172522	459781	ID:4	P	0.18.116.4.0.4.4.4	1795	4240593	1.1E+08	696699	1285944	0	826739	1622	63871	0	416	0	416
10775	3593190317	459781	ID:5	P	0.18.116.5.0.5.5.5	1795	4236766	1.1E+08	698773	1416256	0	959788	1623	63870	0	416	0	416
10776	3593312391	459781	ID:2	P	0.18.116.2.0.2.2.2	1795	4231142	1.1E+08	696849	1223041	0	755630	1622	63870	0	416	0	416
10777	3594594061	460037	ID:6	P	0.18.116.6.0.6.6.6	1796	4279589	1.1E+08	722143	1358008	0	877097	1615	63876	0	416	0	416
10778	3594825310	460037	ID:3	P	0.18.116.3.0.3.3.3	1796	4122130	1.1E+08	624064	1373552	0	949456	1637	63856	0	593	0	593
10779	3594934655	460039	ID:1	P	0.18.116.1.0.1.1.1	1796	8623860	1.1E+08	2428553	2445570	0	721809	6830	59263	2764	1975	0	364
10780	3595172515	460037	ID:4	P	0.18.116.4.0.4.4.4	1796	4242210	1.1E+08	696699	1286537	0	827332	1614	63877	0	593	0	593
10781	3595190283	460037	ID:5	P	0.18.116.5.0.5.5.5	1796	4238397	1.1E+08	698773	1416966	0	960498	1628	63865	0	710	0	710
10782	3595313074	460037	ID:2	P	0.18.116.2.0.2.2.2	1796	4234321	1.1E+08	697363	1223992	0	756020	3176	62322	514	951	0	390
10783	3596594058	460293	ID:6	P	0.18.116.6.0.6.6.6	1797	4281196	1.1E+08	722143	1358614	0	877703	1604	63887	0	606	0	606
10784	3596825303	460293	ID:3	P	0.18.116.3.0.3.3.3	1797	4123737	1.1E+08	624064	1373968	0	949872	1604	63887	0	416	0	416
10785	3596915641	460293	ID:1	P	0.18.116.1.0.1.1.1	1797	8625902	1.1E+08	2428553	2445986	0	722225	2039	62889	0	416	0	416
10786	3597172526	460293	ID:4	P	0.18.116.4.0.4.4.4	1797	4243816	1.1E+08	696699	1286953	0	827748	1603	63888	0	416	0	416
10787	3597190263	460293	ID:5	P	0.18.116.5.0.5.5.5	1797	4240004	1.1E+08	698773	1417382	0	960914	1604	63887	0	416	0	416
10788	3597312372	460293	ID:2	P	0.18.116.2.0.2.2.2	1797	4235922	1.1E+08	697363	1224408	0	756436	1598	63910	0	416	0	416
10789	3598594083	460549	ID:6	P	0.18.116.6.0.6.6.6	1798	4282793	1.1E+08	722143	1359030	0	878119	1594	63897	0	416	0	416
10790	3598826646	460549	ID:3	P	0.18.116.3.0.3.3.3	1798	4128484	1.1E+08	625601	1375503	0	950262	4744	60758	1537	1535	0	390
10791	3598916331	460549	ID:1	P	0.18.116.1.0.1.1.1	1798	8629132	1.1E+08	2428874	2446790	0	722615	3227	62283	321	804	0	390
10792	3599172530	460549	ID:4	P	0.18.116.4.0.4.4.4	1798	4245412	1.1E+08	696699	1287369	0	828164	1593	63897	0	416	0	416
10793	3599191309	460549	ID:5	P	0.18.116.5.0.5.5.5	1798	4243080	1.1E+08	699229	1418517	0	961494	3073	62419	456	1135	0	580
10794	3599312385	460549	ID:2	P	0.18.116.2.0.2.2.2	1798	4237491	1.1E+08	697363	1224824	0	756852	1566	63942	0	416	0	416

Figure 3.12 Benign "pwrtrace.csv"—10,757 to 10,794 records.

3.4 Benign Network Traffic Dataset

3.4.1 Benign Network Traffic Dataset – Generation Process

The generated network traffic dataset constitutes the dataset of the simulated benign IoT network scenario that includes records consisting of IoT network traffic features such as source/destination IPv6 address, packet size, and communication protocol. The Cooja simulator provides the “Radio messages” tool that allowed the collection of data related to the corresponding network traffic features. In Figure 3.13, the “Radio messages” output window is depicted along with the three configuration options that are provided by the “Radio messages” tool:

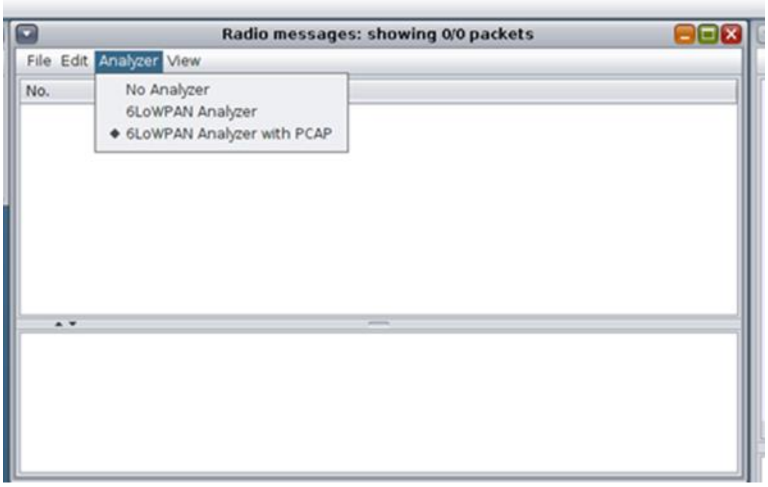
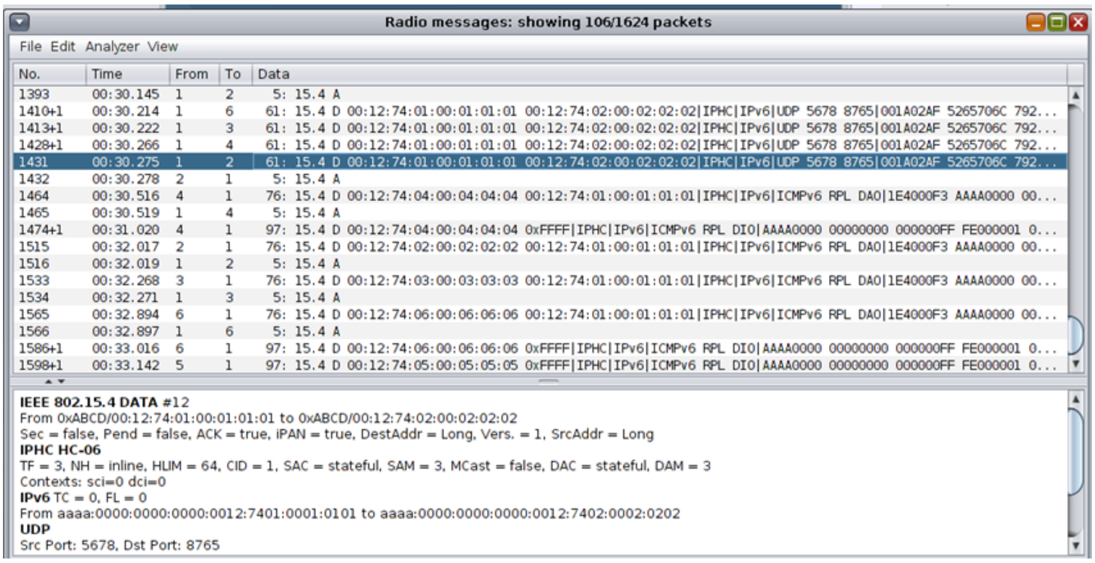

Radio messages: showing 0/0 packets
File
Edit
Analyzer
View
No.
No Analyzer
6LoWPAN Analyzer
◆ 6LoWPAN Analyzer with PCAP

Figure 3.13 “Radio messages” tool—output window.

The “6LoWPAN Analyzer with PCAP” option was selected and the “Radio messages” tool saved the captured network traffic data from the simulated IoT network into a pcap file whose file-naming format was as follows: “radiolog-“+ System.currentTimeMillis()+“.pcap”. During the simulation, the network traffic information about the transmitted data was also being shown in the top part of the “Radio messages” output window as depicted in the top part of Figure 3.14. When the simulation stopped, the generated pcap file was saved as “radiolog-1605811324302.pcap” within the “…/cooja/build” folder.



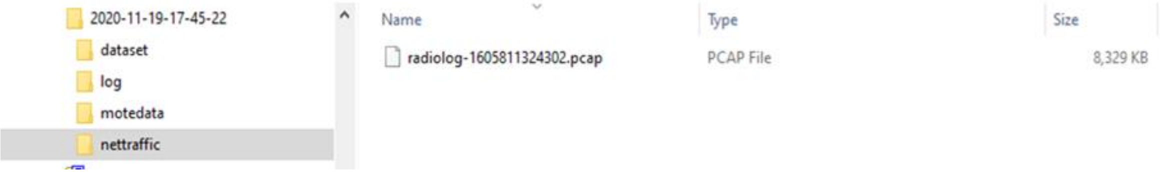
```
No.     Time       From  To  Data
1393    00:30.145  1     2    5: 15.4 A
1410+1  00:30.214  1     6   61: 15.4 D 00:12:74:01:00:01:01:01 00:12:74:02:00:02:02:02|IPHC|IPv6|UDP 5678 8765|001A02AF 5265706C 792...
1413+1  00:30.222  1     3   61: 15.4 D 00:12:74:01:00:01:01:01 00:12:74:02:00:02:02:02|IPHC|IPv6|UDP 5678 8765|001A02AF 5265706C 792...
1428+1  00:30.266  1     4   61: 15.4 D 00:12:74:01:00:01:01:01 00:12:74:02:00:02:02:02|IPHC|IPv6|UDP 5678 8765|001A02AF 5265706C 792...
1431    00:30.275  1     2   61: 15.4 D 00:12:74:01:00:01:01:01 00:12:74:02:00:02:02:02|IPHC|IPv6|UDP 5678 8765|001A02AF 5265706C 792...
1432    00:30.278  2     1    5: 15.4 A
1464    00:30.516  4     1   76: 15.4 D 00:12:74:04:00:04:04:04 00:12:74:01:00:01:01:01|IPHC|IPv6|ICMPv6 RPL DAO|1E4000F3 AAAA0000 00...
1465    00:30.519  1     4    5: 15.4 A
1474+1  00:31.020  4     1   97: 15.4 D 00:12:74:04:00:04:04:04 0xFFFF|IPHC|IPv6|ICMPv6 RPL DIO|AAAA0000 00000000 000000FF FE000001 0...
1515    00:32.017  2     1   76: 15.4 D 00:12:74:02:00:02:02:02 00:12:74:01:00:01:01:01|IPHC|IPv6|ICMPv6 RPL DAO|1E4000F3 AAAA0000 00...
1516    00:32.019  1     2    5: 15.4 A
1533    00:32.268  3     1   76: 15.4 D 00:12:74:03:00:03:03:03 00:12:74:01:00:01:01:01|IPHC|IPv6|ICMPv6 RPL DAO|1E4000F3 AAAA0000 00...
1534    00:32.271  1     3    5: 15.4 A
1565    00:32.894  6     1   76: 15.4 D 00:12:74:06:00:06:06:06 00:12:74:01:00:01:01:01|IPHC|IPv6|ICMPv6 RPL DAO|1E4000F3 AAAA0000 00...
1566    00:32.897  1     6    5: 15.4 A
1586+1  00:33.016  6     1   97: 15.4 D 00:12:74:06:00:06:06:06 0xFFFF|IPHC|IPv6|ICMPv6 RPL DIO|AAAA0000 00000000 000000FF FE000001 0...
1598+1  00:33.142  5     1   97: 15.4 D 00:12:74:05:00:05:05:05 0xFFFF|IPHC|IPv6|ICMPv6 RPL DIO|AAAA0000 00000000 000000FF FE000001 0...
```

```
IEEE 802.15.4 DATA #12
From 0xABCD/00:12:74:01:00:01:01:01 to 0xABCD/00:12:74:02:00:02:02:02
Sec = false, Pend = false, ACK = true, iPAN = true, DestAddr = Long, Vers. = 1, SrcAddr = Long
IPHC HC-06
TF = 3, NH = inline, HLIM = 64, CID = 1, SAC = stateful, SAM = 3, MCast = false, DAC = stateful, DAM = 3
Contexts: sci=0 dci=0
IPv6 TC = 0, FL = 0
From aaaa:0000:0000:0000:0012:7401:0001:0101 to aaaa:0000:0000:0000:0012:7402:0002:0202
UDP
Src Port: 5678, Dst Port: 8765
```

Figure 3.14 Network traffic information from the benign scenario in the “Radio messages” output window.

Having now saved all the captured raw network traffic information, through the “Radio messages” tool, into a pcap file, the challenging task was to extract this information from the pcap file to a csv file that would be the network traffic dataset of the simulated benign IoT network scenario. This challenge was addressed by utilising the “IoT_Simul.sh” file that was also used in the “powertrace” dataset generation process, as described in Section 3.3.1, and the well-known network protocol analyser Wireshark [106].

In particular, the first step was the use of the “IoT_Simul.sh” file in order to copy the “radiolog-1605811324302.pcap” file from the “…/cooja/build” folder located in the Cooja Simulator environment to the “nettraffic” folder that was created by the “IoT_Simul.sh” file inside the root folder “2020-11-19-17-45-22” that was also created by the “IoT_Simul.sh” during the “powertrace” dataset generation process. The “nettraffic” folder inside the root folder “2020-11-19-17-45-22” and the copy of the “radiolog-1605811324302.pcap” file in the “nettraffic” folder are shown in Figure .

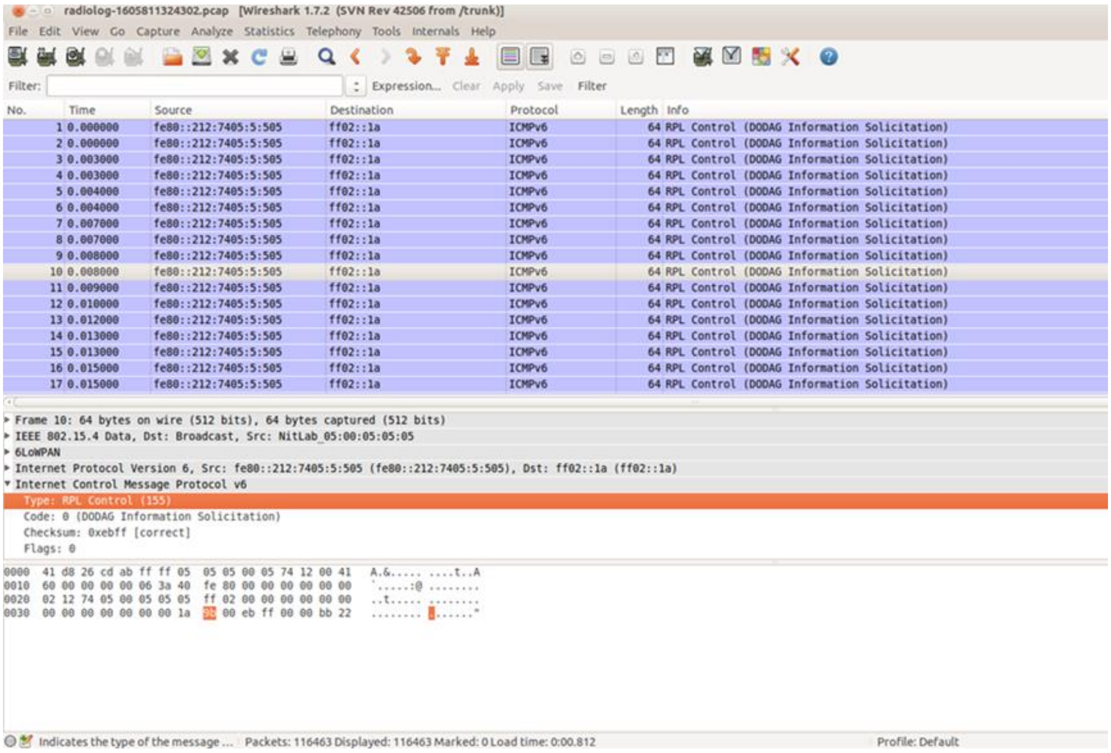


2020-11-19-17-45-22
dataset
log
motedata
nettraffic

Name	Type	Size
radiolog-1605811324302.pcap	PCAP File	8,329 KB

Figure 3.15 The “nettraffic” folder inside the root folder “2020-11-19-17-45-22” and the copy of the “radiolog-1605811324302.pcap” file.

After having the copy of the “radiolog-1605811324302.pcap” file in the “nettraffic” folder, the next step was the extraction of the stored network traffic information from the “radiolog-1605811324302.pcap” file to the “radiolog.csv” file. This was achieved through Wireshark as Wireshark allows opening a pcap file and exporting data to a csv file. In Figure 3.16, the upper panel of the Wireshark window shows the seventeen first packets included in the “radiolog-1605811324302.pcap” file that was opened via Wireshark. The middle panel shows the protocol details of the 10th packet selected in the upper panel and the bottom panel presents the protocol details of the selected 10th packet in both HEX and ASCII format.



radiolog-1605811324302.pcap [Wireshark 1.7.2 (SVN Rev 42506 from /trunk)]

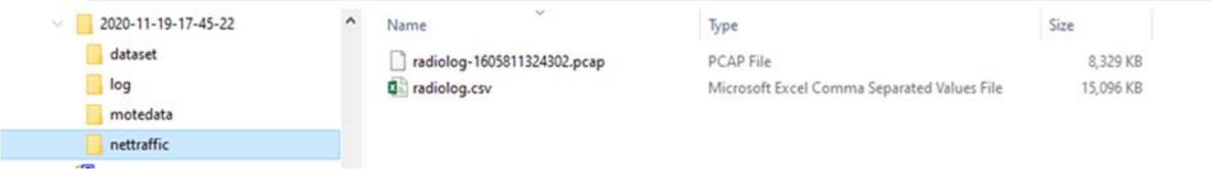
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply Save Filter

```
No.  Time      Source                   Destination  Protocol  Length Info
   1 0.000000  fe80::212:7405:5:505     ff02::1a     ICMPv6        64 RPL Control (DODAG Information Solicitation)
   2 0.000000  fe80::212:7405:5:505     ff02::1a     ICMPv6        64 RPL Control (DODAG Information Solicitation)
   3 0.003000  fe80::212:7405:5:505     ff02::1a     ICMPv6        64 RPL Control (DODAG Information Solicitation)
   4 0.003000  fe80::212:7405:5:505     ff02::1a     ICMPv6        64 RPL Control (DODAG Information Solicitation)
   5 0.004000  fe80::212:7405:5:505     ff02::1a     ICMPv6        64 RPL Control (DODAG Information Solicitation)
   6 0.004000  fe80::212:7405:5:505     ff02::1a     ICMPv6        64 RPL Control (DODAG Information Solicitation)
   7 0.007000  fe80::212:7405:5:505     ff02::1a     ICMPv6        64 RPL Control (DODAG Information Solicitation)
   8 0.007000  fe80::212:7405:5:505     ff02::1a     ICMPv6        64 RPL Control (DODAG Information Solicitation)
   9 0.008000  fe80::212:7405:5:505     ff02::1a     ICMPv6        64 RPL Control (DODAG Information Solicitation)
  10 0.008000  fe80::212:7405:5:505     ff02::1a     ICMPv6        64 RPL Control (DODAG Information Solicitation)
  11 0.009000  fe80::212:7405:5:505     ff02::1a     ICMPv6        64 RPL Control (DODAG Information Solicitation)
  12 0.010000  fe80::212:7405:5:505     ff02::1a     ICMPv6        64 RPL Control (DODAG Information Solicitation)
  13 0.012000  fe80::212:7405:5:505     ff02::1a     ICMPv6        64 RPL Control (DODAG Information Solicitation)
  14 0.013000  fe80::212:7405:5:505     ff02::1a     ICMPv6        64 RPL Control (DODAG Information Solicitation)
  15 0.013000  fe80::212:7405:5:505     ff02::1a     ICMPv6        64 RPL Control (DODAG Information Solicitation)
  16 0.015000  fe80::212:7405:5:505     ff02::1a     ICMPv6        64 RPL Control (DODAG Information Solicitation)
  17 0.015000  fe80::212:7405:5:505     ff02::1a     ICMPv6        64 RPL Control (DODAG Information Solicitation)

▸ Frame 10: 64 bytes on wire (512 bits), 64 bytes captured (512 bits)
▸ IEEE 802.15.4 Data, Dst: Broadcast, Src: NitLab_05:00:05:05:05
▸ 6LoWPAN
▸ Internet Protocol Version 6, Src: fe80::212:7405:5:505 (fe80::212:7405:5:505), Dst: ff02::1a (ff02::1a)
▾ Internet Control Message Protocol v6
    Type: RPL Control (155)
    Code: 0 (DODAG Information Solicitation)
    Checksum: 0xebff [correct]
    Flags: 0

0000  41 d8 26 cd ab ff ff 05  05 05 00 05 74 12 00 41   A.&..... ....t..A
0010  60 00 00 00 00 06 3a 40  fe 80 00 00 00 00 00 00   `......:@ ........
0020  02 12 74 05 00 05 05 05  ff 02 00 00 00 00 00 00   ..t..... ........
0030  00 00 00 00 00 00 00 1a  9b 00 eb ff 00 00 bb 22   ........ .......".
```

Indicates the type of the message ... Packets: 116463 Displayed: 116463 Marked: 0 Load time: 0:00.812 Profile: Default

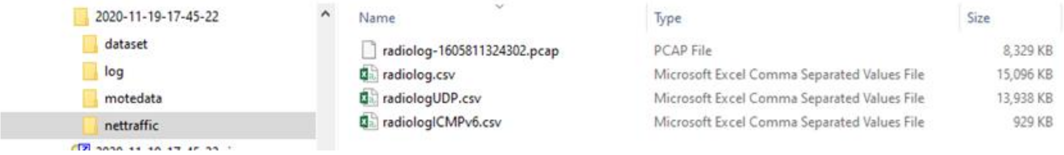
Figure 3.16 The first seventeenth packets in the “radiolog-1605811324302.pcap” file.

The data from the “radiolog-1605811324302.pcap” file were exported and saved, through Wireshark, into the “radiolog.csv” file in the “nettraffic” folder in the project environment, as shown in Figure 3.17. Furthermore, it is worthwhile mentioning that we also used Wireshark to filter the “radiolog-1605811324302.pcap” file based on the ICMPv6 protocol and the UDP protocol and then exported and saved the filtered results, through Wireshark, in the “radiologICMPv6.csv” file and the “radiologUDP.csv” file, respectively, in the “nettraffic” folder in the project environment, as shown in Figure . The “radiologICMPv6.csv” file and the “radiologUDP.csv” file facilitated the analysis of the capture traffic as shown in Chapter 5.



Name	Type	Size
radiolog-1605811324302.pcap	PCAP File	8,329 KB
radiolog.csv	Microsoft Excel Comma Separated Values File	15,096 KB

Figure 3.17 The “radiolog.csv” file in the “nettraffic” folder in the project environment.



Name	Type	Size
radiolog-1605811324302.pcap	PCAP File	8,329 KB
radiolog.csv	Microsoft Excel Comma Separated Values File	15,096 KB
radiologUDP.csv	Microsoft Excel Comma Separated Values File	13,938 KB
radiologICMPv6.csv	Microsoft Excel Comma Separated Values File	929 KB

Figure 3.18 The “radiologICMPv6.csv” file and the “radiologUDP.csv” file in the “nettraffic” folder in the project environment.

Finally, an overview of the above-described process followed to extract the required information from the “radiolog-1605811324302.pcap” file to the “radiolog.csv”, “radiologICMPv6.csv” and “radiologUDP.csv” files is depicted in Figure 3.20.

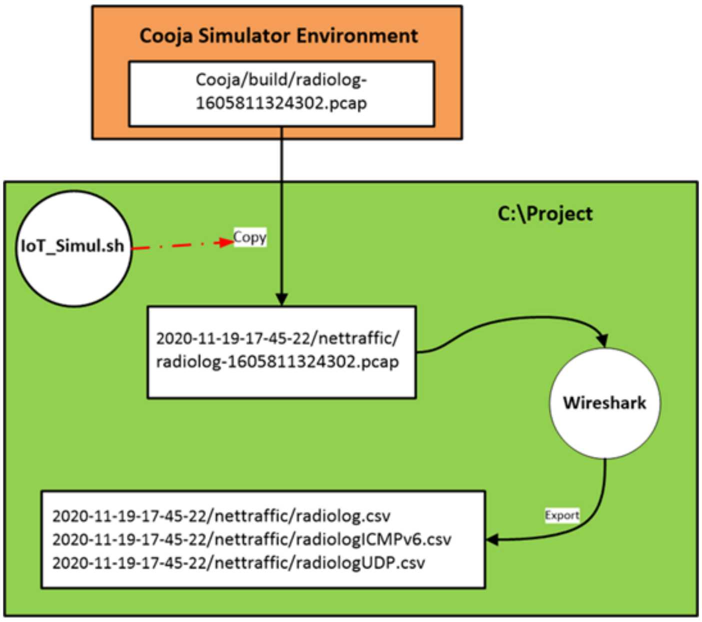

Cooja Simulator Environment
Cooja/build/radiolog-1605811324302.pcap
C:\Project
IoT_Simul.sh
Copy
2020-11-19-17-45-22/nettraffic/radiolog-1605811324302.pcap
Wireshark
Export
2020-11-19-17-45-22/nettraffic/radiolog.csv
2020-11-19-17-45-22/nettraffic/radiologICMPv6.csv
2020-11-19-17-45-22/nettraffic/radiologUDP.csv

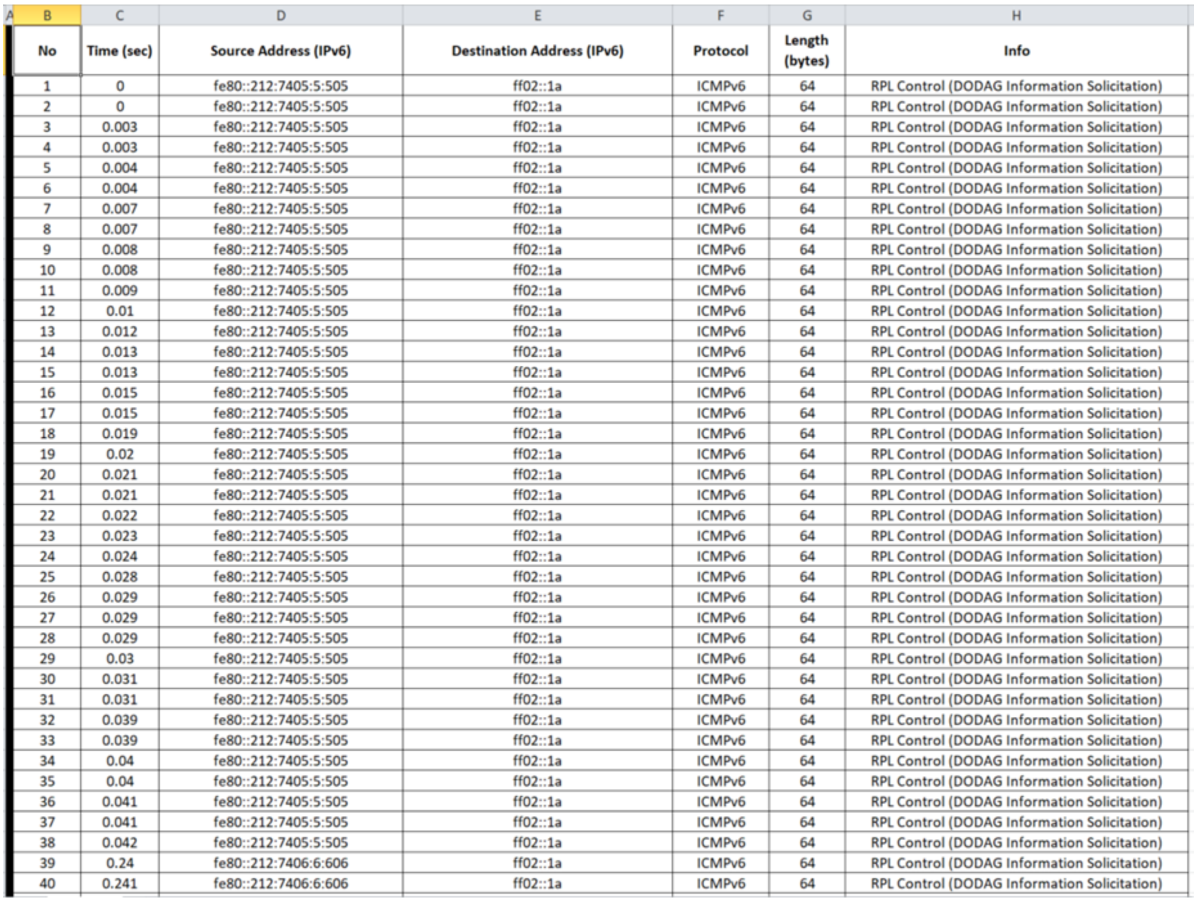
Figure 3.20 An overview of the process followed to extract all the required network traffic information from the "radiolog-1605811324302.pcap" file.

3.4.2 Benign Network Traffic Dataset – Generated Results

The network traffic dataset consists of the following csv files which are located in the "nettraffic" folder in the project environment as described in Section 3.4.1: "radiolog.csv", "radiologICMPv6.csv", and "radiologUDP.csv" files. In this Section, we present sets of records from these files.

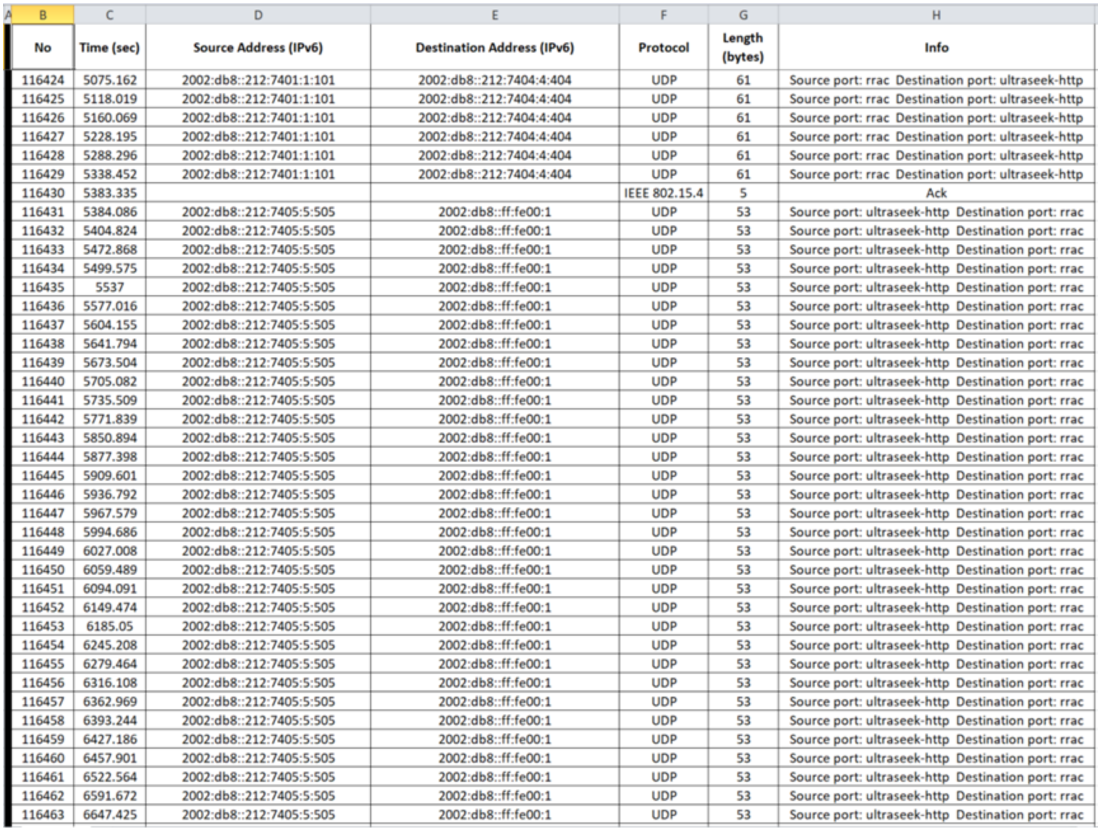
3.4.2.1 Benign "radiolog.csv"

The generated benign "radiolog.csv" file consists of 116,463 records and its first 40 records (i.e., 1–40) and its last 40 records (116,424-116,463) are depicted in Figure 3.21 and Figure 3.22, respectively.



No	Time (sec)	Source Address (IPv6)	Destination Address (IPv6)	Protocol	Length (bytes)	Info
1	0	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
2	0	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
3	0.003	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
4	0.003	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
5	0.004	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
6	0.004	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
7	0.007	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
8	0.007	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
9	0.008	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
10	0.008	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
11	0.009	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
12	0.01	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
13	0.012	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
14	0.013	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
15	0.013	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
16	0.015	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
17	0.015	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
18	0.019	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
19	0.02	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
20	0.021	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
21	0.021	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
22	0.022	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
23	0.023	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
24	0.024	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
25	0.028	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
26	0.029	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
27	0.029	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
28	0.029	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
29	0.03	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
30	0.031	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
31	0.031	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
32	0.039	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
33	0.039	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
34	0.04	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
35	0.04	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
36	0.041	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
37	0.041	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
38	0.042	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
39	0.24	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
40	0.241	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)

Figure 3.21 Benign "radiolog.csv"—1 to 40 records.

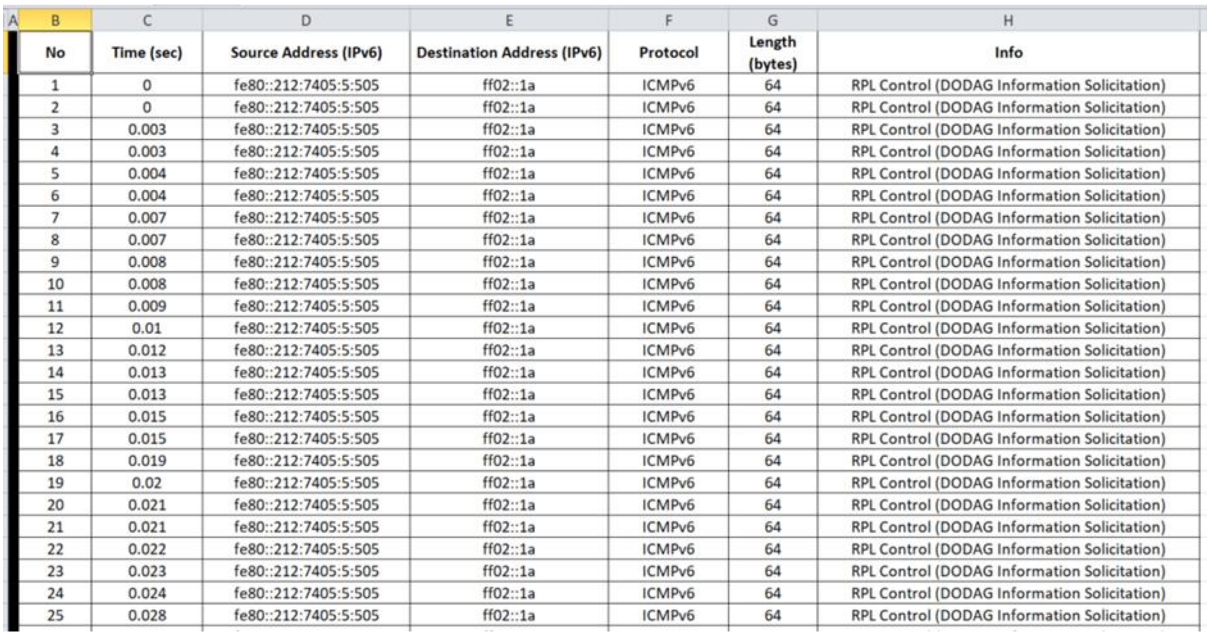


No	Time (sec)	Source Address (IPv6)	Destination Address (IPv6)	Protocol	Length (bytes)	Info
116424	5075.162	2002:db8::212:7401:1:101	2002:db8::212:7404:4:404	UDP	61	Source port: rrac Destination port: ultraseek-http
116425	5118.019	2002:db8::212:7401:1:101	2002:db8::212:7404:4:404	UDP	61	Source port: rrac Destination port: ultraseek-http
116426	5160.069	2002:db8::212:7401:1:101	2002:db8::212:7404:4:404	UDP	61	Source port: rrac Destination port: ultraseek-http
116427	5228.195	2002:db8::212:7401:1:101	2002:db8::212:7404:4:404	UDP	61	Source port: rrac Destination port: ultraseek-http
116428	5288.296	2002:db8::212:7401:1:101	2002:db8::212:7404:4:404	UDP	61	Source port: rrac Destination port: ultraseek-http
116429	5338.452	2002:db8::212:7401:1:101	2002:db8::212:7404:4:404	UDP	61	Source port: rrac Destination port: ultraseek-http
116430	5383.335			IEEE 802.15.4	5	Ack
116431	5384.086	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	53	Source port: ultraseek-http Destination port: rrac
116432	5404.824	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	53	Source port: ultraseek-http Destination port: rrac
116433	5472.868	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	53	Source port: ultraseek-http Destination port: rrac
116434	5499.575	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	53	Source port: ultraseek-http Destination port: rrac
116435	5537	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	53	Source port: ultraseek-http Destination port: rrac
116436	5577.016	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	53	Source port: ultraseek-http Destination port: rrac
116437	5604.155	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	53	Source port: ultraseek-http Destination port: rrac
116438	5641.794	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	53	Source port: ultraseek-http Destination port: rrac
116439	5673.504	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	53	Source port: ultraseek-http Destination port: rrac
116440	5705.082	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	53	Source port: ultraseek-http Destination port: rrac
116441	5735.509	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	53	Source port: ultraseek-http Destination port: rrac
116442	5771.839	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	53	Source port: ultraseek-http Destination port: rrac
116443	5850.894	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	53	Source port: ultraseek-http Destination port: rrac
116444	5877.398	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	53	Source port: ultraseek-http Destination port: rrac
116445	5909.601	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	53	Source port: ultraseek-http Destination port: rrac
116446	5936.792	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	53	Source port: ultraseek-http Destination port: rrac
116447	5967.579	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	53	Source port: ultraseek-http Destination port: rrac
116448	5994.686	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	53	Source port: ultraseek-http Destination port: rrac
116449	6027.008	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	53	Source port: ultraseek-http Destination port: rrac
116450	6059.489	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	53	Source port: ultraseek-http Destination port: rrac
116451	6094.091	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	53	Source port: ultraseek-http Destination port: rrac
116452	6149.474	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	53	Source port: ultraseek-http Destination port: rrac
116453	6185.05	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	53	Source port: ultraseek-http Destination port: rrac
116454	6245.208	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	53	Source port: ultraseek-http Destination port: rrac
116455	6279.464	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	53	Source port: ultraseek-http Destination port: rrac
116456	6316.108	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	53	Source port: ultraseek-http Destination port: rrac
116457	6362.969	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	53	Source port: ultraseek-http Destination port: rrac
116458	6393.244	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	53	Source port: ultraseek-http Destination port: rrac
116459	6427.186	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	53	Source port: ultraseek-http Destination port: rrac
116460	6457.901	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	53	Source port: ultraseek-http Destination port: rrac
116461	6522.564	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	53	Source port: ultraseek-http Destination port: rrac
116462	6591.672	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	53	Source port: ultraseek-http Destination port: rrac
116463	6647.425	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	53	Source port: ultraseek-http Destination port: rrac

Figure 3.22 Benign "radiolog.csv" – 116,424-116,463 records.

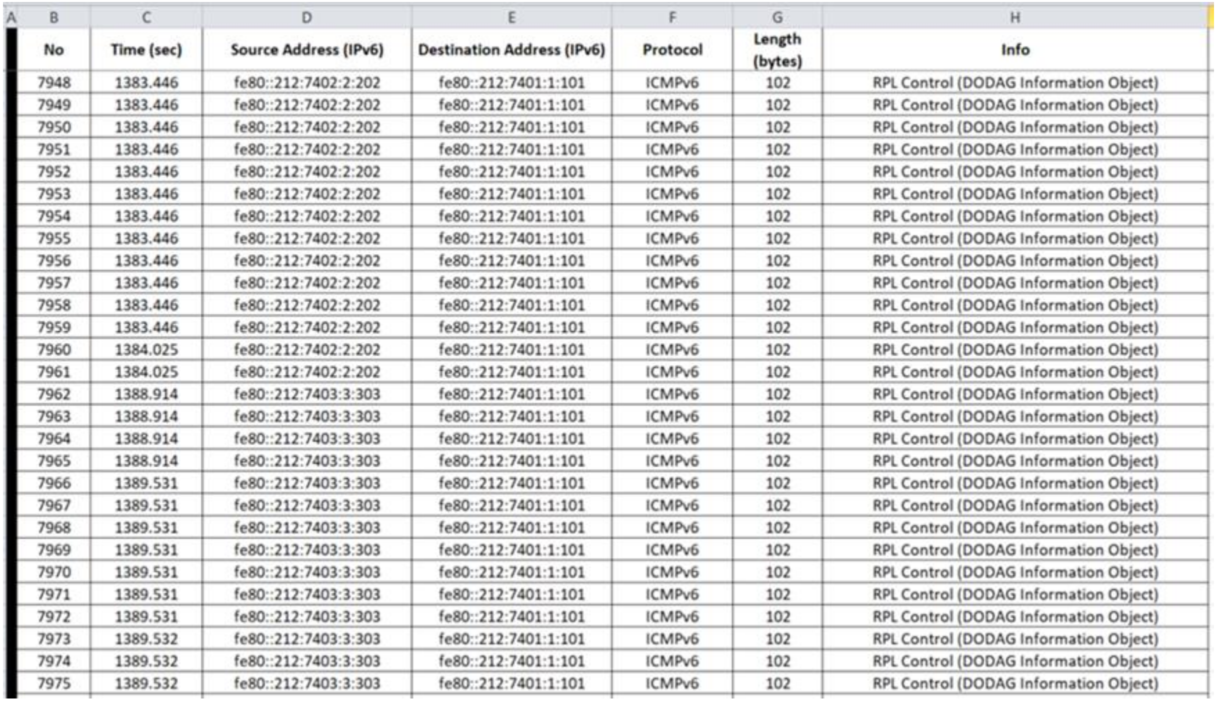
3.4.2.2 Benign “radiologICMPv6.csv”

The generated benign “radiologICMPv6.csv” file consists of 7,975 records and its first 25 records (i.e., 1–25) and its last 27 records (i.e., 7,948–7,975) are depicted in Figure 3.23 and Figure 3.24, respectively.



B	C	D	E	F	G	H
No	Time (sec)	Source Address (IPv6)	Destination Address (IPv6)	Protocol	Length (bytes)	Info
1	0	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
2	0	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
3	0.003	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
4	0.003	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
5	0.004	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
6	0.004	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
7	0.007	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
8	0.007	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
9	0.008	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
10	0.008	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
11	0.009	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
12	0.01	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
13	0.012	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
14	0.013	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
15	0.013	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
16	0.015	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
17	0.015	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
18	0.019	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
19	0.02	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
20	0.021	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
21	0.021	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
22	0.022	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
23	0.023	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
24	0.024	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
25	0.028	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)

Figure 3.23 Benign “radiologICMPv6.csv”—1 to 25 records.

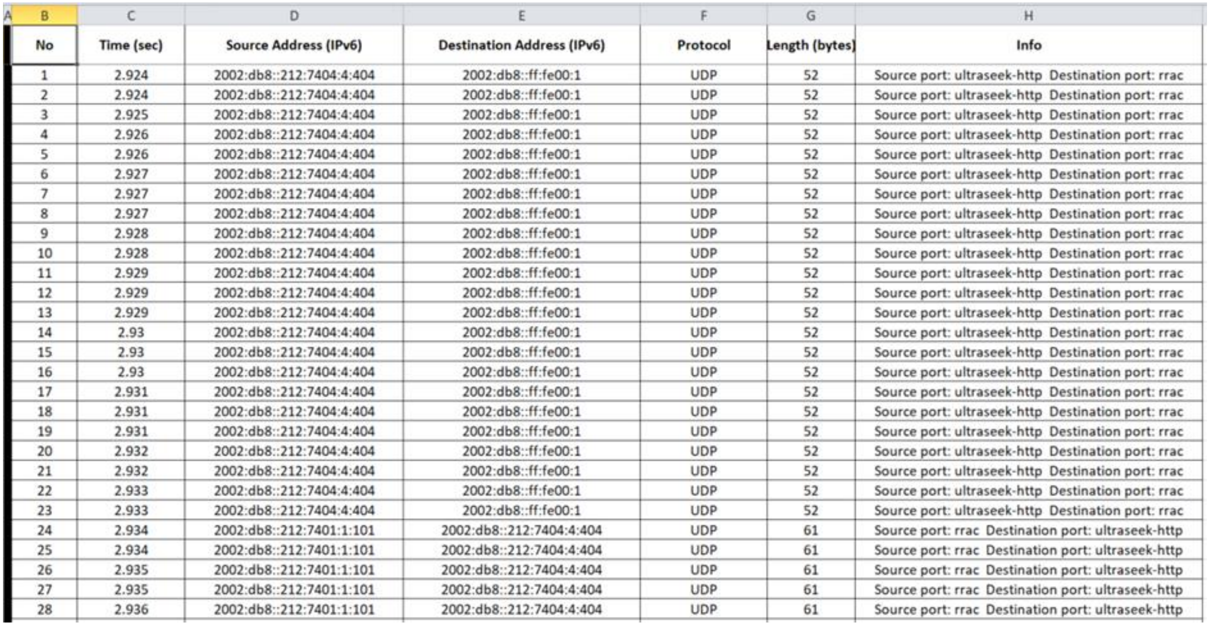


B	C	D	E	F	G	H
No	Time (sec)	Source Address (IPv6)	Destination Address (IPv6)	Protocol	Length (bytes)	Info
7948	1383.446	fe80::212:7402:2:202	fe80::212:7401:1:101	ICMPv6	102	RPL Control (DODAG Information Object)
7949	1383.446	fe80::212:7402:2:202	fe80::212:7401:1:101	ICMPv6	102	RPL Control (DODAG Information Object)
7950	1383.446	fe80::212:7402:2:202	fe80::212:7401:1:101	ICMPv6	102	RPL Control (DODAG Information Object)
7951	1383.446	fe80::212:7402:2:202	fe80::212:7401:1:101	ICMPv6	102	RPL Control (DODAG Information Object)
7952	1383.446	fe80::212:7402:2:202	fe80::212:7401:1:101	ICMPv6	102	RPL Control (DODAG Information Object)
7953	1383.446	fe80::212:7402:2:202	fe80::212:7401:1:101	ICMPv6	102	RPL Control (DODAG Information Object)
7954	1383.446	fe80::212:7402:2:202	fe80::212:7401:1:101	ICMPv6	102	RPL Control (DODAG Information Object)
7955	1383.446	fe80::212:7402:2:202	fe80::212:7401:1:101	ICMPv6	102	RPL Control (DODAG Information Object)
7956	1383.446	fe80::212:7402:2:202	fe80::212:7401:1:101	ICMPv6	102	RPL Control (DODAG Information Object)
7957	1383.446	fe80::212:7402:2:202	fe80::212:7401:1:101	ICMPv6	102	RPL Control (DODAG Information Object)
7958	1383.446	fe80::212:7402:2:202	fe80::212:7401:1:101	ICMPv6	102	RPL Control (DODAG Information Object)
7959	1383.446	fe80::212:7402:2:202	fe80::212:7401:1:101	ICMPv6	102	RPL Control (DODAG Information Object)
7960	1384.025	fe80::212:7402:2:202	fe80::212:7401:1:101	ICMPv6	102	RPL Control (DODAG Information Object)
7961	1384.025	fe80::212:7402:2:202	fe80::212:7401:1:101	ICMPv6	102	RPL Control (DODAG Information Object)
7962	1388.914	fe80::212:7403:3:303	fe80::212:7401:1:101	ICMPv6	102	RPL Control (DODAG Information Object)
7963	1388.914	fe80::212:7403:3:303	fe80::212:7401:1:101	ICMPv6	102	RPL Control (DODAG Information Object)
7964	1388.914	fe80::212:7403:3:303	fe80::212:7401:1:101	ICMPv6	102	RPL Control (DODAG Information Object)
7965	1388.914	fe80::212:7403:3:303	fe80::212:7401:1:101	ICMPv6	102	RPL Control (DODAG Information Object)
7966	1389.531	fe80::212:7403:3:303	fe80::212:7401:1:101	ICMPv6	102	RPL Control (DODAG Information Object)
7967	1389.531	fe80::212:7403:3:303	fe80::212:7401:1:101	ICMPv6	102	RPL Control (DODAG Information Object)
7968	1389.531	fe80::212:7403:3:303	fe80::212:7401:1:101	ICMPv6	102	RPL Control (DODAG Information Object)
7969	1389.531	fe80::212:7403:3:303	fe80::212:7401:1:101	ICMPv6	102	RPL Control (DODAG Information Object)
7970	1389.531	fe80::212:7403:3:303	fe80::212:7401:1:101	ICMPv6	102	RPL Control (DODAG Information Object)
7971	1389.531	fe80::212:7403:3:303	fe80::212:7401:1:101	ICMPv6	102	RPL Control (DODAG Information Object)
7972	1389.531	fe80::212:7403:3:303	fe80::212:7401:1:101	ICMPv6	102	RPL Control (DODAG Information Object)
7973	1389.532	fe80::212:7403:3:303	fe80::212:7401:1:101	ICMPv6	102	RPL Control (DODAG Information Object)
7974	1389.532	fe80::212:7403:3:303	fe80::212:7401:1:101	ICMPv6	102	RPL Control (DODAG Information Object)
7975	1389.532	fe80::212:7403:3:303	fe80::212:7401:1:101	ICMPv6	102	RPL Control (DODAG Information Object)

Figure 3.24 Benign “radiologICMPv6.csv”—7,948 to 7,975 records.

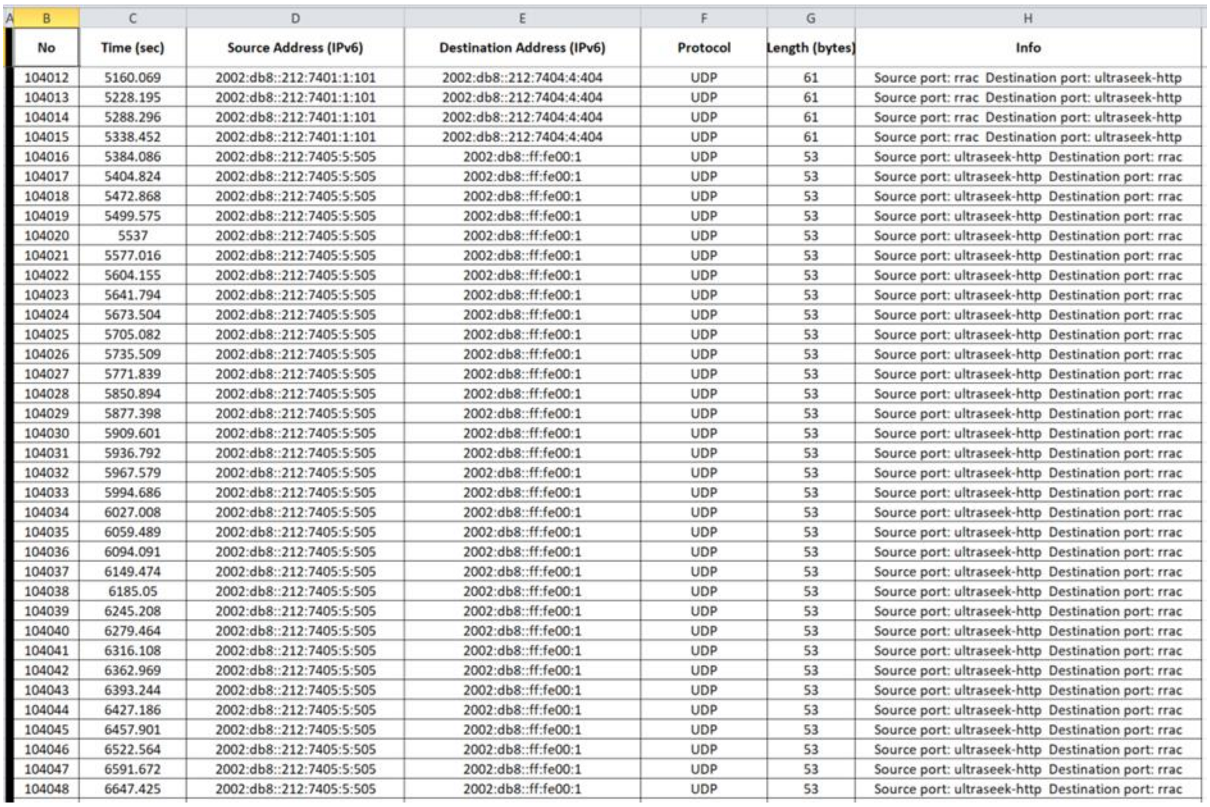
3.4.2.3 Benign “radiologUDP.csv”

The generated benign “radiologUDP.csv” file consists of 104,048 records and its first 28 records (i.e., 1–28) and its last 37 records (i.e., 104,012–104,048) are depicted in Figure 3.25 and Figure 3.26, respectively.



B	C	D	E	F	G	H
No	Time (sec)	Source Address (IPv6)	Destination Address (IPv6)	Protocol	Length (bytes)	Info
1	2.924	2002:db8::212:7404:4:404	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
2	2.924	2002:db8::212:7404:4:404	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
3	2.925	2002:db8::212:7404:4:404	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
4	2.926	2002:db8::212:7404:4:404	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
5	2.926	2002:db8::212:7404:4:404	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
6	2.927	2002:db8::212:7404:4:404	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
7	2.927	2002:db8::212:7404:4:404	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
8	2.927	2002:db8::212:7404:4:404	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
9	2.928	2002:db8::212:7404:4:404	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
10	2.928	2002:db8::212:7404:4:404	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
11	2.929	2002:db8::212:7404:4:404	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
12	2.929	2002:db8::212:7404:4:404	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
13	2.929	2002:db8::212:7404:4:404	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
14	2.93	2002:db8::212:7404:4:404	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
15	2.93	2002:db8::212:7404:4:404	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
16	2.93	2002:db8::212:7404:4:404	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
17	2.931	2002:db8::212:7404:4:404	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
18	2.931	2002:db8::212:7404:4:404	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
19	2.931	2002:db8::212:7404:4:404	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
20	2.932	2002:db8::212:7404:4:404	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
21	2.932	2002:db8::212:7404:4:404	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
22	2.933	2002:db8::212:7404:4:404	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
23	2.933	2002:db8::212:7404:4:404	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
24	2.934	2002:db8::212:7401:1:101	2002:db8::212:7404:4:404	UDP	61	Source port: rrac Destination port: ultraseek-http
25	2.934	2002:db8::212:7401:1:101	2002:db8::212:7404:4:404	UDP	61	Source port: rrac Destination port: ultraseek-http
26	2.935	2002:db8::212:7401:1:101	2002:db8::212:7404:4:404	UDP	61	Source port: rrac Destination port: ultraseek-http
27	2.935	2002:db8::212:7401:1:101	2002:db8::212:7404:4:404	UDP	61	Source port: rrac Destination port: ultraseek-http
28	2.936	2002:db8::212:7401:1:101	2002:db8::212:7404:4:404	UDP	61	Source port: rrac Destination port: ultraseek-http

Figure 3.25 Benign “radiologUDP.csv”—1 to 28 records.



B	C	D	E	F	G	H
No	Time (sec)	Source Address (IPv6)	Destination Address (IPv6)	Protocol	Length (bytes)	Info
104012	5160.069	2002:db8::212:7401:1:101	2002:db8::212:7404:4:404	UDP	61	Source port: rrac Destination port: ultraseek-http
104013	5228.195	2002:db8::212:7401:1:101	2002:db8::212:7404:4:404	UDP	61	Source port: rrac Destination port: ultraseek-http
104014	5288.296	2002:db8::212:7401:1:101	2002:db8::212:7404:4:404	UDP	61	Source port: rrac Destination port: ultraseek-http
104015	5338.452	2002:db8::212:7401:1:101	2002:db8::212:7404:4:404	UDP	61	Source port: rrac Destination port: ultraseek-http
104016	5384.086	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	53	Source port: ultraseek-http Destination port: rrac
104017	5404.824	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	53	Source port: ultraseek-http Destination port: rrac
104018	5472.868	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	53	Source port: ultraseek-http Destination port: rrac
104019	5499.575	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	53	Source port: ultraseek-http Destination port: rrac
104020	5537	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	53	Source port: ultraseek-http Destination port: rrac
104021	5577.016	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	53	Source port: ultraseek-http Destination port: rrac
104022	5604.155	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	53	Source port: ultraseek-http Destination port: rrac
104023	5641.794	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	53	Source port: ultraseek-http Destination port: rrac
104024	5673.504	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	53	Source port: ultraseek-http Destination port: rrac
104025	5705.082	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	53	Source port: ultraseek-http Destination port: rrac
104026	5735.509	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	53	Source port: ultraseek-http Destination port: rrac
104027	5771.839	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	53	Source port: ultraseek-http Destination port: rrac
104028	5850.894	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	53	Source port: ultraseek-http Destination port: rrac
104029	5877.398	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	53	Source port: ultraseek-http Destination port: rrac
104030	5909.601	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	53	Source port: ultraseek-http Destination port: rrac
104031	5936.792	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	53	Source port: ultraseek-http Destination port: rrac
104032	5967.579	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	53	Source port: ultraseek-http Destination port: rrac
104033	5994.686	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	53	Source port: ultraseek-http Destination port: rrac
104034	6027.008	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	53	Source port: ultraseek-http Destination port: rrac
104035	6059.489	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	53	Source port: ultraseek-http Destination port: rrac
104036	6094.091	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	53	Source port: ultraseek-http Destination port: rrac
104037	6149.474	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	53	Source port: ultraseek-http Destination port: rrac
104038	6185.05	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	53	Source port: ultraseek-http Destination port: rrac
104039	6245.208	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	53	Source port: ultraseek-http Destination port: rrac
104040	6279.464	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	53	Source port: ultraseek-http Destination port: rrac
104041	6316.108	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	53	Source port: ultraseek-http Destination port: rrac
104042	6362.969	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	53	Source port: ultraseek-http Destination port: rrac
104043	6393.244	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	53	Source port: ultraseek-http Destination port: rrac
104044	6427.186	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	53	Source port: ultraseek-http Destination port: rrac
104045	6457.901	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	53	Source port: ultraseek-http Destination port: rrac
104046	6522.564	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	53	Source port: ultraseek-http Destination port: rrac
104047	6591.672	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	53	Source port: ultraseek-http Destination port: rrac
104048	6647.425	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	53	Source port: ultraseek-http Destination port: rrac

Figure 3.26 Benign “radiologUDP.csv”—104,012 to 104,048 records.

3.5 Summary

In this Chapter, a detailed description of the approach proposed to generate a set of benign IoT datasets from a benign IoT network scenario implemented in the Cooja simulator was provided. The IoT-specific information from the simulated scenario was captured from the Contiki plugin “powertrace” and the Cooja tool “Radio messages” in order to generate the “powertrace” dataset and the network traffic dataset within csv files, respectively, which constitute the benign IoT datasets for the simulated benign IoT network scenario. In particular, the “powertrace” dataset consists of the following csv files: the “pwrtrace.csv” file and one csv file for each mote (i.e.,“mote1.csv”, “mote2.csv”, “mote3.csv”, “mote4.csv”, “mote5.csv”, and “mote6.csv”) with its corresponding information from the “pwrtrace.csv” file , while the network traffic dataset consists of the following csv files: “radiolog.csv”, “radiologICMPv6.csv”, and “radiologUDP.csv”. The structure of the generated benign IoT datasets from the benign IoT network scenario implemented in the Cooja simulator, as described in this Chapter, is shown in Figure 3.27.

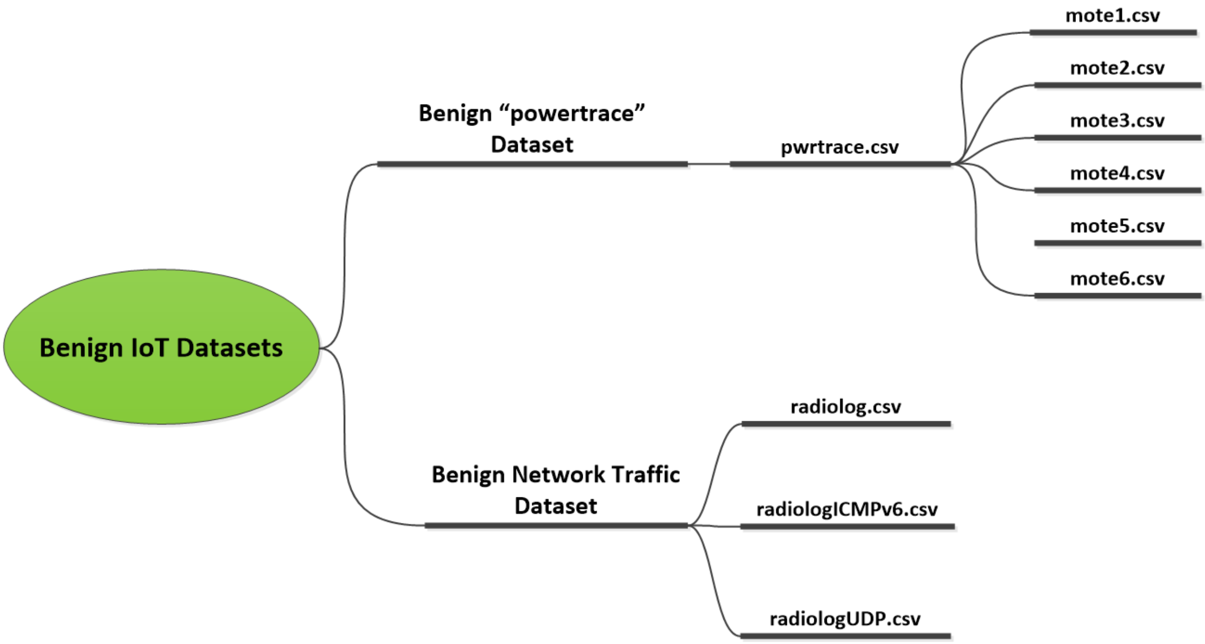


Figure 3.27 Generated Benign IoT Datasets Structure

In principle, the proposed approach in this Chapter can be extended for generating benign IoT datasets from ***j*** different benign scenarios, where each scenario, implemented in the Cooja simulator, may include ***n*** different motes. The generic structure of benign IoT datasets generated according to the proposed approach is shown in Figure 3.28.

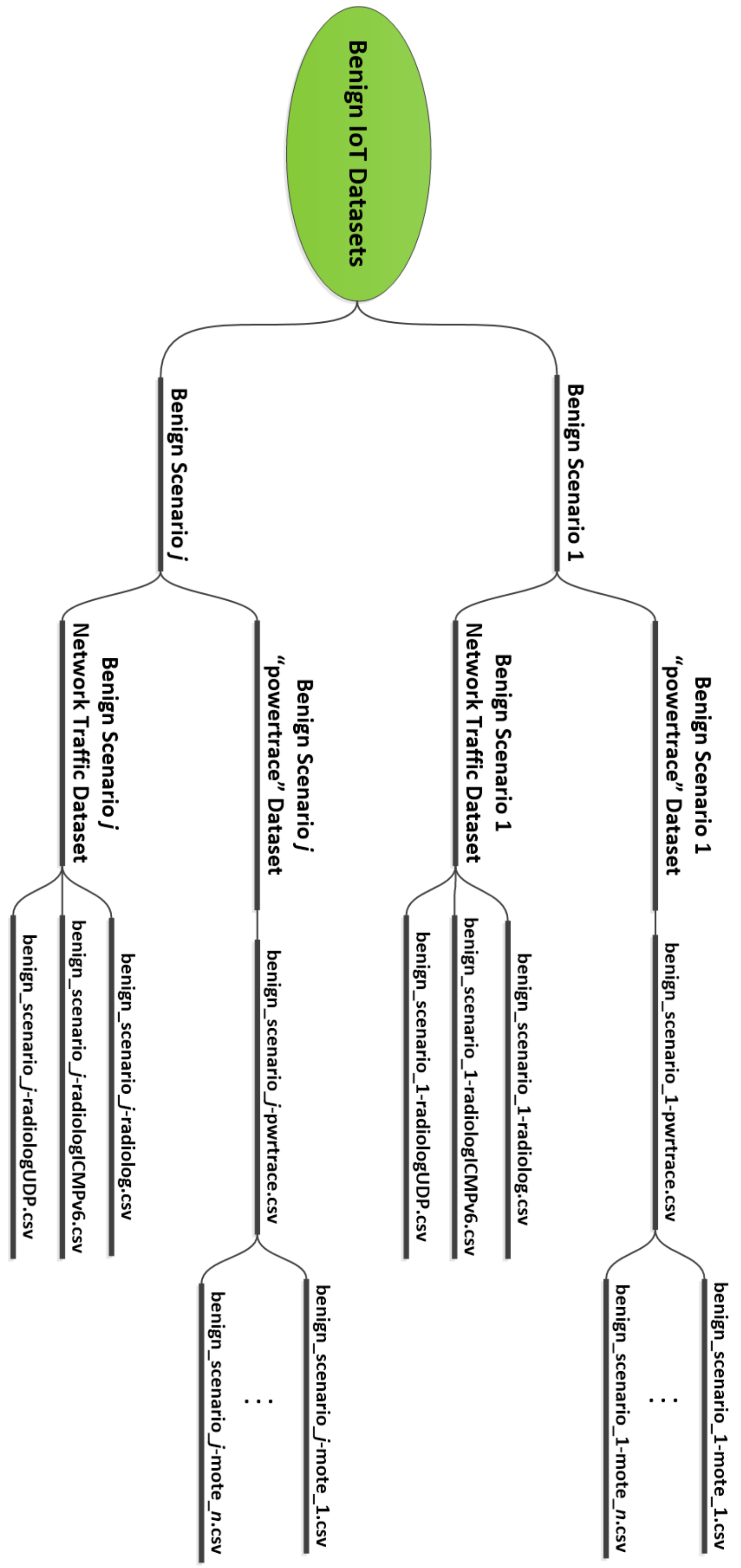

Benign IoT Datasets
Benign Scenario 1
Benign Scenario *j*
Benign Scenario 1 "powertrace" Dataset
Benign Scenario 1 Network Traffic Dataset
Benign Scenario *j* "powertrace" Dataset
Benign Scenario *j* Network Traffic Dataset
benign_scenario_1-pwrtrace.csv
benign_scenario_1-mote_1.csv
. . .
benign_scenario_1-mote_*n*.csv
benign_scenario_1-radiolog.csv
benign_scenario_1-radiologICMPv6.csv
benign_scenario_1-radiologUDP.csv
benign_scenario_*j*-pwrtrace.csv
benign_scenario_*j*-mote_1.csv
. . .
benign_scenario_*j*-mote_*n*.csv
benign_scenario_*j*-radiolog.csv
benign_scenario_*j*-radiologICMPv6.csv
benign_scenario_*j*-radiologUDP.csv

Figure 3.28 Benign IoT Datasets – Generic Structure

This page intentionally left blank.

Chapter 4 Generating Malicious IoT Datasets

4.1 Introduction

This Chapter is focused on the generation of a set of malicious datasets by implementing four scenarios of the following IoT attacks: i) **UDP flooding attack**, ii) **blackhole attack,** iii) **sinkhole attack**, and iv) **sleep deprivation attack**. The implemented scenarios are example scenarios and Cooja has been configured properly to simulate them, as described in Sections 4.2.1, 4.3.1, 4.4.1, and 4.5.1. Similar to the approach followed for the generation of the benign datasets in Chapter 3, the generated IoT-specific information from the simulated attack scenarios was captured from the Contiki plugin "powertrace" (i.e., features such as CPU consumption) and the Cooja tool "Radio messages" (i.e., network traffic features) in order to generate the corresponding "powertrace" and network traffic datasets for the simulated attack scenarios.

4.2 UDP Flooding Attack Datasets

In this Section, we provide a detailed description of the approach followed to generate a set of malicious datasets by implementing a UDP flooding attack scenario in the Cooja simulator, as shown in Figure 4.1.

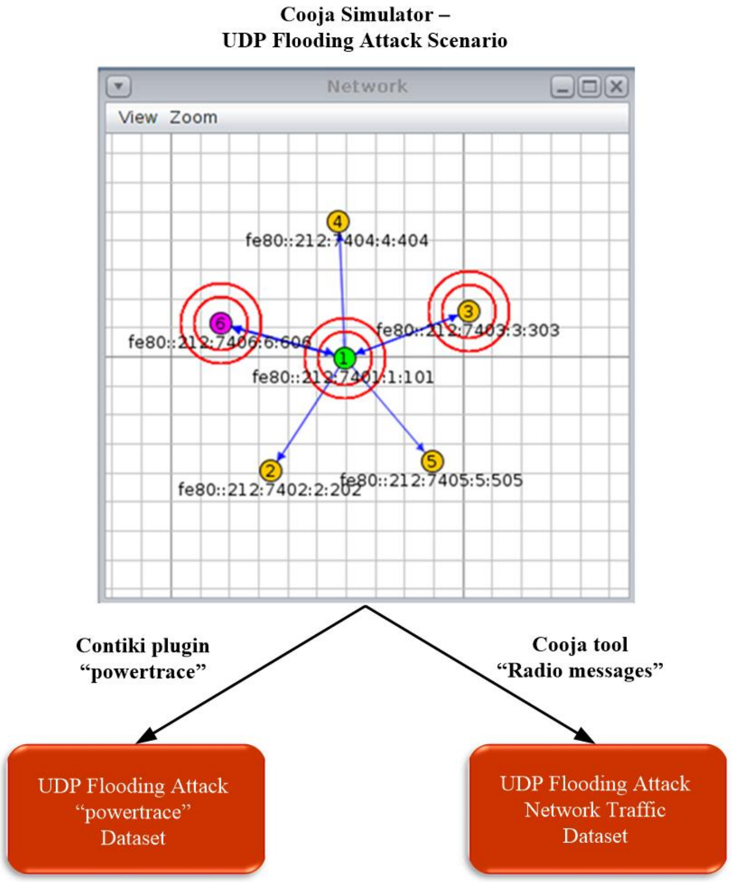
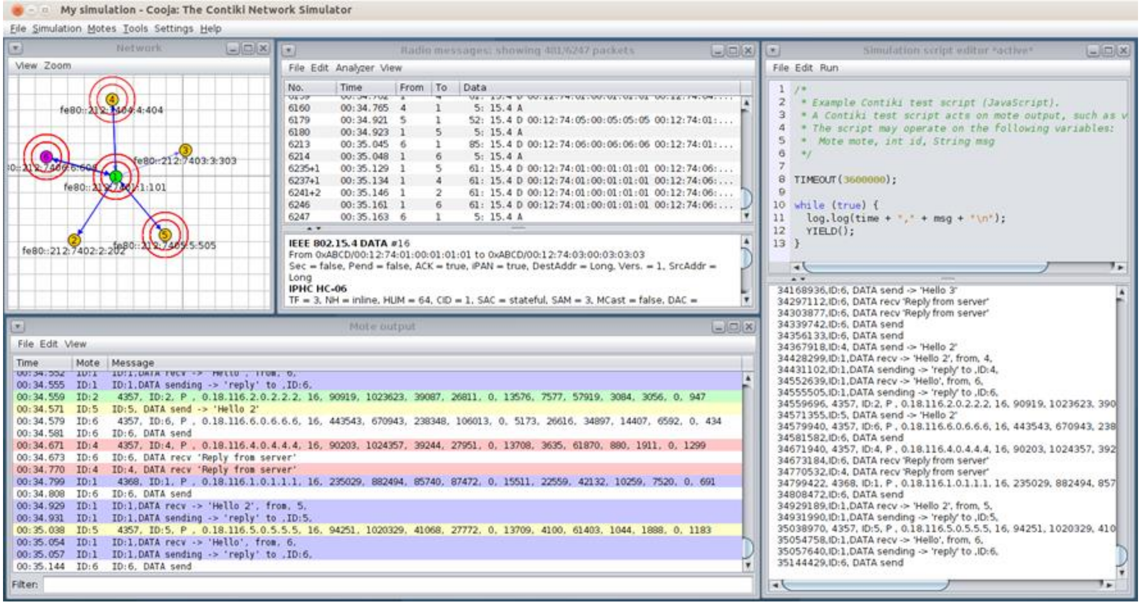

Cooja Simulator –
UDP Flooding Attack Scenario
Network
View Zoom
4
fe80::212:7404:4:404
6
fe80::212:7406:6:606
3
fe80::212:7403:3:303
1
fe80::212:7401:1:101
2
fe80::212:7402:2:202
5
fe80::212:7405:5:505
Contiki plugin "powertrace"
Cooja tool "Radio messages"
UDP Flooding Attack "powertrace" Dataset
UDP Flooding Attack Network Traffic Dataset

Figure 4.1. UDP Flooding Attack Datasets generation by utilising the Cooja simulator.

4.2.1 UDP Flooding Attack Scenario – an example

The network topology of the simulated UDP flooding attack scenario in the Cooja simulator environment consists of 4 yellow (benign) UDP-client motes (i.e., motes 2, 3, 4, and 5), the violet (malicious) UDP-client mote (i.e., mote 6) and the green (benign) UDP-server mote (i.e., mote 1) which is also the target of the attack, as depicted in Figure 4.1. The simulation duration was set to 60 mins and the motes' outputs were printed out in the respective window (e.g., Mote output) while simulations run, as shown in Figure 4.2. Moreover, the 4 yellow (benign) UDP-client motes were configured to send text messages every 10 seconds, approximately, to the UDP-server mote that was configured to provide a corresponding response. On the other hand, the violet (malicious) UDP-client mote (i.e., mote 6) was compromised with malicious code, as shown in Figure 4.3, to send UDP packets within a very short period of time (i.e., every 200ms). Finally, it is noteworthy to say that similar to the benign network scenario, the UDP protocol was used at the Transport Layer, the IPv6 at the network layer, and the type of motes was the Tmote Sky in the UDP flooding attack scenario.



```
No.      Time       From  To   Data
6160     00:34.765  4     1     5: 15.4 A
6179     00:34.921  5     1    52: 15.4 D 00:12:74:05:00:05:05:05 00:12:74:01:...
6180     00:34.923  1     5     5: 15.4 A
6213     00:35.045  6     1    85: 15.4 D 00:12:74:06:00:06:06:06 00:12:74:01:...
6214     00:35.048  1     6     5: 15.4 A
6235+1   00:35.129  1     5    61: 15.4 D 00:12:74:01:00:01:01:01 00:12:74:06:...
6237+1   00:35.134  1     4    61: 15.4 D 00:12:74:01:00:01:01:01 00:12:74:06:...
6241+2   00:35.146  1     2    61: 15.4 D 00:12:74:01:00:01:01:01 00:12:74:06:...
6246     00:35.161  1     6    61: 15.4 D 00:12:74:01:00:01:01:01 00:12:74:06:...
6247     00:35.163  6     1     5: 15.4 A
```

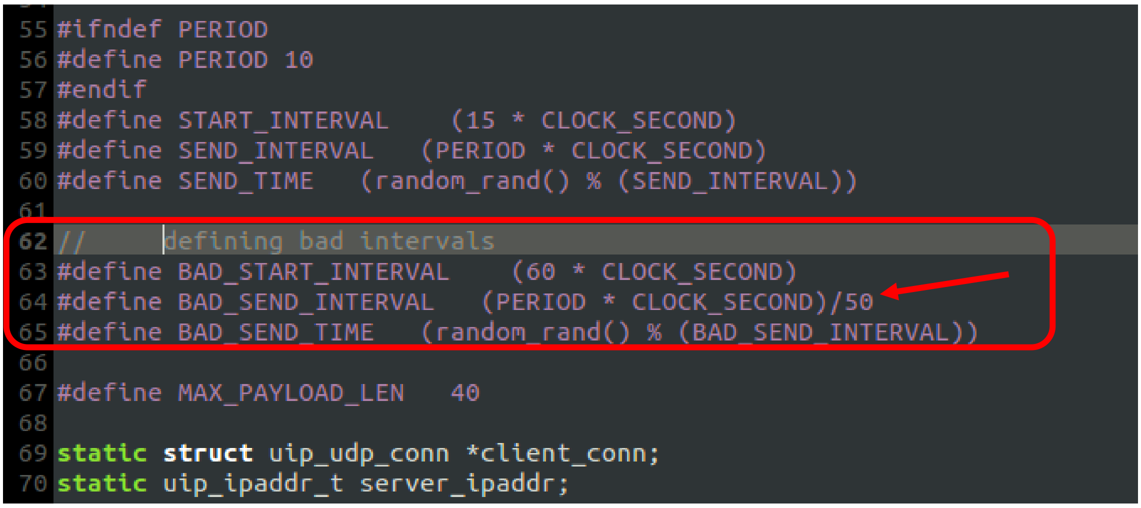
```
IEEE 802.15.4 DATA #16
From 0xABCD/00:12:74:01:00:01:01:01 to 0xABCD/00:12:74:03:00:03:03:03
Sec = false, Pend = false, ACK = true, iPAN = true, DestAddr = Long, Vers. = 1, SrcAddr =
Long
IPHC HC-06
TF = 3, NH = inline, HLIM = 64, CID = 1, SAC = stateful, SAM = 3, MCast = false, DAC =
```

```
/*
 * Example Contiki test script (JavaScript).
 * A Contiki test script acts on mote output, such as v
 * The script may operate on the following variables:
 *  Mote mote, int id, String msg
 */

TIMEOUT(3600000);

while (true) {
  log.log(time + "," + msg + "\n");
  YIELD();
}
```

```
34168936,ID:6, DATA send -> 'Hello 3'
34297112,ID:6, DATA recv 'Reply from server'
34303877,ID:6, DATA recv 'Reply from server'
34339742,ID:6, DATA send
34356133,ID:6, DATA send
34367918,ID:4, DATA send -> 'Hello 2'
34428299,ID:1,DATA recv -> 'Hello 2', from, 4,
34431102,ID:1,DATA sending -> 'reply' to ,ID:4,
34552639,ID:1,DATA recv -> 'Hello', from, 6,
34555505,ID:1,DATA sending -> 'reply' to ,ID:6,
34559696, 4357, ID:2, P , 0.18.116.2.0.2.2.2, 16, 90919, 1023623, 390
34571355,ID:5, DATA send -> 'Hello 2'
34579940, 4357, ID:6, P , 0.18.116.6.0.6.6.6, 16, 443543, 670943, 238
34581582,ID:6, DATA send
34671940, 4357, ID:4, P , 0.18.116.4.0.4.4.4, 16, 90203, 1024357, 392
34673184,ID:6, DATA recv 'Reply from server'
34770532,ID:4, DATA recv 'Reply from server'
34799422, 4368, ID:1, P , 0.18.116.1.0.1.1.1, 16, 235029, 882494, 857
34808472,ID:6, DATA send
34929189,ID:1,DATA recv -> 'Hello 2', from, 5,
34931990,ID:1,DATA sending -> 'reply' to ,ID:5,
35038970, 4357, ID:5, P , 0.18.116.5.0.5.5.5, 16, 94251, 1020329, 410
35054758,ID:1,DATA recv -> 'Hello', from, 6,
35057640,ID:1,DATA sending -> 'reply' to ,ID:6,
35144429,ID:6, DATA send
```

```
Time       Mote  Message
00:34.555  ID:1  ID:1,DATA sending -> 'reply' to ,ID:6,
00:34.559  ID:2   4357, ID:2, P , 0.18.116.2.0.2.2.2, 16, 90919, 1023623, 39087, 26811, 0, 13576, 7577, 57919, 3084, 3056, 0, 947
00:34.571  ID:5  ID:5, DATA send -> 'Hello 2'
00:34.579  ID:6   4357, ID:6, P , 0.18.116.6.0.6.6.6, 16, 443543, 670943, 238348, 106013, 0, 5173, 26616, 34897, 14407, 6592, 0, 434
00:34.581  ID:6  ID:6, DATA send
00:34.671  ID:4   4357, ID:4, P , 0.18.116.4.0.4.4.4, 16, 90203, 1024357, 39244, 27951, 0, 13708, 3635, 61870, 880, 1911, 0, 1299
00:34.673  ID:6  ID:6, DATA recv 'Reply from server'
00:34.770  ID:4  ID:4, DATA recv 'Reply from server'
00:34.799  ID:1   4368, ID:1, P , 0.18.116.1.0.1.1.1, 16, 235029, 882494, 85740, 87472, 0, 15511, 22559, 42132, 10259, 7520, 0, 691
00:34.808  ID:6  ID:6, DATA send
00:34.929  ID:1  ID:1,DATA recv -> 'Hello 2', from, 5,
00:34.931  ID:1  ID:1,DATA sending -> 'reply' to ,ID:5,
00:35.038  ID:5   4357, ID:5, P , 0.18.116.5.0.5.5.5, 16, 94251, 1020329, 41068, 27772, 0, 13709, 4100, 61403, 1044, 1888, 0, 1183
00:35.054  ID:1  ID:1,DATA recv -> 'Hello', from, 6,
00:35.057  ID:1  ID:1,DATA sending -> 'reply' to ,ID:6,
00:35.144  ID:6  ID:6, DATA send
```

Figure 4.2. Cooja Simulator — UDP flooding attack scenario — Motes' outputs



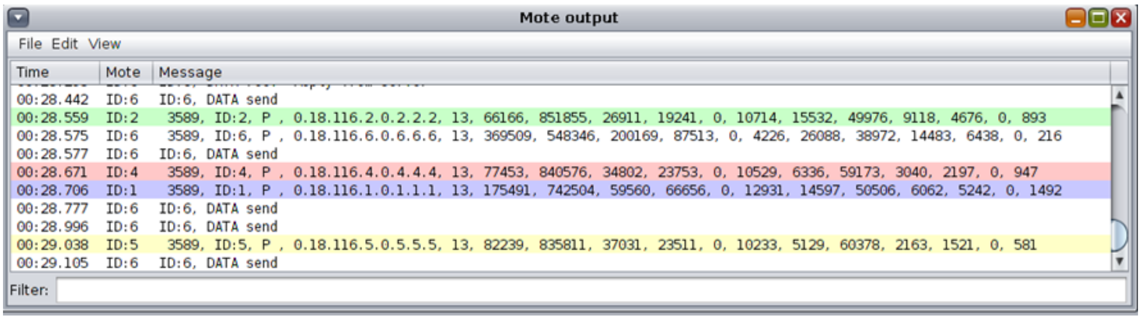
```
#ifndef PERIOD
#define PERIOD 10
#endif
#define START_INTERVAL    (15 * CLOCK_SECOND)
#define SEND_INTERVAL   (PERIOD * CLOCK_SECOND)
#define SEND_TIME   (random_rand() % (SEND_INTERVAL))

//      defining bad intervals
#define BAD_START_INTERVAL    (60 * CLOCK_SECOND)
#define BAD_SEND_INTERVAL   (PERIOD * CLOCK_SECOND)/50
#define BAD_SEND_TIME   (random_rand() % (BAD_SEND_INTERVAL))

#define MAX_PAYLOAD_LEN   40

static struct uip_udp_conn *client_conn;
static uip_ipaddr_t server_ipaddr;
```

Figure 4.3. Malicious code in "udp-client_udp-flood.c" to significantly increase the traffic by 50 times; generating 5 packets per second (i.e., one packet every 200ms) instead of 0.1 packets per second (i.e., one packet every 10 seconds for benign motes).

4.2.2 UDP Flooding Attack “powertrace” Dataset

4.2.2.1 UDP Flooding Attack “powertrace” Dataset – Generation Process

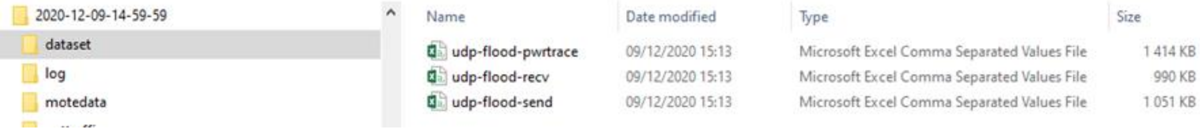
The approach followed for the “powertrace” dataset generation from the UDP flooding attack scenario was similar to the approach followed for the “powertrace” dataset generation from the benign IoT network scenario in Section 3.3.1. In addition, the “powertrace” plugin was similarly enabled for collecting “powertrace” related features, summarised in Table 3, from the motes of the attack scenario every two seconds. In Figure 4.4, the depicted mote output window displays the captured “powertrace” information every two seconds and also the messages sent and received by each mote during the simulation time (60 mins).



```
Time       Mote   Message
00:28.442  ID:6   ID:6, DATA send
00:28.559  ID:2    3589, ID:2, P , 0.18.116.2.0.2.2.2, 13, 66166, 851855, 26911, 19241, 0, 10714, 15532, 49976, 9118, 4676, 0, 893
00:28.575  ID:6    3589, ID:6, P , 0.18.116.6.0.6.6.6, 13, 369509, 548346, 200169, 87513, 0, 4226, 26088, 38972, 14483, 6438, 0, 216
00:28.577  ID:6   ID:6, DATA send
00:28.671  ID:4    3589, ID:4, P , 0.18.116.4.0.4.4.4, 13, 77453, 840576, 34802, 23753, 0, 10529, 6336, 59173, 3040, 2197, 0, 947
00:28.706  ID:1    3589, ID:1, P , 0.18.116.1.0.1.1.1, 13, 175491, 742504, 59560, 66656, 0, 12931, 14597, 50506, 6062, 5242, 0, 1492
00:28.777  ID:6   ID:6, DATA send
00:28.996  ID:6   ID:6, DATA send
00:29.038  ID:5    3589, ID:5, P , 0.18.116.5.0.5.5.5, 13, 82239, 835811, 37031, 23511, 0, 10233, 5129, 60378, 2163, 1521, 0, 581
00:29.105  ID:6   ID:6, DATA send
```

Figure 4.4 Cooja Simulator — UDP flooding attack scenario — Mote output window.

When the timeout occurred, the simulation stopped, and all the captured information and prints were stored in the “COOJA.testlog” file. Afterwards, the “IoT_Simul.sh” file, described in Section 3.3.1, created a) a new root folder named as “2020-12-09-14-59-59”, and b) the “log” folder, inside the “2020-12-09-14-59-59” folder, where the “COOJA.testlog” file was copied from the “…/cooja/build” folder located in the Cooja Simulator. Then, the “IoT_Simul.sh” file following the same process, as described in Section 3.3.1, extracted the required “powertrace” information from the “COOJA.testlog” file and saved it in the “udp-flood-pwrtrace.csv” file in the “dataset” folder that was also created by the batch file inside the “2020-12-09-14-59-59” folder, as shown below in the left part of Figure 4.5. In the “dataset” folder, apart from the “udp-flood-pwrtrace.csv” file, the “IoT_Simul.sh” file generated two more files (i.e., “udp-flood-recv.csv” and “udp-flood-send.csv”), following the same process as in Section 3.1.1. The “udp-flood-recv.csv” file and the “udp-flood-send.csv” file include the “received” and “sent” messages printed by the motes, respectively.



```
2020-12-09-14-59-59
  dataset
  log
  motedata
```

Name	Date modified	Type	Size
udp-flood-pwrtrace	09/12/2020 15:13	Microsoft Excel Comma Separated Values File	1 414 KB
udp-flood-recv	09/12/2020 15:13	Microsoft Excel Comma Separated Values File	990 KB
udp-flood-send	09/12/2020 15:13	Microsoft Excel Comma Separated Values File	1 051 KB

Figure 4.5 Location of the generated “udp-flood-pwrtrace.csv”, “udp-flood-recv.csv”, and “udp-flood-send.csv” files by the “IoT_Simul.sh” file.

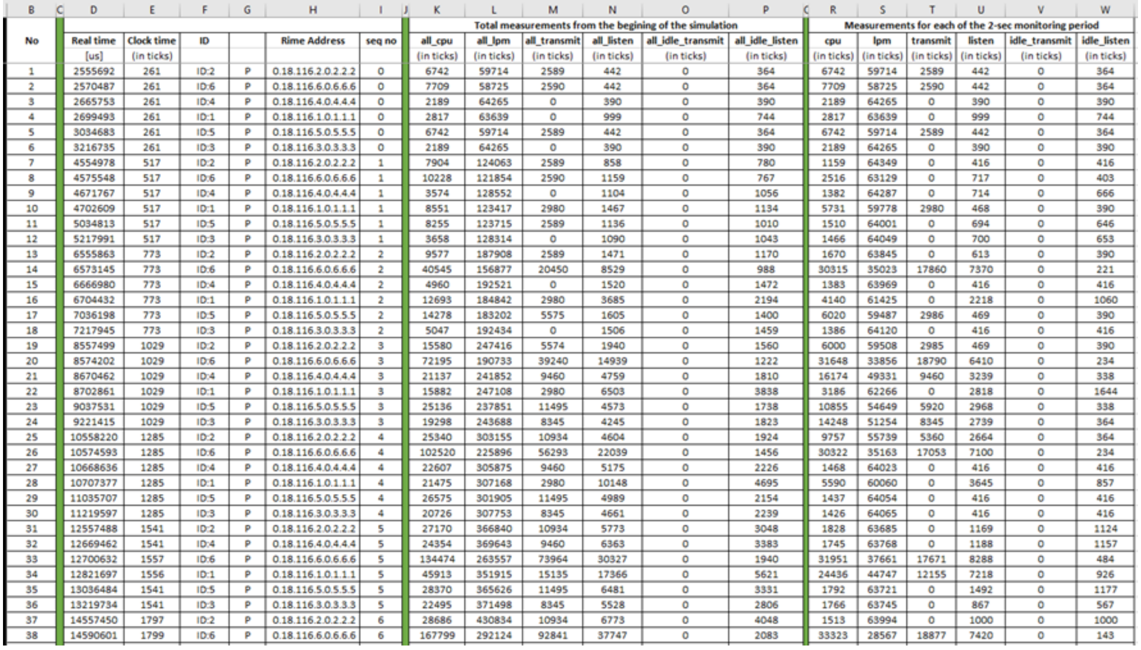
Finally, similar to the benign “powertrace” dataset generation approach in Section 3.3.1, the “IoT_Simul.sh” file extracted the information related to each mote from the “udp-flood-pwrtrace.csv” file and generated one csv file for each mote with the corresponding information from the “udp-flood-pwrtrace.csv” file. The generated six csv files (i.e., “udp-flood-mote1.csv”,…, “udp-flood-mote6.csv”) were stored in the “motedata” folder, created also by the “IoT_Simul.sh” file, as shown in the left part of Figure 4.5.

4.2.2.2 UDP Flooding Attack "powertrace" Dataset – Generated Results

The UDP flooding attack "powertrace" dataset consists of the following csv files: "udp-flood-pwrtrace.csv", "udp-flood-mote1.csv", "udp-flood-mote2.csv" "udp-flood-mote3.csv" "udp-flood-mote4.csv" "udp-flood-mote5.csv", and "udp-flood-mote6.csv". In this Section, we present sets of records from the "udp-flood-pwrtrace.csv", and in Appendix 1 we present sets of records from "udp-flood-mote1.csv", "udp-flood-mote2.csv" and "udp-flood-mote6.csv" files.

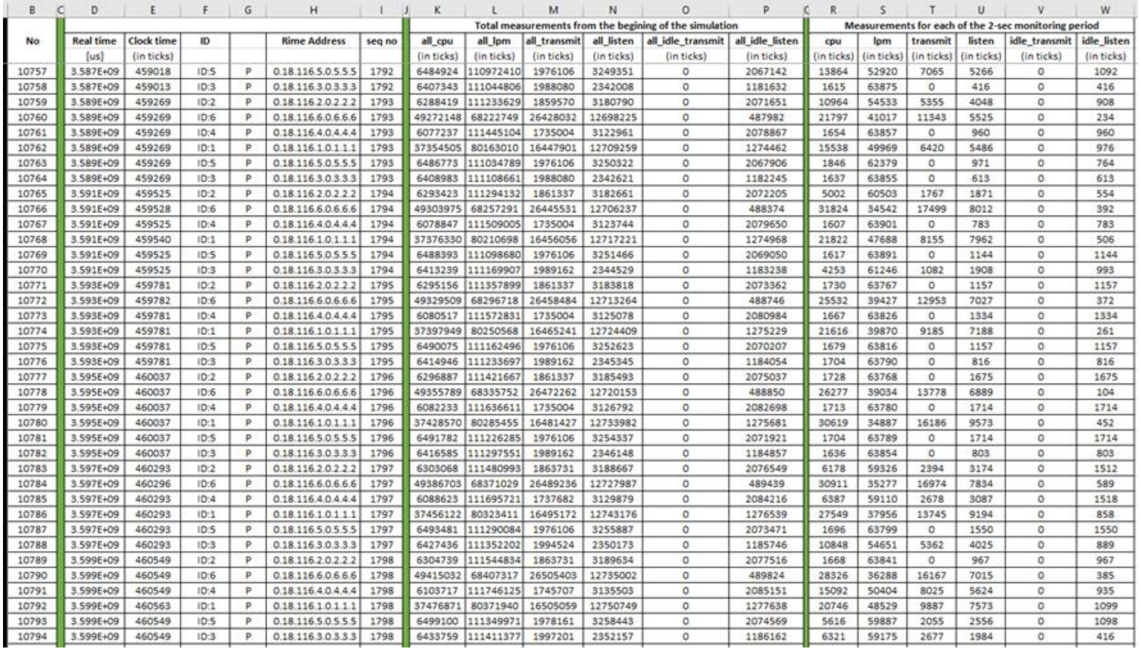
4.2.2.2.1 "udp-flood-pwrtrace.csv"

The generated malicious "udp-flood-pwrtrace.csv" file consists of 10,794 records and its first 38 records (i.e., 1–38) and its last 38 records (i.e., 10,757–10,794) are depicted in Figure 4.6 and Figure 4.7, respectively.



B	D	E	F	G	H	I	K	L	M	N	O	P	R	S	T	U	V	W
							Total measurements from the begining of the simulation						Measurements for each of the 2-sec monitoring period					
No	Real time	Clock time	ID		Rime Address	seq no	all_cpu	all_lpm	all_transmit	all_listen	all_idle_transmit	all_idle_listen	cpu	lpm	transmit	listen	idle_transmit	idle_listen
	[us]	(in ticks)					(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)
1	2555692	261	ID:2	P	0.18.116.2.0.2.2.2	0	6742	59714	2589	442	0	364	6742	59714	2589	442	0	364
2	2570487	261	ID:6	P	0.18.116.6.0.6.6.6	0	7709	58725	2590	442	0	364	7709	58725	2590	442	0	364
3	2665753	261	ID:4	P	0.18.116.4.0.4.4.4	0	2189	64265	0	390	0	390	2189	64265	0	390	0	390
4	2699493	261	ID:1	P	0.18.116.1.0.1.1.1	0	2817	63639	0	999	0	744	2817	63639	0	999	0	744
5	3034683	261	ID:5	P	0.18.116.5.0.5.5.5	0	6742	59714	2589	442	0	364	6742	59714	2589	442	0	364
6	3216735	261	ID:3	P	0.18.116.3.0.3.3.3	0	2189	64265	0	390	0	390	2189	64265	0	390	0	390
7	4554978	517	ID:2	P	0.18.116.2.0.2.2.2	1	7904	124063	2589	858	0	780	1159	64349	0	416	0	416
8	4575548	517	ID:6	P	0.18.116.6.0.6.6.6	1	10228	121854	2590	1159	0	767	2516	63129	0	717	0	403
9	4671767	517	ID:4	P	0.18.116.4.0.4.4.4	1	3574	128552	0	1104	0	1056	1382	64287	0	714	0	666
10	4702609	517	ID:1	P	0.18.116.1.0.1.1.1	1	8551	123417	2980	1467	0	1134	5731	59778	2980	468	0	390
11	5034813	517	ID:5	P	0.18.116.5.0.5.5.5	1	8255	123715	2589	1136	0	1010	1510	64001	0	694	0	646
12	5217991	517	ID:3	P	0.18.116.3.0.3.3.3	1	3658	128314	0	1090	0	1043	1466	64049	0	700	0	653
13	6555863	773	ID:2	P	0.18.116.2.0.2.2.2	2	9577	187908	2589	1471	0	1170	1670	63845	0	613	0	390
14	6573145	773	ID:6	P	0.18.116.6.0.6.6.6	2	40545	156877	20450	8529	0	988	30315	35023	17860	7370	0	221
15	6666980	773	ID:4	P	0.18.116.4.0.4.4.4	2	4960	192521	0	1520	0	1472	1383	63969	0	416	0	416
16	6704432	773	ID:1	P	0.18.116.1.0.1.1.1	2	12693	184842	2980	3685	0	2194	4140	61425	0	2218	0	1060
17	7036198	773	ID:5	P	0.18.116.5.0.5.5.5	2	14278	183202	5575	1605	0	1400	6020	59487	2986	469	0	390
18	7217945	773	ID:3	P	0.18.116.3.0.3.3.3	2	5047	192434	0	1506	0	1459	1386	64120	0	416	0	416
19	8557499	1029	ID:2	P	0.18.116.2.0.2.2.2	3	15580	247416	5574	1940	0	1560	6000	59508	2985	469	0	390
20	8574202	1029	ID:6	P	0.18.116.6.0.6.6.6	3	72195	190733	39240	14939	0	1222	31648	33856	18790	6410	0	234
21	8670462	1029	ID:4	P	0.18.116.4.0.4.4.4	3	21137	241852	9460	4759	0	1810	16174	49331	9460	3239	0	338
22	8702861	1029	ID:1	P	0.18.116.1.0.1.1.1	3	15882	247108	2980	6503	0	3838	3186	62266	0	2818	0	1644
23	9037531	1029	ID:5	P	0.18.116.5.0.5.5.5	3	25136	237851	11495	4573	0	1738	10855	54649	5920	2968	0	338
24	9221415	1029	ID:3	P	0.18.116.3.0.3.3.3	3	19298	243688	8345	4245	0	1823	14248	51254	8345	2739	0	364
25	10558220	1285	ID:2	P	0.18.116.2.0.2.2.2	4	25340	303155	10934	4604	0	1924	9757	55739	5360	2664	0	364
26	10574593	1285	ID:6	P	0.18.116.6.0.6.6.6	4	102520	225896	56293	22039	0	1456	30322	35163	17053	7100	0	234
27	10668636	1285	ID:4	P	0.18.116.4.0.4.4.4	4	22607	305875	9460	5175	0	2226	1468	64023	0	416	0	416
28	10707377	1285	ID:1	P	0.18.116.1.0.1.1.1	4	21475	307168	2980	10148	0	4695	5590	60060	0	3645	0	857
29	11035707	1285	ID:5	P	0.18.116.5.0.5.5.5	4	26575	301905	11495	4989	0	2154	1437	64054	0	416	0	416
30	11219597	1285	ID:3	P	0.18.116.3.0.3.3.3	4	20726	307753	8345	4661	0	2239	1426	64065	0	416	0	416
31	12557488	1541	ID:2	P	0.18.116.2.0.2.2.2	5	27170	366840	10934	5773	0	3048	1828	63685	0	1169	0	1124
32	12669462	1541	ID:4	P	0.18.116.4.0.4.4.4	5	24354	369643	9460	6363	0	3383	1745	63768	0	1188	0	1157
33	12700632	1557	ID:6	P	0.18.116.6.0.6.6.6	5	134474	263557	73964	30327	0	1940	31951	37661	17671	8288	0	484
34	12821697	1556	ID:1	P	0.18.116.1.0.1.1.1	5	45913	351915	15135	17366	0	5621	24436	44747	12155	7218	0	926
35	13036484	1541	ID:5	P	0.18.116.5.0.5.5.5	5	28370	365626	11495	6481	0	3331	1792	63721	0	1492	0	1177
36	13219734	1541	ID:3	P	0.18.116.3.0.3.3.3	5	22495	371498	8345	5528	0	2806	1766	63745	0	867	0	567
37	14557450	1797	ID:2	P	0.18.116.2.0.2.2.2	6	28686	430834	10934	6773	0	4048	1513	63994	0	1000	0	1000
38	14590601	1799	ID:6	P	0.18.116.6.0.6.6.6	6	167799	292124	92841	37747	0	2083	33323	28567	18877	7420	0	143

Figure 4.6 Malicious "udp-flood-pwrtrace.csv"—1 to 38 records.



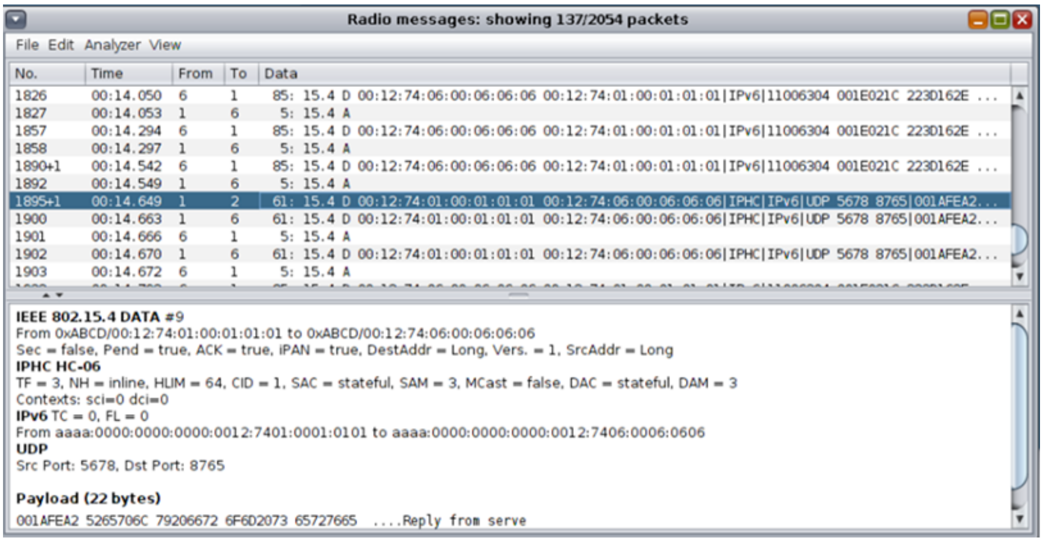
B	D	E	F	G	H	I	K	L	M	N	O	P	R	S	T	U	V	W
							Total measurements from the begining of the simulation						Measurements for each of the 2-sec monitoring period					
No	Real time	Clock time	ID		Rime Address	seq no	all_cpu	all_lpm	all_transmit	all_listen	all_idle_transmit	all_idle_listen	cpu	lpm	transmit	listen	idle_transmit	idle_listen
	[us]	(in ticks)					(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)
10757	3.587E+09	459018	ID:5	P	0.18.116.5.0.5.5.5	1792	6484924	110972410	1976106	3249351	0	2067142	13864	52920	7065	5266	0	1092
10758	3.587E+09	459013	ID:3	P	0.18.116.3.0.3.3.3	1792	6407343	111044806	1988080	2342008	0	1181632	1615	63875	0	416	0	416
10759	3.589E+09	459269	ID:2	P	0.18.116.2.0.2.2.2	1793	6288419	111233629	1859570	3180790	0	2071651	10964	54533	5355	4048	0	908
10760	3.589E+09	459269	ID:6	P	0.18.116.6.0.6.6.6	1793	49272148	68222749	26428032	12698225	0	487982	21797	41017	11343	5525	0	234
10761	3.589E+09	459269	ID:4	P	0.18.116.4.0.4.4.4	1793	6077237	111445104	1735004	3122961	0	2078867	1654	63857	0	960	0	960
10762	3.589E+09	459269	ID:1	P	0.18.116.1.0.1.1.1	1793	37354505	80163010	16447901	12709259	0	1274462	15538	49969	6420	5486	0	976
10763	3.589E+09	459269	ID:5	P	0.18.116.5.0.5.5.5	1793	6486773	111034789	1976106	3250322	0	2067906	1846	62379	0	971	0	764
10764	3.589E+09	459269	ID:3	P	0.18.116.3.0.3.3.3	1793	6408983	111108661	1988080	2342621	0	1182245	1637	63855	0	613	0	613
10765	3.591E+09	459525	ID:2	P	0.18.116.2.0.2.2.2	1794	6293423	111294132	1861337	3182661	0	2072205	5002	60503	1767	1871	0	554
10766	3.591E+09	459528	ID:6	P	0.18.116.6.0.6.6.6	1794	49303975	68257291	26445531	12706237	0	488374	31824	34542	17499	8012	0	392
10767	3.591E+09	459525	ID:4	P	0.18.116.4.0.4.4.4	1794	6078847	111509005	1735004	3123744	0	2079650	1607	63901	0	783	0	783
10768	3.591E+09	459540	ID:1	P	0.18.116.1.0.1.1.1	1794	37376330	80210698	16456056	12717221	0	1274968	21822	47688	8155	7962	0	506
10769	3.591E+09	459525	ID:5	P	0.18.116.5.0.5.5.5	1794	6488393	111098680	1976106	3251466	0	2069050	1617	63891	0	1144	0	1144
10770	3.591E+09	459525	ID:3	P	0.18.116.3.0.3.3.3	1794	6413239	111169907	1989162	2344529	0	1183238	4253	61246	1082	1908	0	993
10771	3.593E+09	459781	ID:2	P	0.18.116.2.0.2.2.2	1795	6295156	111357899	1861337	3183818	0	2073362	1730	63767	0	1157	0	1157
10772	3.593E+09	459782	ID:6	P	0.18.116.6.0.6.6.6	1795	49329509	68296718	26458484	12713264	0	488746	25532	39427	12953	7027	0	372
10773	3.593E+09	459781	ID:4	P	0.18.116.4.0.4.4.4	1795	6080517	111572831	1735004	3125078	0	2080984	1667	63826	0	1334	0	1334
10774	3.593E+09	459781	ID:1	P	0.18.116.1.0.1.1.1	1795	37397949	80250568	16465241	12724409	0	1275229	21616	39870	9185	7188	0	261
10775	3.593E+09	459781	ID:5	P	0.18.116.5.0.5.5.5	1795	6490075	111162496	1976106	3252623	0	2070207	1679	63816	0	1157	0	1157
10776	3.593E+09	459781	ID:3	P	0.18.116.3.0.3.3.3	1795	6414946	111233697	1989162	2345345	0	1184054	1704	63790	0	816	0	816
10777	3.595E+09	460037	ID:2	P	0.18.116.2.0.2.2.2	1796	6296887	111421667	1861337	3185493	0	2075037	1728	63768	0	1675	0	1675
10778	3.595E+09	460037	ID:6	P	0.18.116.6.0.6.6.6	1796	49355789	68335752	26472262	12720153	0	488850	26277	39034	13778	6889	0	104
10779	3.595E+09	460037	ID:4	P	0.18.116.4.0.4.4.4	1796	6082233	111636611	1735004	3126792	0	2082698	1713	63780	0	1714	0	1714
10780	3.595E+09	460037	ID:1	P	0.18.116.1.0.1.1.1	1796	37428570	80285455	16481427	12733982	0	1275681	30619	34887	16186	9573	0	452
10781	3.595E+09	460037	ID:5	P	0.18.116.5.0.5.5.5	1796	6491782	111226285	1976106	3254337	0	2071921	1704	63789	0	1714	0	1714
10782	3.595E+09	460037	ID:3	P	0.18.116.3.0.3.3.3	1796	6416585	111297551	1989162	2346148	0	1184857	1636	63854	0	803	0	803
10783	3.597E+09	460293	ID:2	P	0.18.116.2.0.2.2.2	1797	6303068	111480993	1863731	3188667	0	2076549	6178	59326	2394	3174	0	1512
10784	3.597E+09	460296	ID:6	P	0.18.116.6.0.6.6.6	1797	49386703	68371029	26489236	12727987	0	489439	30911	35277	16974	7834	0	589
10785	3.597E+09	460293	ID:4	P	0.18.116.4.0.4.4.4	1797	6088623	111695721	1737682	3129879	0	2084216	6387	59110	2678	3087	0	1518
10786	3.597E+09	460293	ID:1	P	0.18.116.1.0.1.1.1	1797	37456122	80323411	16495172	12743176	0	1276539	27549	37956	13745	9194	0	858
10787	3.597E+09	460293	ID:5	P	0.18.116.5.0.5.5.5	1797	6493481	111290084	1976106	3255887	0	2073471	1696	63799	0	1550	0	1550
10788	3.597E+09	460293	ID:3	P	0.18.116.3.0.3.3.3	1797	6427436	111352202	1994524	2350173	0	1185746	10848	54651	5362	4025	0	889
10789	3.599E+09	460549	ID:2	P	0.18.116.2.0.2.2.2	1798	6304739	111544834	1863731	3189634	0	2077516	1668	63841	0	967	0	967
10790	3.599E+09	460549	ID:6	P	0.18.116.6.0.6.6.6	1798	49415032	68407317	26505403	12735002	0	489824	28326	36288	16167	7015	0	385
10791	3.599E+09	460549	ID:4	P	0.18.116.4.0.4.4.4	1798	6103717	111746125	1745707	3135503	0	2085151	15092	50404	8025	5624	0	935
10792	3.599E+09	460563	ID:1	P	0.18.116.1.0.1.1.1	1798	37476871	80371940	16505059	12750749	0	1277638	20746	48529	9887	7573	0	1099
10793	3.599E+09	460549	ID:5	P	0.18.116.5.0.5.5.5	1798	6499100	111349971	1978161	3258443	0	2074569	5616	59887	2055	2556	0	1098
10794	3.599E+09	460549	ID:3	P	0.18.116.3.0.3.3.3	1798	6433759	111411377	1997201	2352157	0	1186162	6321	59175	2677	1984	0	416

Figure 4.7 Malicious "udp-flood-pwrtrace.csv"—10,757 to 10,794 records.

4.2.3 UDP Flooding Attack Network Traffic Dataset

4.2.3.1 UDP Flooding Attack Network Traffic Dataset – Generation Process

The approach followed for the network traffic dataset generation from the UDP flooding attack scenario was similar to the approach followed for the network traffic dataset generation from the benign IoT network scenario in Section 3.4.1. The “Radio messages” tool, provided by the Cooja simulator, was similarly used for collecting data related to the corresponding network traffic features (e.g., source/destination IPv6 address, packet size, and protocol) from the network of the attack scenario. During the simulation, the network traffic information was being shown in the top part of the “Radio messages” output window as depicted in the top part of Figure 4.8.



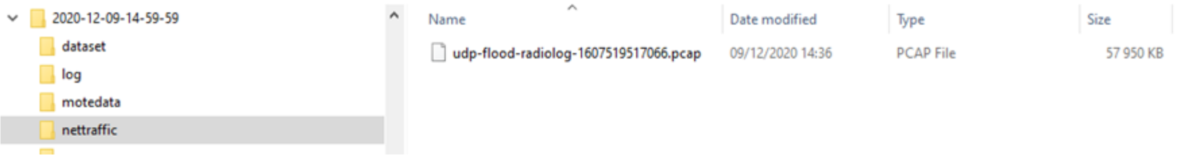
Radio messages: showing 137/2054 packets

No.	Time	From	To	Data
1826	00:14.050	6	1	85: 15.4 D 00:12:74:06:00:06:06:06 00:12:74:01:00:01:01:01|IPv6|11006304 001E021C 223D162E ...
1827	00:14.053	1	6	5: 15.4 A
1857	00:14.294	6	1	85: 15.4 D 00:12:74:06:00:06:06:06 00:12:74:01:00:01:01:01|IPv6|11006304 001E021C 223D162E ...
1858	00:14.297	1	6	5: 15.4 A
1890+1	00:14.542	6	1	85: 15.4 D 00:12:74:06:00:06:06:06 00:12:74:01:00:01:01:01|IPv6|11006304 001E021C 223D162E ...
1892	00:14.549	1	6	5: 15.4 A
1895+1	00:14.649	1	2	61: 15.4 D 00:12:74:01:00:01:01:01 00:12:74:06:00:06:06:06|IPHC|IPv6|UDP 5678 8765|001AFEA2...
1900	00:14.663	1	6	61: 15.4 D 00:12:74:01:00:01:01:01 00:12:74:06:00:06:06:06|IPHC|IPv6|UDP 5678 8765|001AFEA2...
1901	00:14.666	6	1	5: 15.4 A
1902	00:14.670	1	6	61: 15.4 D 00:12:74:01:00:01:01:01 00:12:74:06:00:06:06:06|IPHC|IPv6|UDP 5678 8765|001AFEA2...
1903	00:14.672	6	1	5: 15.4 A

```
IEEE 802.15.4 DATA #9
From 0xABCD/00:12:74:01:00:01:01:01 to 0xABCD/00:12:74:06:00:06:06:06
Sec = false, Pend = true, ACK = true, iPAN = true, DestAddr = Long, Vers. = 1, SrcAddr = Long
IPHC HC-06
TF = 3, NH = inline, HLIM = 64, CID = 1, SAC = stateful, SAM = 3, MCast = false, DAC = stateful, DAM = 3
Contexts: sci=0 dci=0
IPv6 TC = 0, FL = 0
From aaaa:0000:0000:0000:0012:7401:0001:0101 to aaaa:0000:0000:0000:0012:7406:0006:0606
UDP
Src Port: 5678, Dst Port: 8765

Payload (22 bytes)
001AFEA2 5265706C 79206672 6F6D2073 65727665  ....Reply from serve
```

Figure 4.8 Network traffic information from the UDP flooding attack scenario in the “Radio messages” output window.

When the simulation stopped, the generated pcap file was saved as “radiolog-1607519517066.pcap” within the “…/cooja/build” folder. Afterwards, the “IoT_Simul.sh” file, described in Section 3.4.1, created a) a new root folder named as “2020-12-09-14-59-59”, and b) the “nettraffic” folder, inside the “2020-12-09-14-59-59” folder, where the “radiolog-1607519517066.pcap” file, copied from the “…/cooja/build” folder located in the Cooja Simulator, was saved as “udp-flood-radiolog-1607519517066.pcap”. The “nettraffic” folder inside the root folder “2020-12-09-14-59-59” and the “udp-flood-radiolog-1607519517066.pcap” file in the “nettraffic” folder are shown in Figure 4.9.

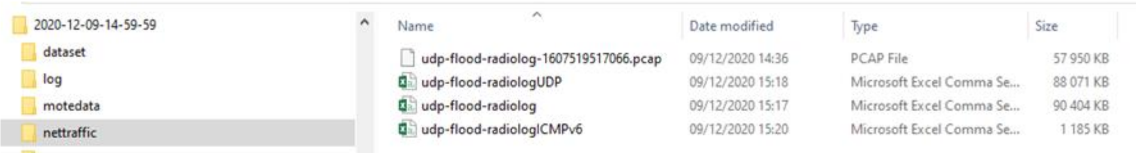


- 2020-12-09-14-59-59
  - dataset
  - log
  - motedata
  - nettraffic

Name	Date modified	Type	Size
udp-flood-radiolog-1607519517066.pcap	09/12/2020 14:36	PCAP File	57 950 KB

Figure 4.9 The “nettraffic” folder inside the root folder “2020-12-09-14-59-59” and the “udp-flood-radiolog-1607519517066.pcap” file.

Then, following the same process, as described in Section 3.4.1, we used Wireshark to extract the stored network traffic information from the “udp-flood-radiolog-1607519517066.pcap” file to the “udp-flood-radiolog.csv” file stored in the “nettraffic” folder as shown in Figure 4.10.



	Name	Date modified	Type	Size
2020-12-09-14-59-59	udp-flood-radiolog-1607519517066.pcap	09/12/2020 14:36	PCAP File	57 950 KB
dataset	udp-flood-radiologUDP	09/12/2020 15:18	Microsoft Excel Comma Se...	88 071 KB
log	udp-flood-radiolog	09/12/2020 15:17	Microsoft Excel Comma Se...	90 404 KB
motedata	udp-flood-radiologICMPv6	09/12/2020 15:20	Microsoft Excel Comma Se...	1 185 KB
nettraffic				

Figure 4.10 The “nettraffic” folder inside the root folder “2020-12-09-14-59-59” and its included files.

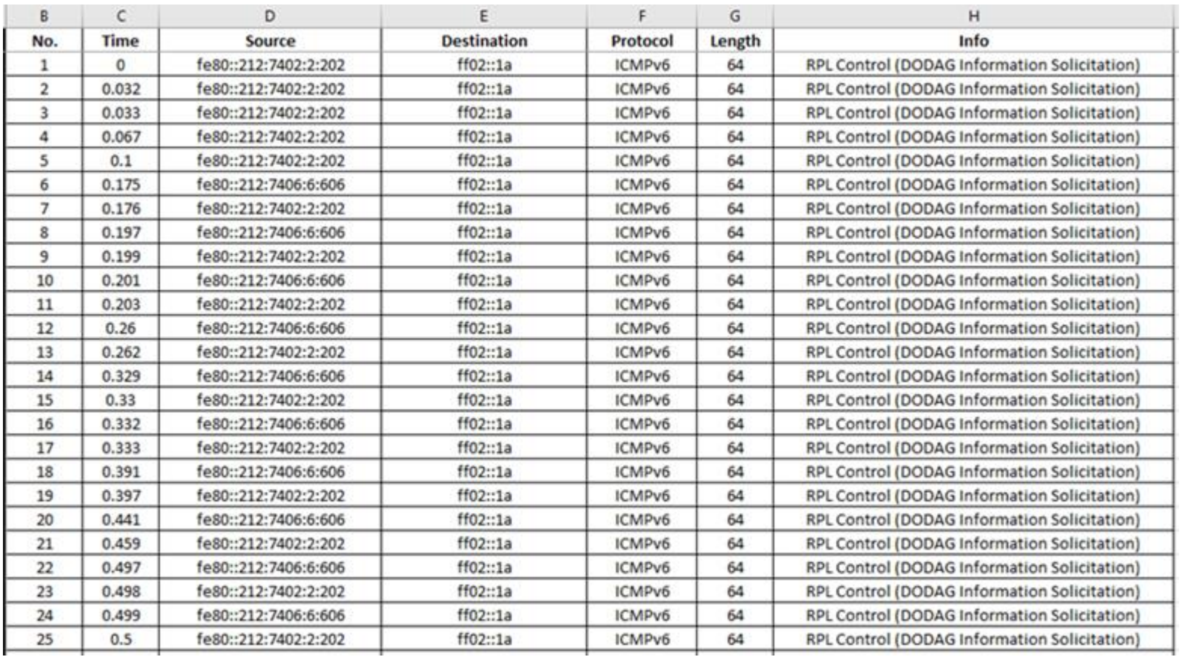
In the “nettraffic” folder, apart from the “udp-flood-radiolog.csv” file, we also used Wireshark, following the same process as in Section 3.4.1, to generate two more files (i.e., the “udp-flood-radiologICMPv6.csv” file and the “udp-flood-radiologUDP.csv” file) from the “udp-flood-radiolog-1607519517066.pcap” file.

4.2.3.2 UDP Flooding Attack Network Traffic Dataset – Generated Results

The UDP flooding attack network traffic dataset consists of the following csv files which are located in the “nettraffic” folder as described in Section 4.2.3.1: “udp-flood-radiolog.csv”, “udp-flood-radiologICMPv6.csv”, and “udp-flood-radiologUDP.csv” files. In this Section, we present sets of records from these files.

4.2.3.2.1 “udp-flood-radiolog.csv”

The generated malicious “udp-flood-radiolog.csv” file consists of 702,332 records and its first 25 records (i.e., 1–25) are depicted below in Figure 4.11.

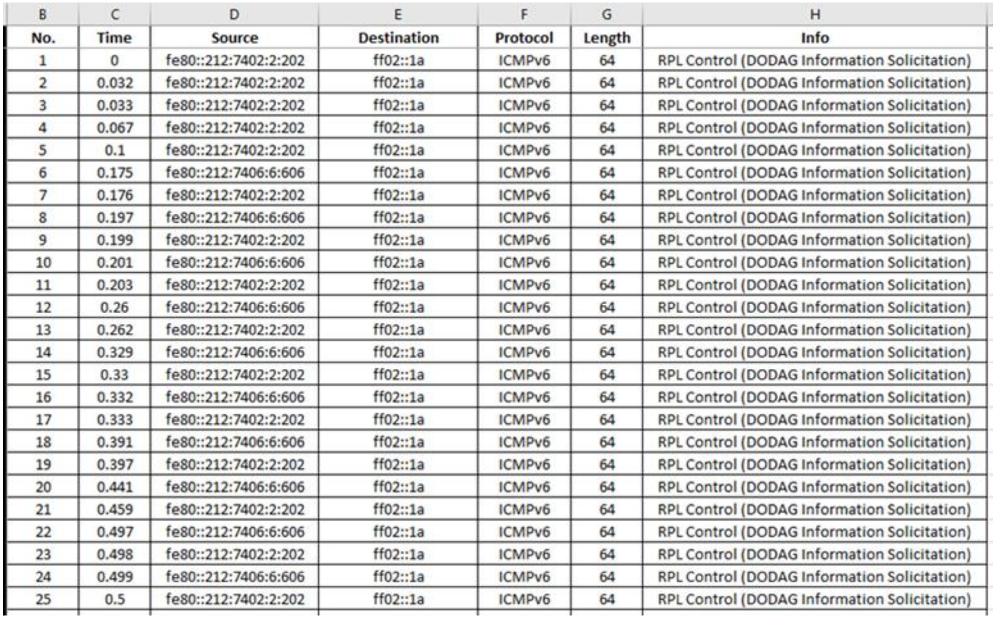


B	C	D	E	F	G	H
No.	**Time**	**Source**	**Destination**	**Protocol**	**Length**	**Info**
1	0	fe80::212:7402:2:202	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
2	0.032	fe80::212:7402:2:202	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
3	0.033	fe80::212:7402:2:202	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
4	0.067	fe80::212:7402:2:202	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
5	0.1	fe80::212:7402:2:202	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
6	0.175	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
7	0.176	fe80::212:7402:2:202	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
8	0.197	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
9	0.199	fe80::212:7402:2:202	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
10	0.201	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
11	0.203	fe80::212:7402:2:202	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
12	0.26	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
13	0.262	fe80::212:7402:2:202	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
14	0.329	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
15	0.33	fe80::212:7402:2:202	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
16	0.332	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
17	0.333	fe80::212:7402:2:202	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
18	0.391	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
19	0.397	fe80::212:7402:2:202	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
20	0.441	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
21	0.459	fe80::212:7402:2:202	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
22	0.497	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
23	0.498	fe80::212:7402:2:202	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
24	0.499	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
25	0.5	fe80::212:7402:2:202	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)

Figure 4.11 Malicious “udp-flood-radiolog.csv”—1 to 25 records.

4.2.3.2.2 “udp-flood-radiologICMPv6.csv”

The generated malicious “udp-flood-radiologICMPv6.csv” file consists of 9,908 records and its first 25 records (i.e., 1–25) are depicted below in Figure 4.12.

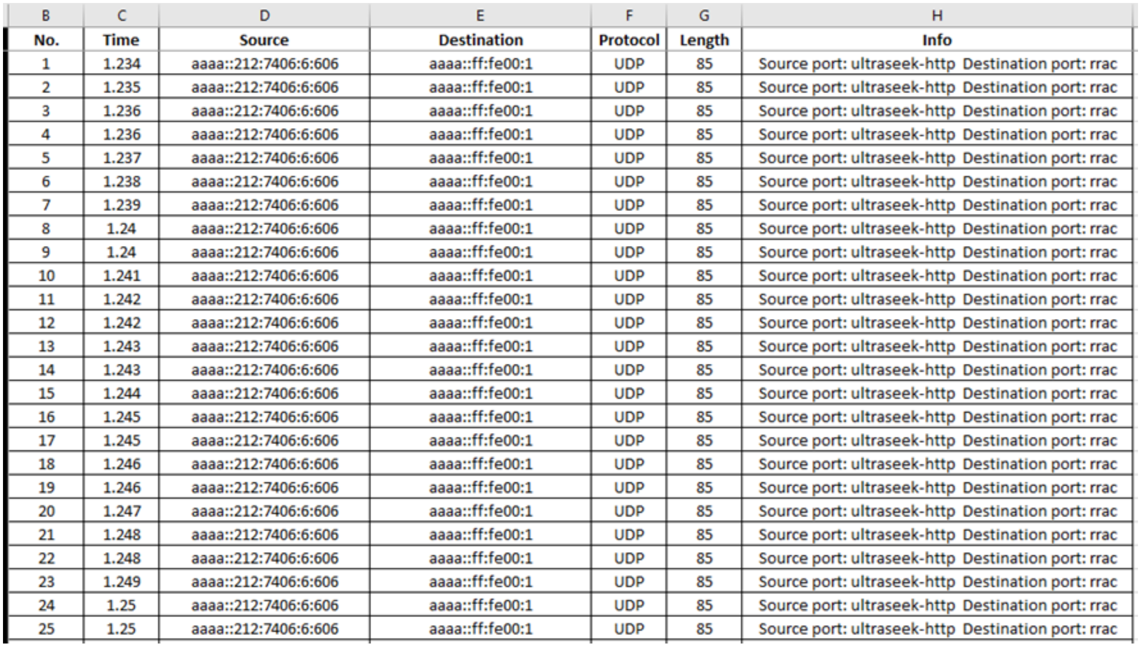


B	C	D	E	F	G	H
No.	Time	Source	Destination	Protocol	Length	Info
1	0	fe80::212:7402:2:202	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
2	0.032	fe80::212:7402:2:202	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
3	0.033	fe80::212:7402:2:202	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
4	0.067	fe80::212:7402:2:202	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
5	0.1	fe80::212:7402:2:202	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
6	0.175	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
7	0.176	fe80::212:7402:2:202	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
8	0.197	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
9	0.199	fe80::212:7402:2:202	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
10	0.201	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
11	0.203	fe80::212:7402:2:202	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
12	0.26	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
13	0.262	fe80::212:7402:2:202	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
14	0.329	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
15	0.33	fe80::212:7402:2:202	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
16	0.332	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
17	0.333	fe80::212:7402:2:202	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
18	0.391	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
19	0.397	fe80::212:7402:2:202	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
20	0.441	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
21	0.459	fe80::212:7402:2:202	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
22	0.497	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
23	0.498	fe80::212:7402:2:202	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
24	0.499	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
25	0.5	fe80::212:7402:2:202	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)

Figure 4.12 Malicious “udp-flood-radiologICMPv6.csv”—1 to 25 records.

4.2.3.2.3 “udp-flood-radiologUDP.csv”

The generated malicious “udp-flood-radiologUDP.csv” file consists of 670,671 records and its first 25 records (i.e., 1–25) are depicted below in Figure 4.13.



B	C	D	E	F	G	H
No.	Time	Source	Destination	Protocol	Length	Info
1	1.234	aaaa::212:7406:6:606	aaaa::ff:fe00:1	UDP	85	Source port: ultraseek-http Destination port: rrac
2	1.235	aaaa::212:7406:6:606	aaaa::ff:fe00:1	UDP	85	Source port: ultraseek-http Destination port: rrac
3	1.236	aaaa::212:7406:6:606	aaaa::ff:fe00:1	UDP	85	Source port: ultraseek-http Destination port: rrac
4	1.236	aaaa::212:7406:6:606	aaaa::ff:fe00:1	UDP	85	Source port: ultraseek-http Destination port: rrac
5	1.237	aaaa::212:7406:6:606	aaaa::ff:fe00:1	UDP	85	Source port: ultraseek-http Destination port: rrac
6	1.238	aaaa::212:7406:6:606	aaaa::ff:fe00:1	UDP	85	Source port: ultraseek-http Destination port: rrac
7	1.239	aaaa::212:7406:6:606	aaaa::ff:fe00:1	UDP	85	Source port: ultraseek-http Destination port: rrac
8	1.24	aaaa::212:7406:6:606	aaaa::ff:fe00:1	UDP	85	Source port: ultraseek-http Destination port: rrac
9	1.24	aaaa::212:7406:6:606	aaaa::ff:fe00:1	UDP	85	Source port: ultraseek-http Destination port: rrac
10	1.241	aaaa::212:7406:6:606	aaaa::ff:fe00:1	UDP	85	Source port: ultraseek-http Destination port: rrac
11	1.242	aaaa::212:7406:6:606	aaaa::ff:fe00:1	UDP	85	Source port: ultraseek-http Destination port: rrac
12	1.242	aaaa::212:7406:6:606	aaaa::ff:fe00:1	UDP	85	Source port: ultraseek-http Destination port: rrac
13	1.243	aaaa::212:7406:6:606	aaaa::ff:fe00:1	UDP	85	Source port: ultraseek-http Destination port: rrac
14	1.243	aaaa::212:7406:6:606	aaaa::ff:fe00:1	UDP	85	Source port: ultraseek-http Destination port: rrac
15	1.244	aaaa::212:7406:6:606	aaaa::ff:fe00:1	UDP	85	Source port: ultraseek-http Destination port: rrac
16	1.245	aaaa::212:7406:6:606	aaaa::ff:fe00:1	UDP	85	Source port: ultraseek-http Destination port: rrac
17	1.245	aaaa::212:7406:6:606	aaaa::ff:fe00:1	UDP	85	Source port: ultraseek-http Destination port: rrac
18	1.246	aaaa::212:7406:6:606	aaaa::ff:fe00:1	UDP	85	Source port: ultraseek-http Destination port: rrac
19	1.246	aaaa::212:7406:6:606	aaaa::ff:fe00:1	UDP	85	Source port: ultraseek-http Destination port: rrac
20	1.247	aaaa::212:7406:6:606	aaaa::ff:fe00:1	UDP	85	Source port: ultraseek-http Destination port: rrac
21	1.248	aaaa::212:7406:6:606	aaaa::ff:fe00:1	UDP	85	Source port: ultraseek-http Destination port: rrac
22	1.248	aaaa::212:7406:6:606	aaaa::ff:fe00:1	UDP	85	Source port: ultraseek-http Destination port: rrac
23	1.249	aaaa::212:7406:6:606	aaaa::ff:fe00:1	UDP	85	Source port: ultraseek-http Destination port: rrac
24	1.25	aaaa::212:7406:6:606	aaaa::ff:fe00:1	UDP	85	Source port: ultraseek-http Destination port: rrac
25	1.25	aaaa::212:7406:6:606	aaaa::ff:fe00:1	UDP	85	Source port: ultraseek-http Destination port: rrac

Figure 4.13 Malicious “udp-flood-radiologUDP.csv”—1 to 25 records.

4.3 Blackhole Attack Datasets

In this Section, we provide a detailed description of the approach followed to generate a set of malicious datasets by implementing a blackhole attack scenario in the Cooja simulator, as shown in Figure 4.14.

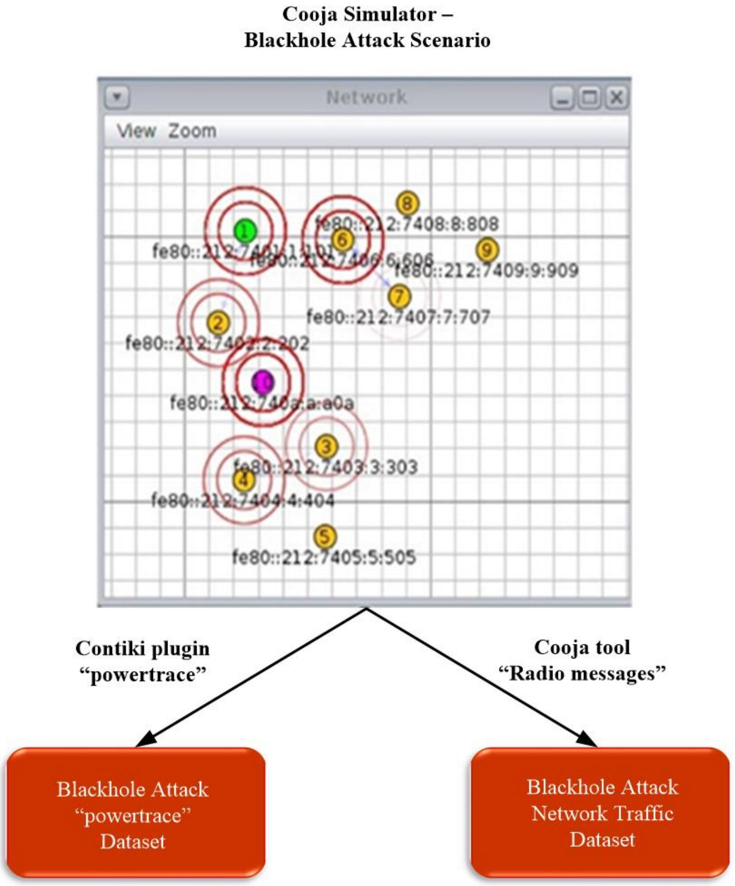
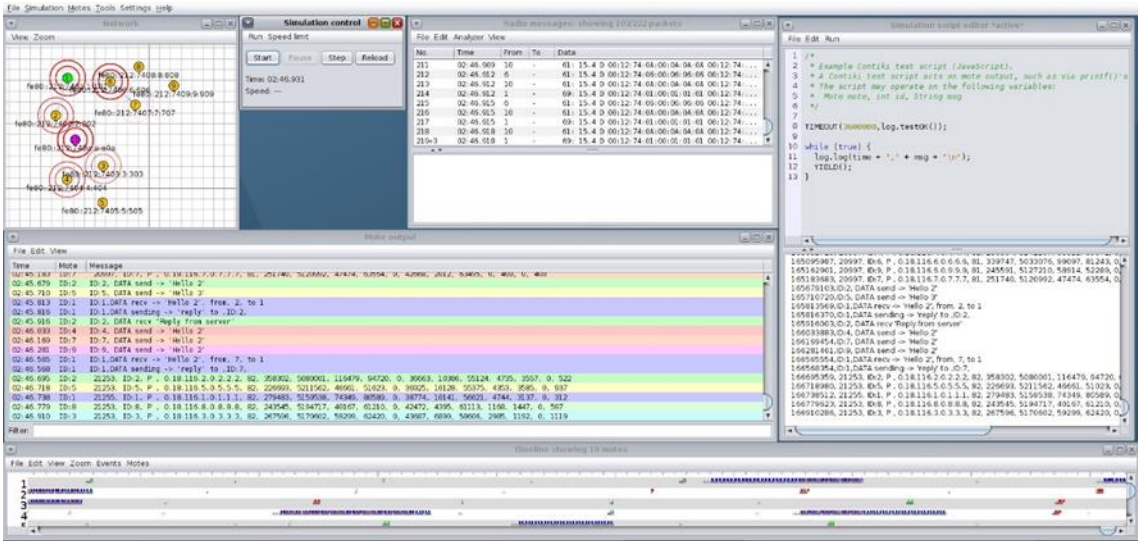


Figure 4.14 Blackhole Attack Datasets generation by utilising the Cooja simulator.

4.3.1 Blackhole Attack Scenario – an example

The network topology of the simulated blackhole attack scenario in the Cooja simulator environment consists of 8 yellow (benign) UDP-client motes (i.e., motes 2, 3, 4, 5, 6, 7, 8 and 9), the violet (malicious) UDP-client mote (i.e., mote 10) and the green (benign) UDP-server mote (i.e., mote 1), as depicted in Figure 4.14. The simulation duration was set to 60 mins and the motes' outputs were printed out in the respective window (e.g., Mote output) while simulations run, as shown in Figure 4.15. Moreover, the 8 yellow (benign) UDP-client motes were configured to send text messages every 30 seconds, approximately, to the UDP-server mote that was configured to provide a corresponding response. On the other hand, the violet (malicious) UDP-client mote (i.e., mote 10) was compromised with malicious code, as shown in Figure , to switch off transmission and disrupt the communication chain. The (malicious) mote was programmed to start as a normal mote and after 25 minutes later to switch off the radio, leading to a blackhole attack. Finally, it is noteworthy to say that similar to the benign network scenario, the UDP protocol was used at the Transport Layer, the IPv6 at the network layer, and the type of motes was the Tmote Sky in the blackhole attack scenario.



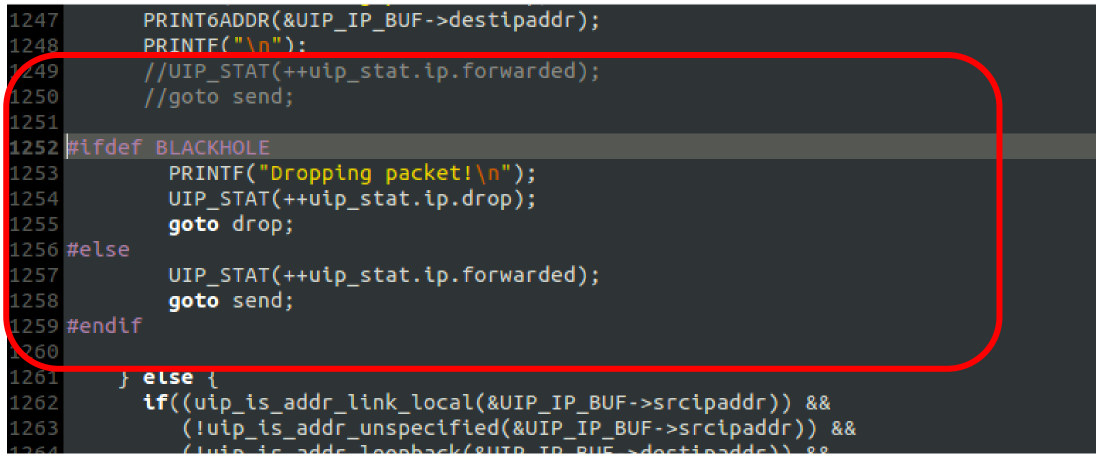
No.	Time	From	To	Data
211	02:46.909	10	-	61: 15.4 D 00:12:74:0A:00:0A:0A:0A 00:12:74:...
212	02:46.912	6	-	61: 15.4 D 00:12:74:06:00:06:06:06 00:12:74:...
213	02:46.912	10	-	61: 15.4 D 00:12:74:0A:00:0A:0A:0A 00:12:74:...
214	02:46.912	1	-	69: 15.4 D 00:12:74:01:00:01:01:01 00:12:74:...
215	02:46.915	6	-	61: 15.4 D 00:12:74:06:00:06:06:06 00:12:74:...
216	02:46.915	10	-	61: 15.4 D 00:12:74:0A:00:0A:0A:0A 00:12:74:...
217	02:46.915	1	-	69: 15.4 D 00:12:74:01:00:01:01:01 00:12:74:...
218	02:46.918	10	-	61: 15.4 D 00:12:74:0A:00:0A:0A:0A 00:12:74:...
219+3	02:46.918	1	-	69: 15.4 D 00:12:74:01:00:01:01:01 00:12:74:...

```
/*
 * Example Contiki test script (JavaScript).
 * A Contiki test script acts on mote output, such as via printf()'s
 * The script may operate on the following variables:
 *  Mote mote, int id, String msg
 */

TIMEOUT(3600000,log.testOK());

while (true) {
  log.log(time + ":" + msg + "\n");
  YIELD();
}
```

```
Time       Mote   Message
02:45.679  ID:2   ID:2, DATA send -> 'Hello 2'
02:45.710  ID:5   ID:5, DATA send -> 'Hello 3'
02:45.813  ID:1   ID:1,DATA recv -> 'Hello 2', from, 2, to 1
02:45.816  ID:1   ID:1,DATA sending -> 'reply' to ,ID:2,
02:45.916  ID:2   ID:2, DATA recv 'Reply from server'
02:46.033  ID:4   ID:4, DATA send -> 'Hello 2'
02:46.169  ID:7   ID:7, DATA send -> 'Hello 2'
02:46.281  ID:9   ID:9, DATA send -> 'Hello 2'
02:46.565  ID:1   ID:1,DATA recv -> 'Hello 2', from, 7, to 1
02:46.568  ID:1   ID:1,DATA sending -> 'reply' to ,ID:7,
02:46.695  ID:2    21253, ID:2, P , 0.18.116.2.0.2.2.2, 82, 358302, 5080001, 116479, 94720, 0, 36663, 10386, 55124, 4735, 3557, 0, 522
02:46.718  ID:5    21253, ID:5, P , 0.18.116.5.0.5.5.5, 82, 226693, 5211562, 46661, 51023, 0, 36925, 10128, 55375, 4353, 3585, 0, 937
02:46.738  ID:1    21255, ID:1, P , 0.18.116.1.0.1.1.1, 82, 279483, 5159538, 74349, 80589, 0, 38774, 10141, 56621, 4744, 3137, 0, 312
02:46.779  ID:8    21253, ID:8, P , 0.18.116.8.0.8.8.8, 82, 243545, 5194717, 40167, 61210, 0, 42472, 4395, 61113, 1168, 1447, 0, 587
02:46.910  ID:3    21253, ID:3, P , 0.18.116.3.0.3.3.3, 82, 267596, 5170602, 59299, 62420, 0, 43687, 6899, 50606, 2985, 1192, 0, 1119
```

```
165095907, 20997, ID:6, P , 0.18.116.6.0.6.6.6, 81, 339747, 5033076, 99097, 81243, 0,
165162901, 20997, ID:9, P , 0.18.116.9.0.9.9.9, 81, 245591, 5127210, 58914, 52289, 0,
165183683, 20997, ID:7, P , 0.18.116.7.0.7.7.7, 81, 251746, 5120992, 47474, 63554, 0,
165679103,ID:2, DATA send -> 'Hello 2'
165710720,ID:5, DATA send -> 'Hello 3'
165813569,ID:1,DATA recv -> 'Hello 2', from, 2, to 1
165816370,ID:1,DATA sending -> 'reply' to ,ID:2,
165916003,ID:2, DATA recv 'Reply from server'
166033883,ID:4, DATA send -> 'Hello 2'
166169454,ID:7, DATA send -> 'Hello 2'
166281461,ID:9, DATA send -> 'Hello 2'
166565554,ID:1,DATA recv -> 'Hello 2', from, 7, to 1
166568354,ID:1,DATA sending -> 'reply' to ,ID:7,
166695359, 21253, ID:2, P , 0.18.116.2.0.2.2.2, 82, 358302, 5080001, 116479, 94720,
166718980, 21253, ID:5, P , 0.18.116.5.0.5.5.5, 82, 226693, 5211562, 46661, 51023, 0,
166738512, 21255, ID:1, P , 0.18.116.1.0.1.1.1, 82, 279483, 5159538, 74349, 80589, 0,
166779623, 21253, ID:8, P , 0.18.116.8.0.8.8.8, 82, 243545, 5194717, 40167, 61210, 0,
166910286, 21253, ID:3, P , 0.18.116.3.0.3.3.3, 82, 267596, 5170602, 59299, 62420, 0,
```

Figure 4.15 Cooja Simulator – Blackhole attack scenario – Motes' outputs.



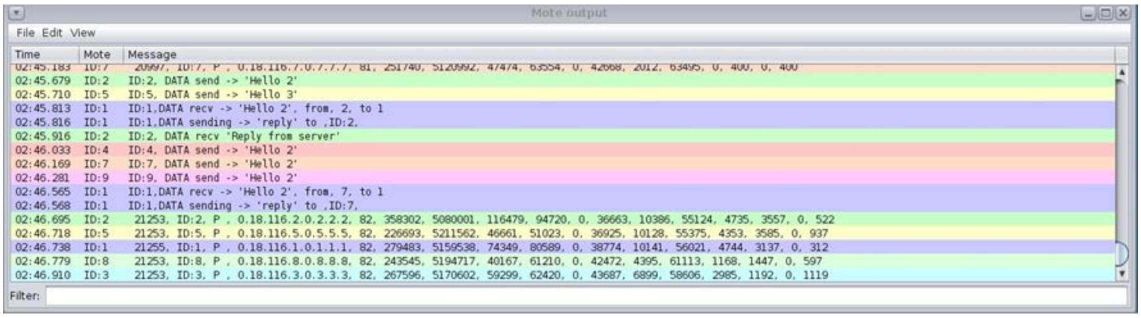
```
   PRINT6ADDR(&UIP_IP_BUF->destipaddr);
   PRINTF("\n");
   //UIP_STAT(++uip_stat.ip.forwarded);
   //goto send;

#ifdef BLACKHOLE
    PRINTF("Dropping packet!\n");
    UIP_STAT(++uip_stat.ip.drop);
    goto drop;
#else
    UIP_STAT(++uip_stat.ip.forwarded);
    goto send;
#endif

  } else {
   if((uip_is_addr_link_local(&UIP_IP_BUF->srcipaddr)) &&
     (!uip_is_addr_unspecified(&UIP_IP_BUF->srcipaddr)) &&
     (!uip_is_addr_loopback(&UIP_IP_BUF->destipaddr)) &&
```

Figure 4.16 Malicious code in "contiki/core/net/ipv6/uip6.c" to cause a blackhole attack by dropping all packets that are to be forwarded.

4.3.2 Blackhole Attack “powertrace” Dataset

4.3.2.1 Blackhole Attack “powetrace” Dataset – Generation Process

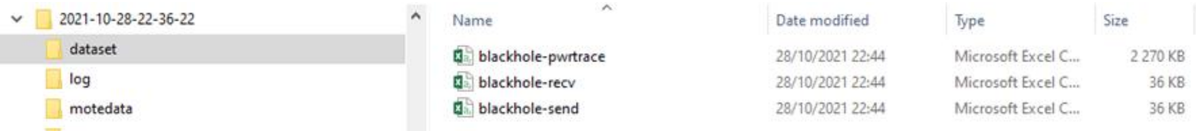
The approach followed for the “powertrace” dataset generation from the blackhole attack scenario was similar to the approach followed for the “powertrace” dataset generation from the benign IoT network scenario in Section 3.3.1. In addition, the “powertrace” plugin was similarly enabled for collecting “powertrace” related features, summarised in Table 3, from the motes of the attack scenario every two seconds. In Figure 4.17, the depicted mote output window displays the captured “powertrace” information every two seconds and also the messages sent and received by each mote during the simulation time (60 mins).



```
Time       Mote   Message
02:45.183  ID:7    20997, ID:7, P , 0.18.116.7.0.7.7.7, 81, 251740, 5120992, 47474, 63554, 0, 42668, 2012, 63495, 0, 400, 0, 400
02:45.679  ID:2   ID:2, DATA send -> 'Hello 2'
02:45.710  ID:5   ID:5, DATA send -> 'Hello 3'
02:45.813  ID:1   ID:1,DATA recv -> 'Hello 2', from, 2, to 1
02:45.816  ID:1   ID:1,DATA sending -> 'reply' to ,ID:2,
02:45.916  ID:2   ID:2, DATA recv 'Reply from server'
02:46.033  ID:4   ID:4, DATA send -> 'Hello 2'
02:46.169  ID:7   ID:7, DATA send -> 'Hello 2'
02:46.281  ID:9   ID:9, DATA send -> 'Hello 2'
02:46.565  ID:1   ID:1,DATA recv -> 'Hello 2', from, 7, to 1
02:46.568  ID:1   ID:1,DATA sending -> 'reply' to ,ID:7,
02:46.695  ID:2    21253, ID:2, P , 0.18.116.2.0.2.2.2, 82, 358302, 5080001, 116479, 94720, 0, 36663, 10386, 55124, 4735, 3557, 0, 522
02:46.718  ID:5    21253, ID:5, P , 0.18.116.5.0.5.5.5, 82, 226693, 5211562, 46661, 51023, 0, 36925, 10128, 55375, 4353, 3585, 0, 937
02:46.738  ID:1    21255, ID:1, P , 0.18.116.1.0.1.1.1, 82, 279483, 5159538, 74349, 80589, 0, 38774, 10141, 56021, 4744, 3137, 0, 312
02:46.779  ID:8    21253, ID:8, P , 0.18.116.8.0.8.8.8, 82, 243545, 5194717, 40167, 61210, 0, 42472, 4395, 61113, 1168, 1447, 0, 597
02:46.910  ID:3    21253, ID:3, P , 0.18.116.3.0.3.3.3, 82, 267596, 5170602, 59299, 62420, 0, 43687, 6899, 58606, 2985, 1192, 0, 1119
```

Figure 4.17 Cooja Simulator – Blackhole attack scenario – Mote output window

When the timeout occurred, the simulation stopped, and all the captured information and prints were stored in the “COOJA.testlog” file. Afterwards, the “IoT_Simul.sh” file, described in Section 3.3.1, created a) a new root folder named as “2021-10-28-22-36-22”, and b) the “log” folder, inside the “2021-10-28-22-36-22” folder, where the “COOJA.testlog” file was copied from the “…/cooja/build” folder located in the Cooja Simulator. Then, the “IoT_Simul.sh” file following the same process, as described in Section 3.3.1, extracted the required “powertrace” information from the “COOJA.testlog” file and saved it in the “blackhole-pwrtrace.csv” file in the “dataset” folder that was also created by the batch file inside the “2021-10-28-22-36-22” folder, as shown below in the left part of Figure 4.18. In the “dataset” folder, apart from the “blackhole-pwrtrace.csv” file, the “IoT_Simul.sh” file generated two more files (i.e., “blackhole-recv.csv” and “blackhole-send.csv”), following the same process as in Section 3.3.1. The “blackhole-recv.csv” file and the “blackhole-send.csv” file include the “received” and “sent” messages printed by the motes, respectively.



2021-10-28-22-36-22
- dataset
- log
- motedata

Name	Date modified	Type	Size
blackhole-pwrtrace	28/10/2021 22:44	Microsoft Excel C...	2 270 KB
blackhole-recv	28/10/2021 22:44	Microsoft Excel C...	36 KB
blackhole-send	28/10/2021 22:44	Microsoft Excel C...	36 KB

Figure 4.18 Location of the generated “blackhole-pwrtrace.csv”, “blackhole-recv.csv”, and “blackhole-send.csv” files by the “IoT_Simul.sh” bash file.

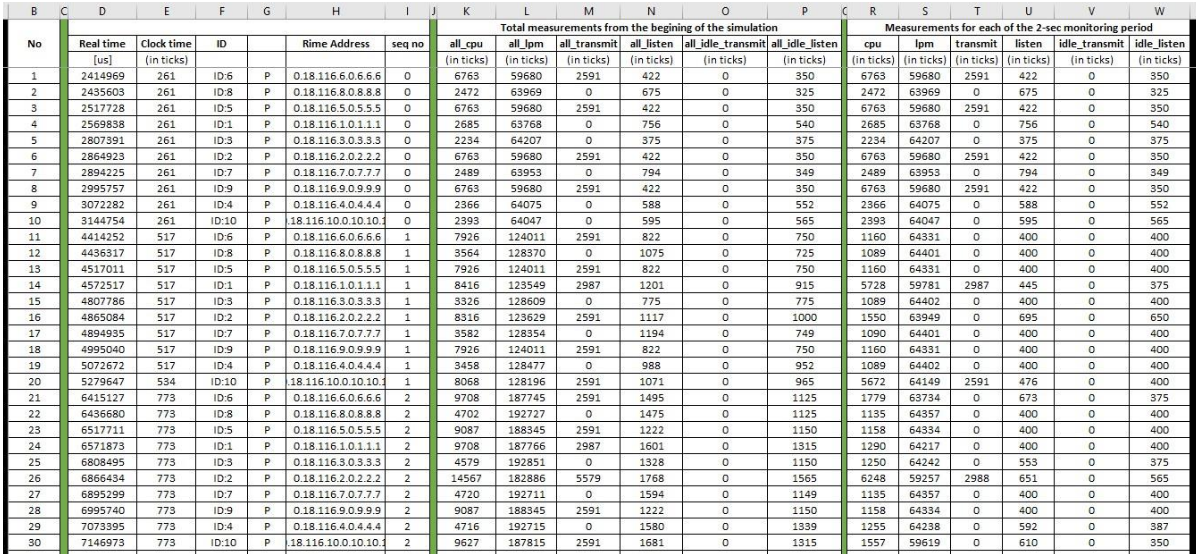
Finally, similar to the benign “powertrace” dataset generation approach in Section 3.3.1, the “IoT_Simul.sh” file extracted the information related to each mote from the “blackhole-pwrtrace.csv” file and generated one csv file for each mote with the corresponding information from the “blackhole-pwrtrace.csv” file. The generated ten csv files (i.e., “blackhole-mote1.csv”,…, “blackhole-mote10.csv”) were stored in the “motedata” folder, created also by the “IoT_Simul.sh” file, as shown in the left part of Figure 4.18.

4.3.2.2 Blackhole Attack "powetrace" Dataset – Generated Results

The blackhole attack "powertrace" dataset consists of the following csv files: "blackhole-pwrtrace.csv", "blackhole-mote1.csv" "blackhole-mote2.csv" "blackhole-mote3.csv" "blackhole-mote4.csv" "blackhole-mote5.csv" "blackhole-mote6.csv" "blackhole-mote7.csv" "blackhole-mote8.csv" "blackhole-mote9.csv" and "blackhole-mote10.csv". In this Section, we present sets of records from the "blackhole-pwrtrace.csv", and in Appendix 1 we present sets of records from "blackhole-mote1.csv", "blackhole-mote4.csv" and "blackhole-mote10.csv" files.

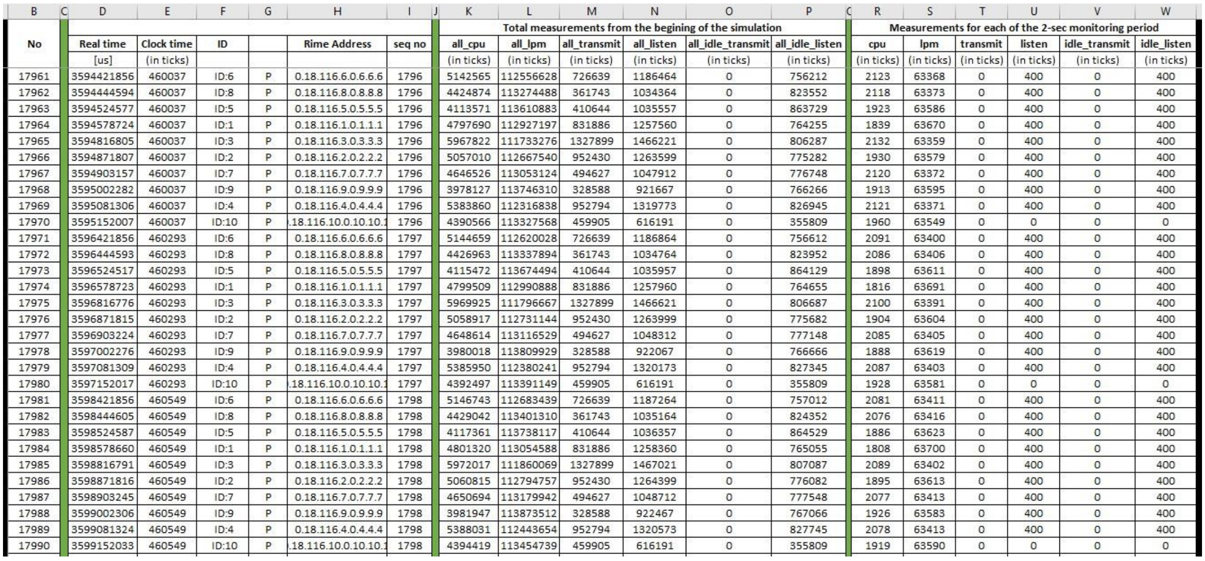
4.3.2.2.1 "blackhole-pwrtrace.csv"

The generated malicious "blackhole-pwrtrace.csv" file consists of 17,990 records and its first 30 records (i.e., 1–30) and its last 30 records (17,961–17,990) are depicted in Figure 4.19 and Figure 4.20, respectively.



B	D	E	F	G	H	I	K	L	M	N	O	P	R	S	T	U	V	W
No							Total measurements from the begining of the simulation						Measurements for each of the 2-sec monitoring period					
	Real time	Clock time	ID		Rime Address	seq no	all_cpu	all_lpm	all_transmit	all_listen	all_idle_transmit	all_idle_listen	cpu	lpm	transmit	listen	idle_transmit	idle_listen
	[us]	(in ticks)					(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)
1	2414969	261	ID:6	P	0.18.116.6.0.6.6.6	0	6763	59680	2591	422	0	350	6763	59680	2591	422	0	350
2	2435603	261	ID:8	P	0.18.116.8.0.8.8.8	0	2472	63969	0	675	0	325	2472	63969	0	675	0	325
3	2517728	261	ID:5	P	0.18.116.5.0.5.5.5	0	6763	59680	2591	422	0	350	6763	59680	2591	422	0	350
4	2569838	261	ID:1	P	0.18.116.1.0.1.1.1	0	2685	63768	0	756	0	540	2685	63768	0	756	0	540
5	2807391	261	ID:3	P	0.18.116.3.0.3.3.3	0	2234	64207	0	375	0	375	2234	64207	0	375	0	375
6	2864923	261	ID:2	P	0.18.116.2.0.2.2.2	0	6763	59680	2591	422	0	350	6763	59680	2591	422	0	350
7	2894225	261	ID:7	P	0.18.116.7.0.7.7.7	0	2489	63953	0	794	0	349	2489	63953	0	794	0	349
8	2995757	261	ID:9	P	0.18.116.9.0.9.9.9	0	6763	59680	2591	422	0	350	6763	59680	2591	422	0	350
9	3072282	261	ID:4	P	0.18.116.4.0.4.4.4	0	2366	64075	0	588	0	552	2366	64075	0	588	0	552
10	3144754	261	ID:10	P	.18.116.10.0.10.10.1	0	2393	64047	0	595	0	565	2393	64047	0	595	0	565
11	4414252	517	ID:6	P	0.18.116.6.0.6.6.6	1	7926	124011	2591	822	0	750	1160	64331	0	400	0	400
12	4436317	517	ID:8	P	0.18.116.8.0.8.8.8	1	3564	128370	0	1075	0	725	1089	64401	0	400	0	400
13	4517011	517	ID:5	P	0.18.116.5.0.5.5.5	1	7926	124011	2591	822	0	750	1160	64331	0	400	0	400
14	4572517	517	ID:1	P	0.18.116.1.0.1.1.1	1	8416	123549	2987	1201	0	915	5728	59781	2987	445	0	375
15	4807786	517	ID:3	P	0.18.116.3.0.3.3.3	1	3326	128609	0	775	0	775	1089	64402	0	400	0	400
16	4865084	517	ID:2	P	0.18.116.2.0.2.2.2	1	8316	123629	2591	1117	0	1000	1550	63949	0	695	0	650
17	4894935	517	ID:7	P	0.18.116.7.0.7.7.7	1	3582	128354	0	1194	0	749	1090	64401	0	400	0	400
18	4995040	517	ID:9	P	0.18.116.9.0.9.9.9	1	7926	124011	2591	822	0	750	1160	64331	0	400	0	400
19	5072672	517	ID:4	P	0.18.116.4.0.4.4.4	1	3458	128477	0	988	0	952	1089	64402	0	400	0	400
20	5279647	534	ID:10	P	.18.116.10.0.10.10.1	1	8068	128196	2591	1071	0	965	5672	64149	2591	476	0	400
21	6415127	773	ID:6	P	0.18.116.6.0.6.6.6	2	9708	187745	2591	1495	0	1125	1779	63734	0	673	0	375
22	6436680	773	ID:8	P	0.18.116.8.0.8.8.8	2	4702	192727	0	1475	0	1125	1135	64357	0	400	0	400
23	6517711	773	ID:5	P	0.18.116.5.0.5.5.5	2	9087	188345	2591	1222	0	1150	1158	64334	0	400	0	400
24	6571873	773	ID:1	P	0.18.116.1.0.1.1.1	2	9708	187766	2987	1601	0	1315	1290	64217	0	400	0	400
25	6808495	773	ID:3	P	0.18.116.3.0.3.3.3	2	4579	192851	0	1328	0	1150	1250	64242	0	553	0	375
26	6866434	773	ID:2	P	0.18.116.2.0.2.2.2	2	14567	182886	5579	1768	0	1565	6248	59257	2988	651	0	565
27	6895299	773	ID:7	P	0.18.116.7.0.7.7.7	2	4720	192711	0	1594	0	1149	1135	64357	0	400	0	400
28	6995740	773	ID:9	P	0.18.116.9.0.9.9.9	2	9087	188345	2591	1222	0	1150	1158	64334	0	400	0	400
29	7073395	773	ID:4	P	0.18.116.4.0.4.4.4	2	4716	192715	0	1580	0	1339	1255	64238	0	592	0	387
30	7146973	773	ID:10	P	.18.116.10.0.10.10.1	2	9627	187815	2591	1681	0	1315	1557	59619	0	610	0	350

Figure 4.19 Malicious "blackhole-pwrtrace.csv" — 1 to 30 records.



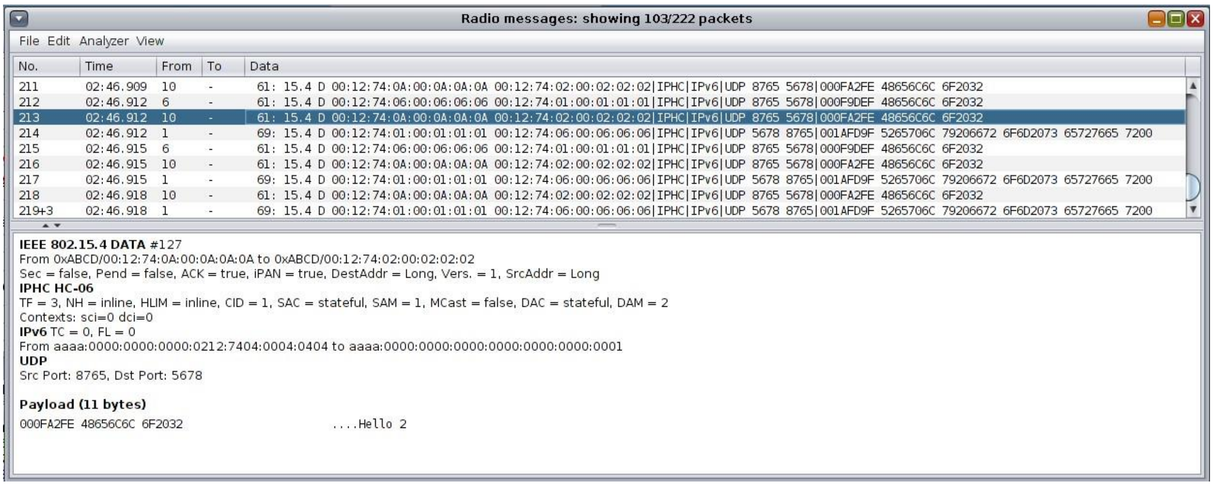
B	D	E	F	G	H	I	K	L	M	N	O	P	R	S	T	U	V	W
No							Total measurements from the begining of the simulation						Measurements for each of the 2-sec monitoring period					
	Real time	Clock time	ID		Rime Address	seq no	all_cpu	all_lpm	all_transmit	all_listen	all_idle_transmit	all_idle_listen	cpu	lpm	transmit	listen	idle_transmit	idle_listen
	[us]	(in ticks)					(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)
17961	3594421856	460037	ID:6	P	0.18.116.6.0.6.6.6	1796	5142565	112556628	726639	1186464	0	756212	2123	63368	0	400	0	400
17962	3594444594	460037	ID:8	P	0.18.116.8.0.8.8.8	1796	4424874	113274488	361743	1034364	0	823552	2118	63373	0	400	0	400
17963	3594524577	460037	ID:5	P	0.18.116.5.0.5.5.5	1796	4113571	113610883	410644	1035557	0	863729	1923	63586	0	400	0	400
17964	3594578724	460037	ID:1	P	0.18.116.1.0.1.1.1	1796	4797690	112927197	831886	1257560	0	764255	1839	63670	0	400	0	400
17965	3594816805	460037	ID:3	P	0.18.116.3.0.3.3.3	1796	5967822	111733276	1327899	1466221	0	806287	2132	63359	0	400	0	400
17966	3594871807	460037	ID:2	P	0.18.116.2.0.2.2.2	1796	5057010	112667540	952430	1263599	0	775282	1930	63579	0	400	0	400
17967	3594903157	460037	ID:7	P	0.18.116.7.0.7.7.7	1796	4646526	113053124	494627	1047912	0	776748	2120	63372	0	400	0	400
17968	3595002282	460037	ID:9	P	0.18.116.9.0.9.9.9	1796	3978127	113746310	328588	921667	0	766266	1913	63595	0	400	0	400
17969	3595081306	460037	ID:4	P	0.18.116.4.0.4.4.4	1796	5383860	112316838	952794	1319773	0	826945	2121	63371	0	400	0	400
17970	3595152007	460037	ID:10	P	.18.116.10.0.10.10.1	1796	4390566	113327568	459905	616191	0	355809	1960	63549	0	0	0	0
17971	3596421856	460293	ID:6	P	0.18.116.6.0.6.6.6	1797	5144659	112620028	726639	1186864	0	756612	2091	63400	0	400	0	400
17972	3596444593	460293	ID:8	P	0.18.116.8.0.8.8.8	1797	4426963	113337894	361743	1034764	0	823952	2086	63406	0	400	0	400
17973	3596524517	460293	ID:5	P	0.18.116.5.0.5.5.5	1797	4115472	113674494	410644	1035957	0	864129	1898	63611	0	400	0	400
17974	3596578723	460293	ID:1	P	0.18.116.1.0.1.1.1	1797	4799509	112990888	831886	1257960	0	764655	1816	63691	0	400	0	400
17975	3596816776	460293	ID:3	P	0.18.116.3.0.3.3.3	1797	5969925	111796667	1327899	1466621	0	806687	2100	63391	0	400	0	400
17976	3596871815	460293	ID:2	P	0.18.116.2.0.2.2.2	1797	5058917	112731144	952430	1263999	0	775682	1904	63604	0	400	0	400
17977	3596903224	460293	ID:7	P	0.18.116.7.0.7.7.7	1797	4648614	113116529	494627	1048312	0	777148	2085	63405	0	400	0	400
17978	3597002276	460293	ID:9	P	0.18.116.9.0.9.9.9	1797	3980018	113809929	328588	922067	0	766666	1888	63619	0	400	0	400
17979	3597081309	460293	ID:4	P	0.18.116.4.0.4.4.4	1797	5385950	112380241	952794	1320173	0	827345	2087	63403	0	400	0	400
17980	3597152017	460293	ID:10	P	.18.116.10.0.10.10.1	1797	4392497	113391149	459905	616191	0	355809	1928	63581	0	0	0	0
17981	3598421856	460549	ID:6	P	0.18.116.6.0.6.6.6	1798	5146743	112683439	726639	1187264	0	757012	2081	63411	0	400	0	400
17982	3598444605	460549	ID:8	P	0.18.116.8.0.8.8.8	1798	4429042	113401310	361743	1035164	0	824352	2076	63416	0	400	0	400
17983	3598524587	460549	ID:5	P	0.18.116.5.0.5.5.5	1798	4117361	113738117	410644	1036357	0	864529	1886	63623	0	400	0	400
17984	3598578660	460549	ID:1	P	0.18.116.1.0.1.1.1	1798	4801320	113054588	831886	1258360	0	765055	1808	63700	0	400	0	400
17985	3598816791	460549	ID:3	P	0.18.116.3.0.3.3.3	1798	5972017	111860069	1327899	1467021	0	807087	2089	63402	0	400	0	400
17986	3598871816	460549	ID:2	P	0.18.116.2.0.2.2.2	1798	5060815	112794757	952430	1264399	0	776082	1895	63613	0	400	0	400
17987	3598903245	460549	ID:7	P	0.18.116.7.0.7.7.7	1798	4650694	113179942	494627	1048712	0	777548	2077	63413	0	400	0	400
17988	3599002306	460549	ID:9	P	0.18.116.9.0.9.9.9	1798	3981947	113873512	328588	922467	0	767066	1926	63583	0	400	0	400
17989	3599081324	460549	ID:4	P	0.18.116.4.0.4.4.4	1798	5388031	112443654	952794	1320573	0	827745	2078	63413	0	400	0	400
17990	3599152033	460549	ID:10	P	.18.116.10.0.10.10.1	1798	4394419	113454739	459905	616191	0	355809	1919	63590	0	0	0	0

Figure 4.20 Malicious "blackhole-pwrtrace.csv"—17,961 to 17,990 records.

4.3.3 Blackhole Attack Network Traffic Dataset

4.3.3.1 Blackhole Attack Network Traffic Dataset – Generation Process

The approach followed for the network traffic dataset generation from the blackhole attack scenario was similar to the approach followed for the network traffic dataset generation from the benign IoT network scenario in Section 3.4.1. The “Radio messages” tool, provided by the Cooja simulator, was similarly used for collecting data related to the corresponding network traffic features (e.g., source/destination IPv6 address, packet size, and communication protocol) from the network of the attack scenario. During the simulation, the network traffic information was being shown in the top part of the “Radio messages” output window as depicted in the top part of Figure 4.21.

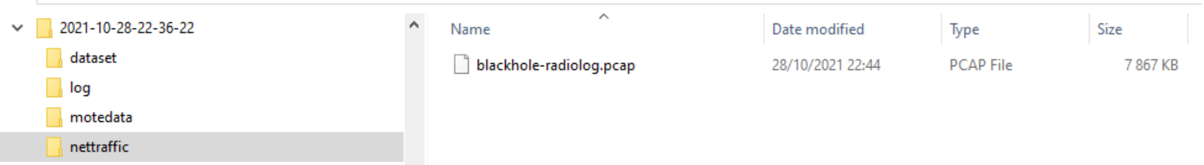


```
No.     Time       From  To   Data
211     02:46.909  10    -    61: 15.4 D 00:12:74:0A:00:0A:0A:0A 00:12:74:02:00:02:02:02|IPHC|IPv6|UDP 8765 5678|000FA2FE 48656C6C 6F2032
212     02:46.912  6     -    61: 15.4 D 00:12:74:06:00:06:06:06 00:12:74:01:00:01:01:01|IPHC|IPv6|UDP 8765 5678|000F9DEF 48656C6C 6F2032
213     02:46.912  10    -    61: 15.4 D 00:12:74:0A:00:0A:0A:0A 00:12:74:02:00:02:02:02|IPHC|IPv6|UDP 8765 5678|000FA2FE 48656C6C 6F2032
214     02:46.912  1     -    69: 15.4 D 00:12:74:01:00:01:01:01 00:12:74:06:00:06:06:06|IPHC|IPv6|UDP 5678 8765|001AFD9F 5265706C 79206672 6F6D2073 65727665 7200
215     02:46.915  6     -    61: 15.4 D 00:12:74:06:00:06:06:06 00:12:74:01:00:01:01:01|IPHC|IPv6|UDP 8765 5678|000F9DEF 48656C6C 6F2032
216     02:46.915  10    -    61: 15.4 D 00:12:74:0A:00:0A:0A:0A 00:12:74:02:00:02:02:02|IPHC|IPv6|UDP 8765 5678|000FA2FE 48656C6C 6F2032
217     02:46.915  1     -    69: 15.4 D 00:12:74:01:00:01:01:01 00:12:74:06:00:06:06:06|IPHC|IPv6|UDP 5678 8765|001AFD9F 5265706C 79206672 6F6D2073 65727665 7200
218     02:46.918  10    -    61: 15.4 D 00:12:74:0A:00:0A:0A:0A 00:12:74:02:00:02:02:02|IPHC|IPv6|UDP 8765 5678|000FA2FE 48656C6C 6F2032
219+3   02:46.918  1     -    69: 15.4 D 00:12:74:01:00:01:01:01 00:12:74:06:00:06:06:06|IPHC|IPv6|UDP 5678 8765|001AFD9F 5265706C 79206672 6F6D2073 65727665 7200
```

```
IEEE 802.15.4 DATA #127
From 0xABCD/00:12:74:0A:00:0A:0A:0A to 0xABCD/00:12:74:02:00:02:02:02
Sec = false, Pend = false, ACK = true, iPAN = true, DestAddr = Long, Vers. = 1, SrcAddr = Long
IPHC HC-06
TF = 3, NH = inline, HLIM = inline, CID = 1, SAC = stateful, SAM = 1, MCast = false, DAC = stateful, DAM = 2
Contexts: sci=0 dci=0
IPv6 TC = 0, FL = 0
From aaaa:0000:0000:0000:0212:7404:0004:0404 to aaaa:0000:0000:0000:0000:0000:0000:0001
UDP
Src Port: 8765, Dst Port: 5678

Payload (11 bytes)
000FA2FE 48656C6C 6F2032                    ....Hello 2
```

Figure 4.21 Network traffic information from the blackhole attack scenario in the “Radio messages” output window.

When the simulation stopped, the generated pcap file was saved as “radiolog.pcap” within the “…/cooja/build” folder. Afterwards, the “IoT_Simul.sh” file, described in Section 3.4.1, created a) a new root folder named as “2021-10-28-22-36-22”, and b) the “nettraffic” folder, inside the “2021-10-28-22-36-22” folder, where the “radiolog.pcap”, copied from the “…/cooja/build” folder located in the Cooja Simulator, was saved as “blackhole-radiolog.pcap”. The “nettraffic” folder inside the root folder “2021-10-28-22-36-22” and the “blackhole-radiolog.pcap” file in the “nettraffic” folder are shown in Figure 4.22.

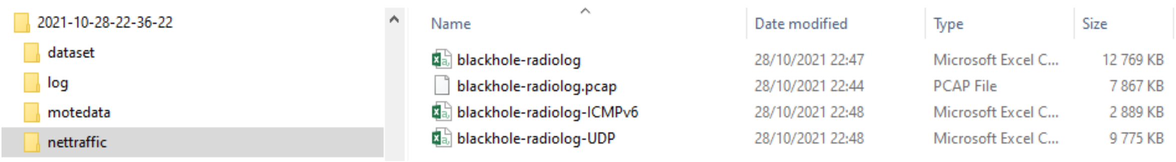


2021-10-28-22-36-22
dataset
log
motedata
nettraffic

Name	Date modified	Type	Size
blackhole-radiolog.pcap	28/10/2021 22:44	PCAP File	7 867 KB

Figure 4.22 The “nettraffic” folder inside the root folder “2021-10-28-22-36-22” and the “blackhole-radiolog.pcap” file.

Then, following the same process, as described in Section 3.4.1, we used Wireshark to extract the stored network traffic information from the “blackhole-radiolog.pcap” file to the “blackhole-radiolog.csv” file stored in the “nettraffic” folder as shown in Figure 4.23.



- 2021-10-28-22-36-22
  - dataset
  - log
  - motedata
  - nettraffic

Name	Date modified	Type	Size
blackhole-radiolog	28/10/2021 22:47	Microsoft Excel C...	12 769 KB
blackhole-radiolog.pcap	28/10/2021 22:44	PCAP File	7 867 KB
blackhole-radiolog-ICMPv6	28/10/2021 22:48	Microsoft Excel C...	2 889 KB
blackhole-radiolog-UDP	28/10/2021 22:48	Microsoft Excel C...	9 775 KB

Figure 4.23 The “nettraffic” folder inside the root folder “2021-10-28-22-36-22” and its included files.

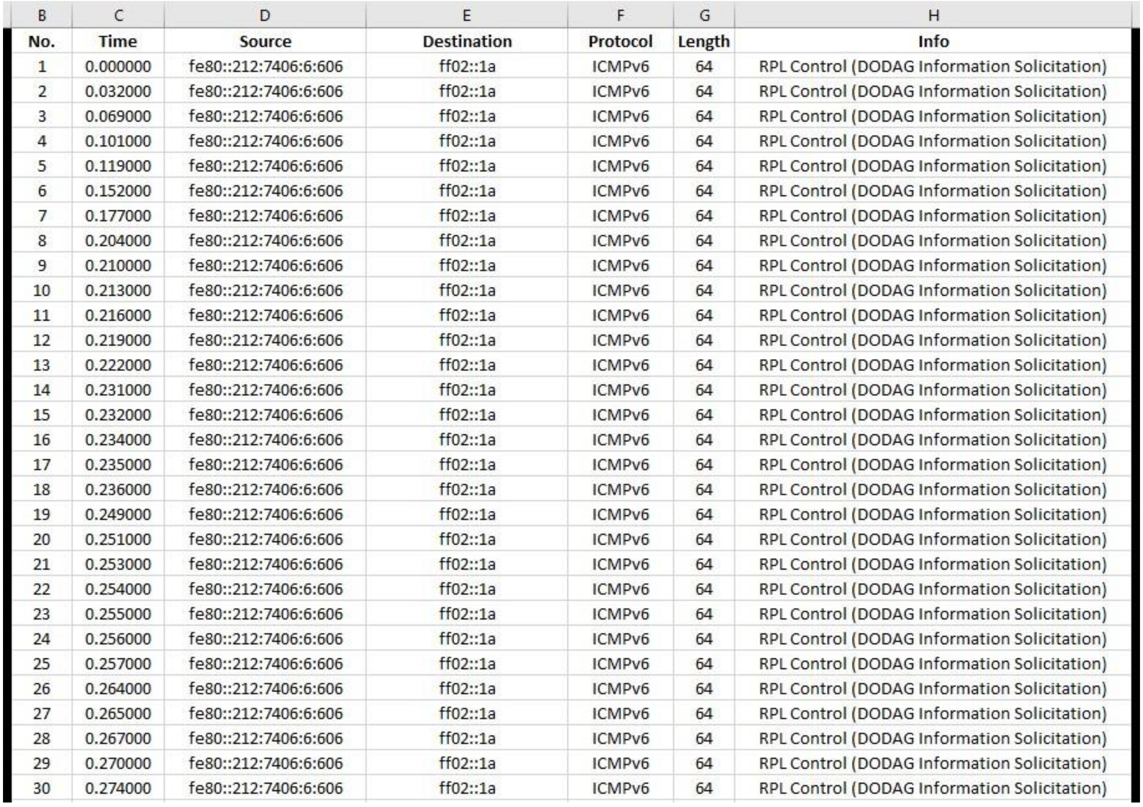
In the “nettraffic” folder, apart from the “blackhole-radiolog.csv” file, we also used Wireshark, following the same process as in Section 3.4.1, to generate two more files (i.e., “blackhole-radiolog-ICMPv6.csv” and “blackhole-radiolog-UDP.csv”) from the “blackhole-radiolog.pcap” file.

4.3.3.2 Blackhole Attack Network Traffic Dataset – Generated Results

The blackhole attack network traffic dataset consists of the following csv files which are located in the “nettraffic” folder as described in Section 4.3.3.1: “blackhole-radiolog.csv”, “blackhole-radiolog-ICMPv6.csv”, and “blackhole-radiolog-UDP.csv” files. In this Section, we present sets of records from these files.

4.3.3.2.1 “blackhole-radiolog.csv”

The generated malicious “blackhole-radiolog.csv” file consists of 99,622 records and its first 30 records (i.e., 1–30) are depicted below in Figure 4.24.

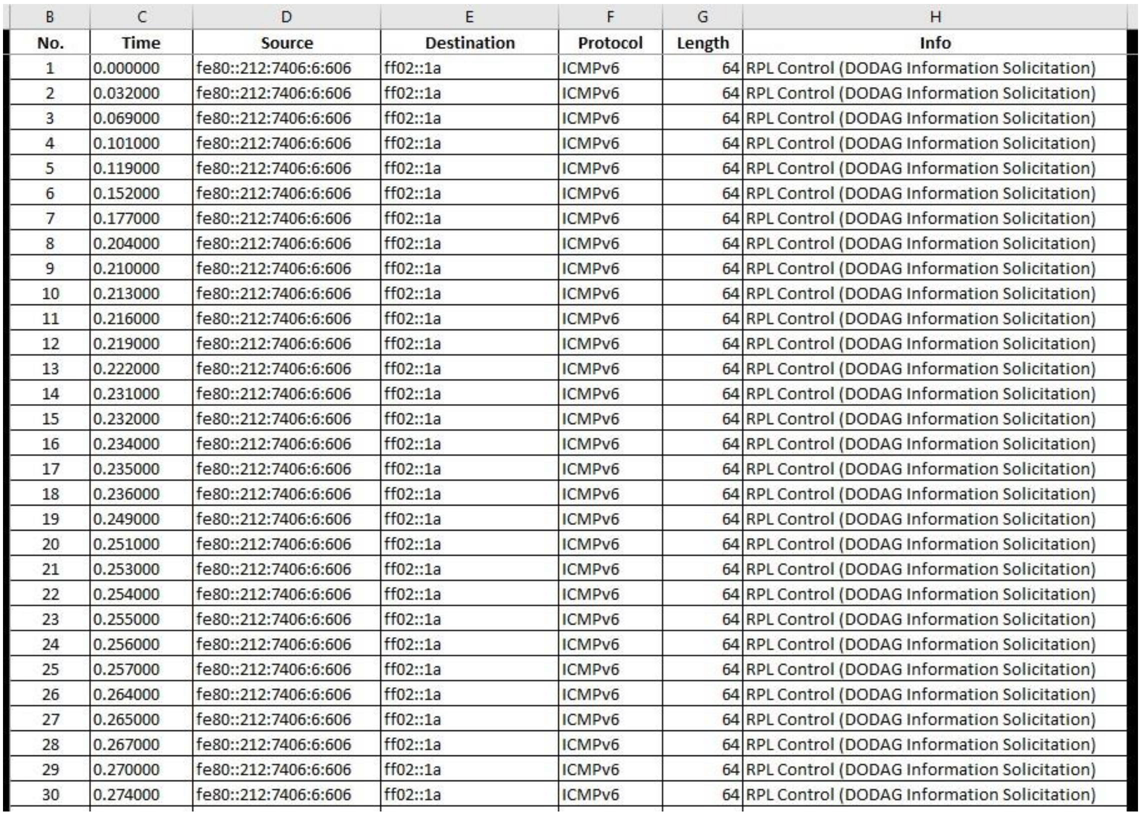


B	C	D	E	F	G	H
No.	**Time**	**Source**	**Destination**	**Protocol**	**Length**	**Info**
1	0.000000	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
2	0.032000	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
3	0.069000	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
4	0.101000	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
5	0.119000	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
6	0.152000	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
7	0.177000	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
8	0.204000	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
9	0.210000	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
10	0.213000	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
11	0.216000	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
12	0.219000	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
13	0.222000	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
14	0.231000	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
15	0.232000	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
16	0.234000	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
17	0.235000	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
18	0.236000	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
19	0.249000	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
20	0.251000	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
21	0.253000	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
22	0.254000	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
23	0.255000	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
24	0.256000	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
25	0.257000	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
26	0.264000	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
27	0.265000	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
28	0.267000	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
29	0.270000	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
30	0.274000	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)

Figure 4.24 Malicious “blackhole-radiolog.csv”—1 to 30 records.

4.3.3.2.2 “blackhole-radiolog-ICMPv6.csv”

The generated malicious “blackhole-radiolog-ICMPv6.csv” file consists of 24,011 records and its first 30 records (i.e., 1–30) are depicted below in Figure .4.25

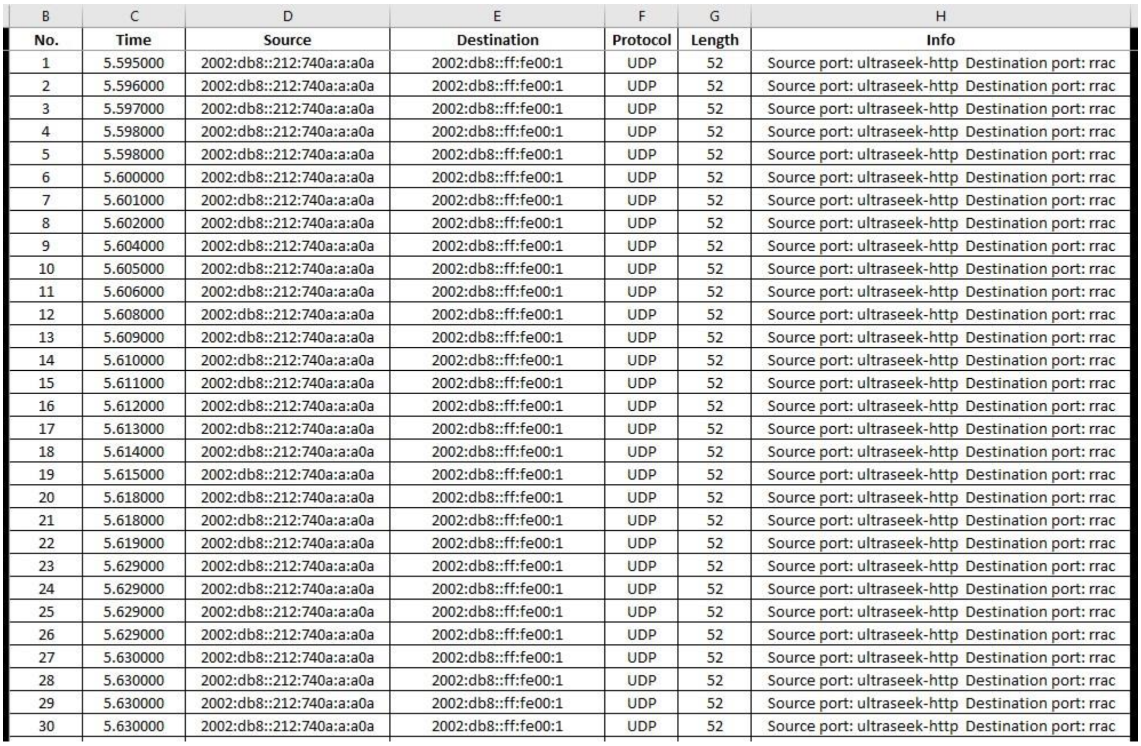


B	C	D	E	F	G	H
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
2	0.032000	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
3	0.069000	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
4	0.101000	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
5	0.119000	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
6	0.152000	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
7	0.177000	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
8	0.204000	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
9	0.210000	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
10	0.213000	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
11	0.216000	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
12	0.219000	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
13	0.222000	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
14	0.231000	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
15	0.232000	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
16	0.234000	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
17	0.235000	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
18	0.236000	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
19	0.249000	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
20	0.251000	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
21	0.253000	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
22	0.254000	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
23	0.255000	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
24	0.256000	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
25	0.257000	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
26	0.264000	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
27	0.265000	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
28	0.267000	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
29	0.270000	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
30	0.274000	fe80::212:7406:6:606	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)

Figure 4.25 Malicious "blackhole-radiolog-ICMPv6.csv"—1 to 30 records.

4.3.3.2.3 "blackhole-radiolog-UDP.csv"

The generated malicious "blackhole-radiolog-UDP.csv" file consists of 73,551 records and its first 30 records (i.e., 1–30) are depicted below in Figure 4.26.



B	C	D	E	F	G	H
No.	**Time**	**Source**	**Destination**	**Protocol**	**Length**	**Info**
1	5.595000	2002:db8::212:740a:a:a0a	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
2	5.596000	2002:db8::212:740a:a:a0a	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
3	5.597000	2002:db8::212:740a:a:a0a	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
4	5.598000	2002:db8::212:740a:a:a0a	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
5	5.598000	2002:db8::212:740a:a:a0a	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
6	5.600000	2002:db8::212:740a:a:a0a	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
7	5.601000	2002:db8::212:740a:a:a0a	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
8	5.602000	2002:db8::212:740a:a:a0a	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
9	5.604000	2002:db8::212:740a:a:a0a	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
10	5.605000	2002:db8::212:740a:a:a0a	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
11	5.606000	2002:db8::212:740a:a:a0a	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
12	5.608000	2002:db8::212:740a:a:a0a	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
13	5.609000	2002:db8::212:740a:a:a0a	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
14	5.610000	2002:db8::212:740a:a:a0a	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
15	5.611000	2002:db8::212:740a:a:a0a	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
16	5.612000	2002:db8::212:740a:a:a0a	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
17	5.613000	2002:db8::212:740a:a:a0a	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
18	5.614000	2002:db8::212:740a:a:a0a	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
19	5.615000	2002:db8::212:740a:a:a0a	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
20	5.618000	2002:db8::212:740a:a:a0a	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
21	5.618000	2002:db8::212:740a:a:a0a	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
22	5.619000	2002:db8::212:740a:a:a0a	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
23	5.629000	2002:db8::212:740a:a:a0a	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
24	5.629000	2002:db8::212:740a:a:a0a	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
25	5.629000	2002:db8::212:740a:a:a0a	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
26	5.629000	2002:db8::212:740a:a:a0a	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
27	5.630000	2002:db8::212:740a:a:a0a	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
28	5.630000	2002:db8::212:740a:a:a0a	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
29	5.630000	2002:db8::212:740a:a:a0a	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
30	5.630000	2002:db8::212:740a:a:a0a	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac

Figure 4.26 Malicious "blackhole-radiolog-UDP.csv"—1 to 30 records.

4.4 Sinkhole Attack Datasets

In this Section, we provide a detailed description of the approach followed to generate a set of malicious datasets by implementing a sinkhole attack scenario in the Cooja simulator, as shown in Figure 4.27.

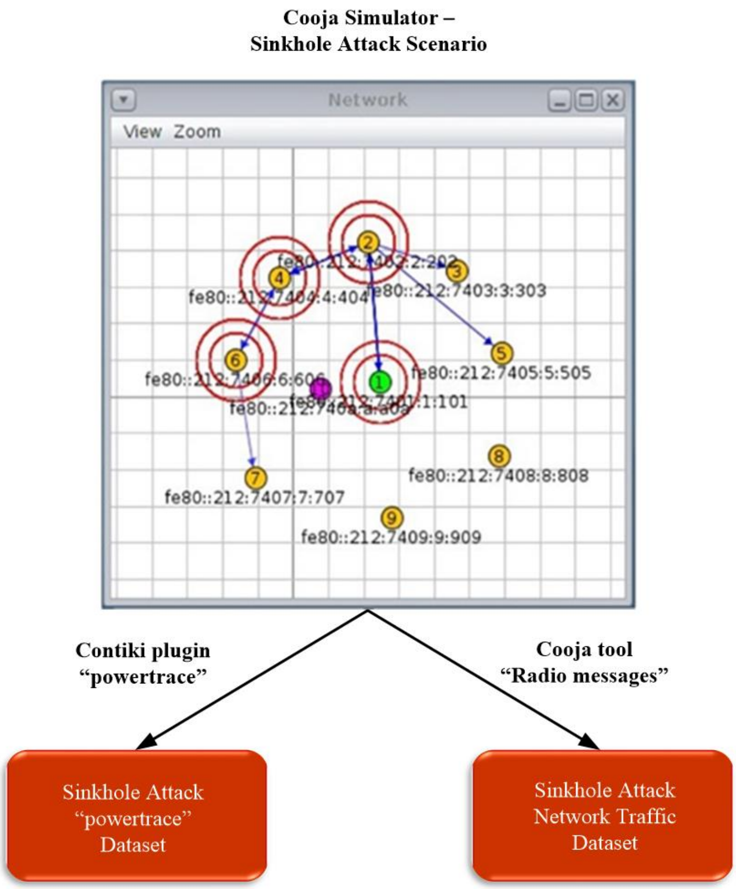
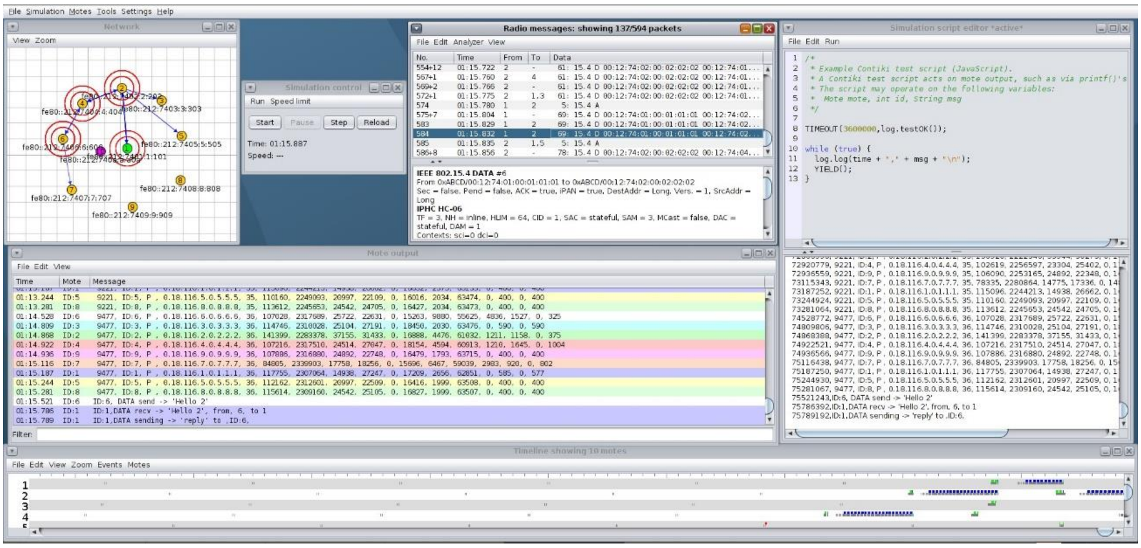


Figure 4.27 Sinkhole Attack Datasets generation by utilising the Cooja simulator.

4.4.1 Sinkhole Attack Scenario – an example

The network topology of the simulated sinkhole attack scenario in the Cooja simulator environment consists of 8 yellow (benign) UDP-client motes (i.e., motes 2, 3, 4, 5, 6, 7, 8 and 9), the violet (malicious) UDP-server mote (i.e., mote 10) and the green (benign) UDP-server mote (i.e., mote 1), as depicted in Figure 4.27. The simulation duration was set to 60 mins and the motes' outputs were printed out in the respective window (e.g., Mote output) while simulations run, as shown in Figure 4.28. Moreover, the 8 yellow (benign) UDP-client motes were configured to send text messages every 30 seconds, approximately, to the UDP-server mote that was configured to provide a corresponding response. On the other hand, the violet (malicious) UDP-server mote (i.e., mote 10) was compromised with malicious code, as shown in Figure 4.29 and Figure 4.30, to decrease the malicious mote's Rank number and make it the preferable parent node. With most of the neighbours to be connected to it, it starts dropping all the traffic that it should forward. The malicious mote was programmed to start 20 minutes later than the others allowing the network to work properly before the attack. Finally, it is noteworthy to say that similar to the benign network scenario, the UDP protocol was used at the Transport Layer, the IPv6 at the network layer, and the type of motes was the Tmote Sky in the sinkhole attack scenario.



Time: 01:15.887
Speed: ---

```
No.     Time       From  To   Data
554+12  01:15.722  2     -    61: 15.4 D 00:12:74:02:00:02:02:02 00:12:74:01...
567+1   01:15.760  2     4    61: 15.4 D 00:12:74:02:00:02:02:02 00:12:74:01...
569+2   01:15.766  2     -    61: 15.4 D 00:12:74:02:00:02:02:02 00:12:74:01...
572+1   01:15.775  2     1.3  61: 15.4 D 00:12:74:02:00:02:02:02 00:12:74:01...
574     01:15.780  1     2     5: 15.4 A
575+7   01:15.804  1     -    69: 15.4 D 00:12:74:01:00:01:01:01 00:12:74:02...
583     01:15.829  1     2    69: 15.4 D 00:12:74:01:00:01:01:01 00:12:74:02...
584     01:15.832  1     2    69: 15.4 D 00:12:74:01:00:01:01:01 00:12:74:02...
585     01:15.835  2     1.5   5: 15.4 A
586+8   01:15.856  2     -    78: 15.4 D 00:12:74:02:00:02:02:02 00:12:74:04...
```

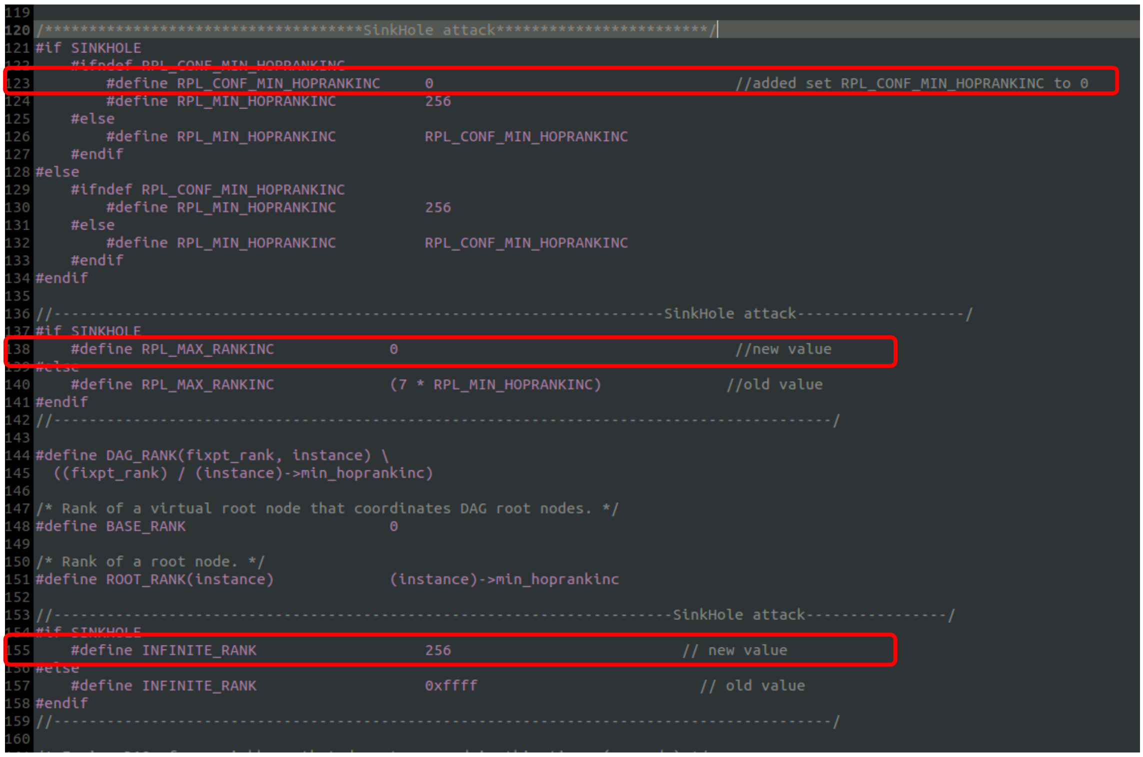
```
IEEE 802.15.4 DATA #6
From 0xABCD/00:12:74:01:00:01:01:01 to 0xABCD/00:12:74:02:00:02:02:02
Sec = false, Pend = false, ACK = true, iPAN = true, DestAddr = Long, Vers. = 1, SrcAddr =
Long
IPHC HC-06
TF = 3, NH = inline, HLIM = 64, CID = 1, SAC = stateful, SAM = 3, MCast = false, DAC =
stateful, DAM = 1
Contexts: sci=0 dci=0
```

```
/*
 * Example Contiki test script (JavaScript).
 * A Contiki test script acts on mote output, such as via printf()'s
 * The script may operate on the following variables:
 *  Mote mote, int id, String msg
 */

TIMEOUT(3600000,log.testOK());

while (true) {
  log.log(time + "," + msg + "\n");
  YIELD();
}
```

```
Time       Mote  Message
01:13.244  ID:5   9221, ID:5, P , 0.18.116.5.0.5.5.5, 35, 110160, 2249093, 20997, 22109, 0, 16016, 2034, 63474, 0, 400, 0, 400
01:13.281  ID:8   9221, ID:8, P , 0.18.116.8.0.8.8.8, 35, 113612, 2245653, 24542, 24705, 0, 16427, 2034, 63473, 0, 400, 0, 400
01:14.528  ID:6   9477, ID:6, P , 0.18.116.6.0.6.6.6, 36, 107028, 2317689, 25722, 22631, 0, 15263, 9880, 55625, 4836, 1527, 0, 325
01:14.809  ID:3   9477, ID:3, P , 0.18.116.3.0.3.3.3, 36, 114746, 2310028, 25104, 27191, 0, 18450, 2030, 63476, 0, 590, 0, 590
01:14.868  ID:2   9477, ID:2, P , 0.18.116.2.0.2.2.2, 36, 141399, 2283378, 37155, 31433, 0, 16888, 4476, 61032, 1211, 1158, 0, 375
01:14.922  ID:4   9477, ID:4, P , 0.18.116.4.0.4.4.4, 36, 107216, 2317510, 24514, 27047, 0, 18154, 4594, 60913, 1210, 1645, 0, 1004
01:14.936  ID:9   9477, ID:9, P , 0.18.116.9.0.9.9.9, 36, 107886, 2316880, 24892, 22748, 0, 16479, 1793, 63715, 0, 400, 0, 400
01:15.116  ID:7   9477, ID:7, P , 0.18.116.7.0.7.7.7, 36, 84805, 2339903, 17758, 18256, 0, 15696, 6467, 59039, 2983, 920, 0, 802
01:15.187  ID:1   9477, ID:1, P , 0.18.116.1.0.1.1.1, 36, 117755, 2307064, 14938, 27247, 0, 17209, 2656, 62851, 0, 585, 0, 577
01:15.244  ID:5   9477, ID:5, P , 0.18.116.5.0.5.5.5, 36, 112162, 2312601, 20997, 22509, 0, 16416, 1999, 63508, 0, 400, 0, 400
01:15.281  ID:8   9477, ID:8, P , 0.18.116.8.0.8.8.8, 36, 115614, 2309160, 24542, 25105, 0, 16827, 1999, 63507, 0, 400, 0, 400
01:15.521  ID:6  ID:6, DATA send -> 'Hello 2'
01:15.786  ID:1  ID:1,DATA recv -> 'Hello 2', from, 6, to 1
01:15.789  ID:1  ID:1,DATA sending -> 'reply' to ,ID:6,
```

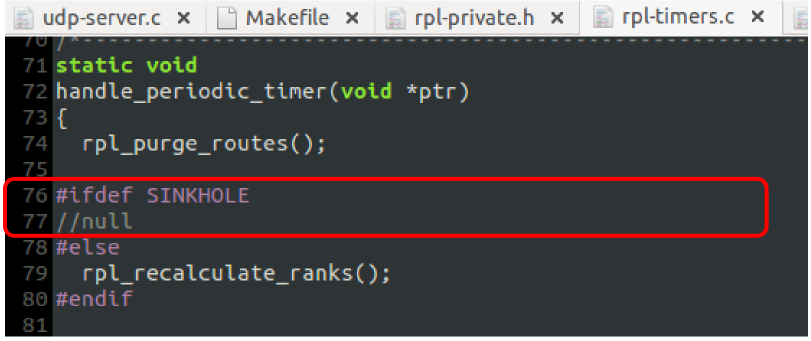
```
72920779, 9221, ID:4, P , 0.18.116.4.0.4.4.4, 35, 102619, 2256597, 23304, 25402, 0, 1
72936559, 9221, ID:9, P , 0.18.116.9.0.9.9.9, 35, 106090, 2253165, 24892, 22348, 0, 1
73115343, 9221, ID:7, P , 0.18.116.7.0.7.7.7, 35, 78335, 2280864, 14775, 17336, 0, 14
73187252, 9221, ID:1, P , 0.18.116.1.0.1.1.1, 35, 115096, 2244213, 14938, 26662, 0, 1
73244924, 9221, ID:5, P , 0.18.116.5.0.5.5.5, 35, 110160, 2249093, 20997, 22109, 0, 1
73281064, 9221, ID:8, P , 0.18.116.8.0.8.8.8, 35, 113612, 2245653, 24542, 24705, 0, 1
74528772, 9477, ID:6, P , 0.18.116.6.0.6.6.6, 36, 107028, 2317689, 25722, 22631, 0, 1
74809806, 9477, ID:3, P , 0.18.116.3.0.3.3.3, 36, 114746, 2310028, 25104, 27191, 0, 1
74868388, 9477, ID:2, P , 0.18.116.2.0.2.2.2, 36, 141399, 2283378, 37155, 31433, 0, 1
74922521, 9477, ID:4, P , 0.18.116.4.0.4.4.4, 36, 107216, 2317510, 24514, 27047, 0, 1
74936566, 9477, ID:9, P , 0.18.116.9.0.9.9.9, 36, 107886, 2316880, 24892, 22748, 0, 1
75116438, 9477, ID:7, P , 0.18.116.7.0.7.7.7, 36, 84805, 2339903, 17758, 18256, 0, 15
75187250, 9477, ID:1, P , 0.18.116.1.0.1.1.1, 36, 117755, 2307064, 14938, 27247, 0, 1
75244930, 9477, ID:5, P , 0.18.116.5.0.5.5.5, 36, 112162, 2312601, 20997, 22509, 0, 1
75281067, 9477, ID:8, P , 0.18.116.8.0.8.8.8, 36, 115614, 2309160, 24542, 25105, 0, 1
75521243,ID:6, DATA send -> 'Hello 2'
75786392,ID:1,DATA recv -> 'Hello 2', from, 6, to 1
75789192,ID:1,DATA sending -> 'reply' to ,ID:6,
```

Figure 4.28 Cooja Simulator – Sinkhole attack scenario – Motes' outputs.



```

/*************************************SinkHole attack************************/
#if SINKHOLE
    #ifndef RPL_CONF_MIN_HOPRANKINC
        #define RPL_CONF_MIN_HOPRANKINC     0                                   //added set RPL_CONF_MIN_HOPRANKINC to 0
        #define RPL_MIN_HOPRANKINC          256
    #else
        #define RPL_MIN_HOPRANKINC          RPL_CONF_MIN_HOPRANKINC
    #endif
#else
    #ifndef RPL_CONF_MIN_HOPRANKINC
        #define RPL_MIN_HOPRANKINC          256
    #else
        #define RPL_MIN_HOPRANKINC          RPL_CONF_MIN_HOPRANKINC
    #endif
#endif

//----------------------------------------------------------------------SinkHole attack------------------/
#if SINKHOLE
    #define RPL_MAX_RANKINC             0                                   //new value
#else
    #define RPL_MAX_RANKINC             (7 * RPL_MIN_HOPRANKINC)            //old value
#endif
//------------------------------------------------------------------------------------------/

#define DAG_RANK(fixpt_rank, instance) \
  ((fixpt_rank) / (instance)->min_hoprankinc)

/* Rank of a virtual root node that coordinates DAG root nodes. */
#define BASE_RANK                       0

/* Rank of a root node. */
#define ROOT_RANK(instance)             (instance)->min_hoprankinc

//----------------------------------------------------------------------SinkHole attack---------------/
#if SINKHOLE
    #define INFINITE_RANK                   256                         // new value
#else
    #define INFINITE_RANK                   0xffff                        // old value
#endif
//------------------------------------------------------------------------------------------/

```

Figure 4.29 Malicious code in "contiki_modified/core/net/rpl/rpl-private.c" to cause a sinkhole attack.



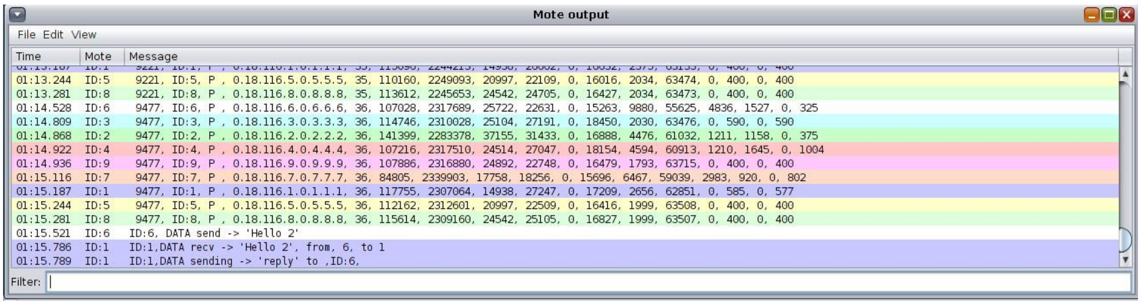
```
/*---------------------------------------------------------------------------*/
static void
handle_periodic_timer(void *ptr)
{
  rpl_purge_routes();

#ifdef SINKHOLE
//null
#else
  rpl_recalculate_ranks();
#endif

```

Figure 4.30 Malicious code in "contiki_modified/core/net/rpl/rpl-timers.c" to cause a sinkhole attack.

4.4.2 Sinkhole Attack “powertrace” Dataset

4.4.2.1 Sinkhole Attack “powertrace” Dataset – Generation Process

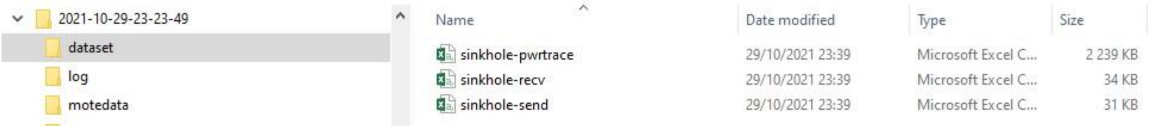
The approach followed for the “powertrace” dataset generation from the sinkhole attack scenario was similar to the approach followed for the “powertrace” dataset generation from the benign IoT network scenario in Section 3.3.1. In addition, the “powertrace” plugin was similarly enabled for collecting “powertrace” related features, summarised in Table 3, from the motes of the attack scenario every two seconds. In Figure 4.31, the depicted mote output window displays the captured “powertrace” information every two seconds and also the messages sent and received by each mote during the simulation time (60 mins).



```
Time       Mote   Message
01:13.244  ID:5    9221, ID:5, P , 0.18.116.5.0.5.5.5, 35, 110160, 2249093, 20997, 22109, 0, 16016, 2034, 63474, 0, 400, 0, 400
01:13.281  ID:8    9221, ID:8, P , 0.18.116.8.0.8.8.8, 35, 113612, 2245653, 24542, 24705, 0, 16427, 2034, 63473, 0, 400, 0, 400
01:14.528  ID:6    9477, ID:6, P , 0.18.116.6.0.6.6.6, 36, 107028, 2317689, 25722, 22631, 0, 15263, 9880, 55625, 4836, 1527, 0, 325
01:14.809  ID:3    9477, ID:3, P , 0.18.116.3.0.3.3.3, 36, 114746, 2310028, 25104, 27191, 0, 18450, 2030, 63476, 0, 590, 0, 590
01:14.868  ID:2    9477, ID:2, P , 0.18.116.2.0.2.2.2, 36, 141399, 2283378, 37155, 31433, 0, 16888, 4476, 61032, 1211, 1158, 0, 375
01:14.922  ID:4    9477, ID:4, P , 0.18.116.4.0.4.4.4, 36, 107216, 2317510, 24514, 27047, 0, 18154, 4594, 60913, 1210, 1645, 0, 1004
01:14.936  ID:9    9477, ID:9, P , 0.18.116.9.0.9.9.9, 36, 107886, 2316880, 24892, 22748, 0, 16479, 1793, 63715, 0, 400, 0, 400
01:15.116  ID:7    9477, ID:7, P , 0.18.116.7.0.7.7.7, 36, 84805, 2339903, 17758, 18256, 0, 15696, 6467, 59039, 2983, 920, 0, 802
01:15.187  ID:1    9477, ID:1, P , 0.18.116.1.0.1.1.1, 36, 117755, 2307064, 14938, 27247, 0, 17209, 2656, 62851, 0, 585, 0, 577
01:15.244  ID:5    9477, ID:5, P , 0.18.116.5.0.5.5.5, 36, 112162, 2312601, 20997, 22509, 0, 16416, 1999, 63508, 0, 400, 0, 400
01:15.281  ID:8    9477, ID:8, P , 0.18.116.8.0.8.8.8, 36, 115614, 2309160, 24542, 25105, 0, 16827, 1999, 63507, 0, 400, 0, 400
01:15.521  ID:6   ID:6, DATA send -> 'Hello 2'
01:15.786  ID:1   ID:1,DATA recv -> 'Hello 2', from, 6, to 1
01:15.789  ID:1   ID:1,DATA sending -> 'reply' to ,ID:6,
Filter:
```

Figure 4.31 Cooja Simulator – Sinkhole attack scenario – Mote output window

When the timeout occurred, the simulation stopped, and all the captured information and prints were stored in the “COOJA.testlog” file. Afterwards, the “IoT_Simul.sh” file, described in Section 3.3.1, created a) a new root folder named as “2021-10-29-23-23-49”, and b) the “log” folder, inside the “2021-10-29-23-23-49” folder, where the “COOJA.testlog” file was copied from the “…/cooja/build” folder located in the Cooja Simulator. Then, the “IoT_Simul.sh” file following the same process, as described in Section 3.3.1, extracted the required “powertrace” information from the “COOJA.testlog” file and saved it in the “sinkhole-pwrtrace.csv” file in the “dataset” folder that was also created by the batch file inside the “2021-10-29-23-23-49” folder, as shown below in the left part of Figure 4.32. In the “dataset” folder, apart from the “sinkhole-pwrtrace.csv” file, the “IoT_Simul.sh” file generated two more files (i.e., “sinkhole-recv.csv” and “sinkhole-send.csv”), following the same process as in Section 3.3.1. The “sinkhole-recv.csv” file and the “sinkhole-send.csv” file include the “received” and “sent” messages printed by the motes, respectively.



2021-10-29-23-23-49
dataset
log
motedata

Name	Date modified	Type	Size
sinkhole-pwrtrace	29/10/2021 23:39	Microsoft Excel C...	2 239 KB
sinkhole-recv	29/10/2021 23:39	Microsoft Excel C...	34 KB
sinkhole-send	29/10/2021 23:39	Microsoft Excel C...	31 KB

Figure 4.32 Location of the generated “sinkhole-pwrtrace.csv”, “sinkhole-recv.csv”, and “sinkhole-send.csv” files by the “IoT_Simul.sh” bash file.

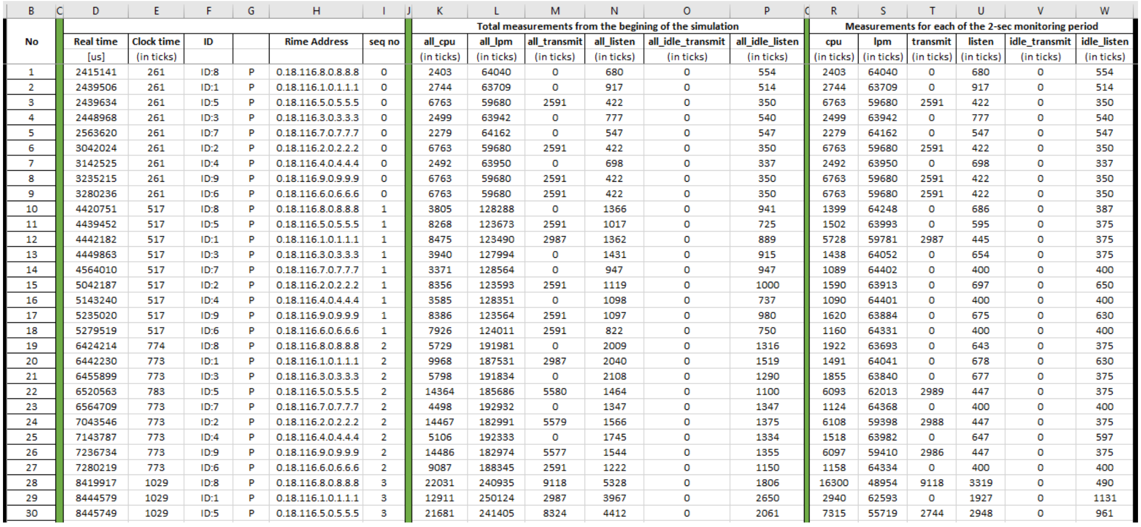
Finally, similar to the benign “powertrace” dataset generation approach in Section 3.3.1, the “IoT_Simul.sh” file extracted the information related to each mote from the “sinkhole-pwrtrace.csv” file and generated one csv file for each mote with the corresponding information from the “sinkhole-pwrtrace.csv” file. The generated ten csv files (i.e., sinkhole-mote1.csv, …, sinkhole-mote10.csv) were stored in the “motedata” folder, created also by the “IoT_Simul.sh” file, as shown in the left part of Figure 4.32.

4.4.2.2 Sinkhole Attack “powertrace” Dataset – Generated Results

The sinkhole attack “powertrace” dataset consists of the following csv files: “sinkhole-pwrtrace.csv”, “sinkhole-mote1.csv”, “sinkhole-mote2.csv”, “sinkhole-mote3.csv”, “sinkhole-mote4.csv”, “sinkhole-mote5.csv”, “sinkhole-mote6.csv”, “sinkhole-mote7.csv”, “sinkhole-mote8.csv”, “sinkhole-mote9.csv”, and “sinkhole-mote10.csv” files. In this Section, we present sets of records from the “sinkhole-pwrtrace.csv”, and in Appendix 1 we present sets of records from “sinkhole-mote1.csv”, “sinkhole-mote5.csv” and “sinkhole-mote10.csv” files.

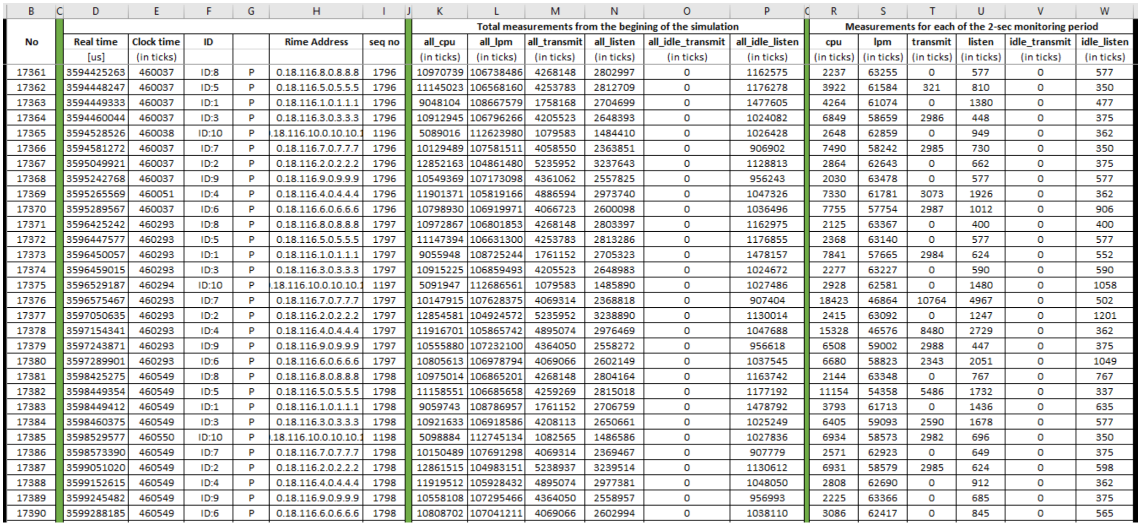
4.4.2.2.1 “sinkhole-pwrtrace.csv”

The generated malicious “sinkhole-pwrtrace.csv” file consists of 17,390 records and its first 30 records (i.e., 1–30) and its last 30 records (17,361–17,390) are depicted in Figure 4.33 and Figure 4.34, respectively.



B	D	E	F	G	H	I	K	L	M	N	O	P	R	S	T	U	V	W
No							Total measurements from the begining of the simulation						Measurements for each of the 2-sec monitoring period					
	Real time	Clock time	ID		Rime Address	seq no	all_cpu	all_lpm	all_transmit	all_listen	all_idle_transmit	all_idle_listen	cpu	lpm	transmit	listen	idle_transmit	idle_listen
	[us]	(in ticks)					(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)
1	2415141	261	ID:8	P	0.18.116.8.0.8.8.8	0	2403	64040	0	680	0	554	2403	64040	0	680	0	554
2	2439506	261	ID:1	P	0.18.116.1.0.1.1.1	0	2744	63709	0	917	0	514	2744	63709	0	917	0	514
3	2439634	261	ID:5	P	0.18.116.5.0.5.5.5	0	6763	59680	2591	422	0	350	6763	59680	2591	422	0	350
4	2448968	261	ID:3	P	0.18.116.3.0.3.3.3	0	2499	63942	0	777	0	540	2499	63942	0	777	0	540
5	2563620	261	ID:7	P	0.18.116.7.0.7.7.7	0	2279	64162	0	547	0	547	2279	64162	0	547	0	547
6	3042024	261	ID:2	P	0.18.116.2.0.2.2.2	0	6763	59680	2591	422	0	350	6763	59680	2591	422	0	350
7	3142525	261	ID:4	P	0.18.116.4.0.4.4.4	0	2492	63950	0	698	0	337	2492	63950	0	698	0	337
8	3235215	261	ID:9	P	0.18.116.9.0.9.9.9	0	6763	59680	2591	422	0	350	6763	59680	2591	422	0	350
9	3280236	261	ID:6	P	0.18.116.6.0.6.6.6	0	6763	59680	2591	422	0	350	6763	59680	2591	422	0	350
10	4420751	517	ID:8	P	0.18.116.8.0.8.8.8	1	3805	128288	0	1366	0	941	1399	64248	0	686	0	387
11	4439452	517	ID:5	P	0.18.116.5.0.5.5.5	1	8268	123673	2591	1017	0	725	1502	63993	0	595	0	375
12	4442182	517	ID:1	P	0.18.116.1.0.1.1.1	1	8475	123490	2987	1362	0	889	5728	59781	2987	445	0	375
13	4449863	517	ID:3	P	0.18.116.3.0.3.3.3	1	3940	127994	0	1431	0	915	1438	64052	0	654	0	375
14	4564010	517	ID:7	P	0.18.116.7.0.7.7.7	1	3371	128564	0	947	0	947	1089	64402	0	400	0	400
15	5042187	517	ID:2	P	0.18.116.2.0.2.2.2	1	8356	123593	2591	1119	0	1000	1590	63913	0	697	0	650
16	5143240	517	ID:4	P	0.18.116.4.0.4.4.4	1	3585	128351	0	1098	0	737	1090	64401	0	400	0	400
17	5235020	517	ID:9	P	0.18.116.9.0.9.9.9	1	8386	123564	2591	1097	0	980	1620	63884	0	675	0	630
18	5279519	517	ID:6	P	0.18.116.6.0.6.6.6	1	7926	124011	2591	822	0	750	1160	64331	0	400	0	400
19	6424214	774	ID:8	P	0.18.116.8.0.8.8.8	2	5729	191981	0	2009	0	1316	1922	63693	0	643	0	375
20	6442230	773	ID:1	P	0.18.116.1.0.1.1.1	2	9968	187531	2987	2040	0	1519	1491	64041	0	678	0	630
21	6455899	773	ID:3	P	0.18.116.3.0.3.3.3	2	5798	191834	0	2108	0	1290	1855	63840	0	677	0	375
22	6520563	783	ID:5	P	0.18.116.5.0.5.5.5	2	14364	185686	5580	1464	0	1100	6093	62013	2989	447	0	375
23	6564709	773	ID:7	P	0.18.116.7.0.7.7.7	2	4498	192932	0	1347	0	1347	1124	64368	0	400	0	400
24	7043546	773	ID:2	P	0.18.116.2.0.2.2.2	2	14467	182991	5579	1566	0	1375	6108	59398	2988	447	0	375
25	7143787	773	ID:4	P	0.18.116.4.0.4.4.4	2	5106	192333	0	1745	0	1334	1518	63982	0	647	0	597
26	7236734	773	ID:9	P	0.18.116.9.0.9.9.9	2	14486	182974	5577	1544	0	1355	6097	59410	2986	447	0	375
27	7280219	773	ID:6	P	0.18.116.6.0.6.6.6	2	9087	188345	2591	1222	0	1150	1158	64334	0	400	0	400
28	8419917	1029	ID:8	P	0.18.116.8.0.8.8.8	3	22031	240935	9118	5328	0	1806	16300	48954	9118	3319	0	490
29	8444579	1029	ID:1	P	0.18.116.1.0.1.1.1	3	12911	250124	2987	3967	0	2650	2940	62593	0	1927	0	1131
30	8445749	1029	ID:5	P	0.18.116.5.0.5.5.5	3	21681	241405	8324	4412	0	2061	7315	55719	2744	2948	0	961

Figure 4.33 Malicious “sinkhole-pwrtrace.csv” – 1 to 30 records



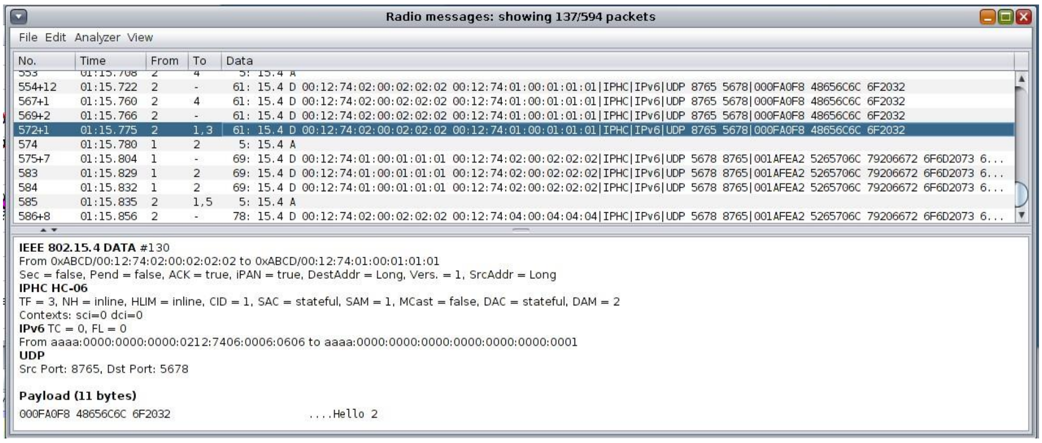
B	D	E	F	G	H	I	K	L	M	N	O	P	R	S	T	U	V	W
No							Total measurements from the begining of the simulation						Measurements for each of the 2-sec monitoring period					
	Real time	Clock time	ID		Rime Address	seq no	all_cpu	all_lpm	all_transmit	all_listen	all_idle_transmit	all_idle_listen	cpu	lpm	transmit	listen	idle_transmit	idle_listen
	[us]	(in ticks)					(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)
17361	3594425263	460037	ID:8	P	0.18.116.8.0.8.8.8	1796	10970739	106738486	4268148	2802997	0	1162575	2237	63255	0	577	0	577
17362	3594448247	460037	ID:5	P	0.18.116.5.0.5.5.5	1796	11145023	106568160	4253783	2812709	0	1176278	3922	61584	321	810	0	350
17363	3594449333	460037	ID:1	P	0.18.116.1.0.1.1.1	1796	9048104	108667579	1758168	2704699	0	1477605	4264	61074	0	1380	0	477
17364	3594460044	460037	ID:3	P	0.18.116.3.0.3.3.3	1796	10912945	106796266	4205523	2648393	0	1024082	6849	58659	2986	448	0	375
17365	3594528526	460038	ID:10	P	.18.116.10.0.10.10.1	1196	5089016	112623980	1079583	1484410	0	1026428	2648	62859	0	949	0	362
17366	3594581272	460037	ID:7	P	0.18.116.7.0.7.7.7	1796	10129489	107581511	4058550	2363851	0	906902	7490	58242	2985	730	0	350
17367	3595049921	460037	ID:2	P	0.18.116.2.0.2.2.2	1796	12852163	104861480	5235952	3237643	0	1128813	2864	62643	0	662	0	375
17368	3595242768	460037	ID:9	P	0.18.116.9.0.9.9.9	1796	10549369	107173098	4361062	2557825	0	956243	2030	63478	0	577	0	577
17369	3595265569	460051	ID:4	P	0.18.116.4.0.4.4.4	1796	11901371	105819166	4886594	2973740	0	1047326	7330	61781	3073	1926	0	362
17370	3595289567	460037	ID:6	P	0.18.116.6.0.6.6.6	1796	10798930	106919971	4066723	2600098	0	1036496	7755	57754	2987	1012	0	906
17371	3596425242	460293	ID:8	P	0.18.116.8.0.8.8.8	1797	10972867	106801853	4268148	2803397	0	1162975	2125	63367	0	400	0	400
17372	3596447577	460293	ID:5	P	0.18.116.5.0.5.5.5	1797	11147394	106631300	4253783	2813286	0	1176855	2368	63140	0	577	0	577
17373	3596450057	460293	ID:1	P	0.18.116.1.0.1.1.1	1797	9055948	108725244	1761152	2705323	0	1478157	7841	57665	2984	624	0	552
17374	3596459015	460293	ID:3	P	0.18.116.3.0.3.3.3	1797	10915225	106859493	4205523	2648983	0	1024672	2277	63227	0	590	0	590
17375	3596529187	460294	ID:10	P	.18.116.10.0.10.10.1	1197	5091947	112686561	1079583	1485890	0	1027486	2928	62581	0	1480	0	1058
17376	3596575467	460293	ID:7	P	0.18.116.7.0.7.7.7	1797	10147915	107628375	4069314	2368818	0	907404	18423	46864	10764	4967	0	502
17377	3597050635	460293	ID:2	P	0.18.116.2.0.2.2.2	1797	12854581	104924572	5235952	3238890	0	1130014	2415	63092	0	1247	0	1201
17378	3597154341	460293	ID:4	P	0.18.116.4.0.4.4.4	1797	11916701	105865742	4895074	2976469	0	1047688	15328	46576	8480	2729	0	362
17379	3597243871	460293	ID:9	P	0.18.116.9.0.9.9.9	1797	10555880	107232100	4364050	2558272	0	956618	6508	59002	2988	447	0	375
17380	3597289901	460293	ID:6	P	0.18.116.6.0.6.6.6	1797	10805613	106978794	4069066	2602149	0	1037545	6680	58823	2343	2051	0	1049
17381	3598425275	460549	ID:8	P	0.18.116.8.0.8.8.8	1798	10975014	106865201	4268148	2804164	0	1163742	2144	63348	0	767	0	767
17382	3598449354	460549	ID:5	P	0.18.116.5.0.5.5.5	1798	11158551	106685658	4259269	2815018	0	1177192	11154	54358	5486	1732	0	337
17383	3598449412	460549	ID:1	P	0.18.116.1.0.1.1.1	1798	9059743	108786957	1761152	2706759	0	1478792	3793	61713	0	1436	0	635
17384	3598460375	460549	ID:3	P	0.18.116.3.0.3.3.3	1798	10921633	106918586	4208113	2650661	0	1025249	6405	59093	2590	1678	0	577
17385	3598529577	460550	ID:10	P	.18.116.10.0.10.10.1	1198	5098884	112745134	1082565	1486586	0	1027836	6934	58573	2982	696	0	350
17386	3598573390	460549	ID:7	P	0.18.116.7.0.7.7.7	1798	10150489	107691298	4069314	2369467	0	907779	2571	62923	0	649	0	375
17387	3599051020	460549	ID:2	P	0.18.116.2.0.2.2.2	1798	12861515	104983151	5238937	3239514	0	1130612	6931	58579	2985	624	0	598
17388	3599152615	460549	ID:4	P	0.18.116.4.0.4.4.4	1798	11919512	105928432	4895074	2977381	0	1048050	2808	62690	0	912	0	362
17389	3599245482	460549	ID:9	P	0.18.116.9.0.9.9.9	1798	10558108	107295466	4364050	2558957	0	956993	2225	63366	0	685	0	375
17390	3599288185	460549	ID:6	P	0.18.116.6.0.6.6.6	1798	10808702	107041211	4069066	2602994	0	1038110	3086	62417	0	845	0	565

Figure 4.34 Malicious “sinkhole-pwrtrace.csv”— 17,361 to 17,390 records.

4.4.3 Sinkhole Attack Network Traffic Dataset

4.4.3.1 Sinkhole Attack Network Traffic Dataset – Generation Process

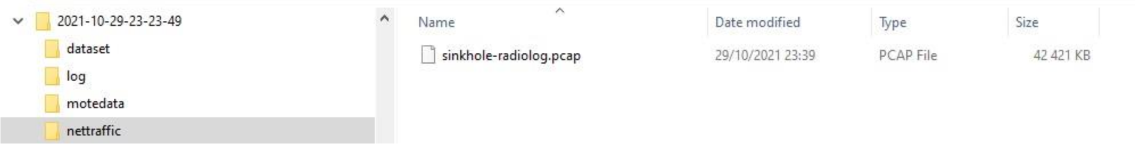
The approach followed for the network traffic dataset generation from the sinkhole attack scenario was similar to the approach followed for the network traffic dataset generation from the benign IoT network scenario in Section 3.4.1. The "Radio messages" tool, provided by the Cooja simulator, was similarly used for collecting data related to the corresponding network traffic features (e.g., source/destination IPv6 address, packet size, and communication protocol) from the network of the attack scenario. During the simulation, the network traffic information was being shown in the top part of the "Radio messages" output window as depicted in the top part of Figure 4.35.



```
No.       Time       From  To    Data
553       01:15.708  2     4       5: 15.4 A
554+12    01:15.722  2     -      61: 15.4 D 00:12:74:02:00:02:02:02 00:12:74:01:00:01:01:01|IPHC|IPv6|UDP 8765 5678|000FA0F8 48656C6C 6F2032
567+1     01:15.760  2     4      61: 15.4 D 00:12:74:02:00:02:02:02 00:12:74:01:00:01:01:01|IPHC|IPv6|UDP 8765 5678|000FA0F8 48656C6C 6F2032
569+2     01:15.766  2     -      61: 15.4 D 00:12:74:02:00:02:02:02 00:12:74:01:00:01:01:01|IPHC|IPv6|UDP 8765 5678|000FA0F8 48656C6C 6F2032
572+1     01:15.775  2     1,3    61: 15.4 D 00:12:74:02:00:02:02:02 00:12:74:01:00:01:01:01|IPHC|IPv6|UDP 8765 5678|000FA0F8 48656C6C 6F2032
574       01:15.780  1     2       5: 15.4 A
575+7     01:15.804  1     -      69: 15.4 D 00:12:74:01:00:01:01:01 00:12:74:02:00:02:02:02|IPHC|IPv6|UDP 5678 8765|001AFEA2 5265706C 79206672 6F6D2073 6...
583       01:15.829  1     2      69: 15.4 D 00:12:74:01:00:01:01:01 00:12:74:02:00:02:02:02|IPHC|IPv6|UDP 5678 8765|001AFEA2 5265706C 79206672 6F6D2073 6...
584       01:15.832  1     2      69: 15.4 D 00:12:74:01:00:01:01:01 00:12:74:02:00:02:02:02|IPHC|IPv6|UDP 5678 8765|001AFEA2 5265706C 79206672 6F6D2073 6...
585       01:15.835  2     1,5     5: 15.4 A
586+8     01:15.856  2     -      78: 15.4 D 00:12:74:02:00:02:02:02 00:12:74:04:00:04:04:04|IPHC|IPv6|UDP 5678 8765|001AFEA2 5265706C 79206672 6F6D2073 6...

IEEE 802.15.4 DATA #130
From 0xABCD/00:12:74:02:00:02:02:02 to 0xABCD/00:12:74:01:00:01:01:01
Sec = false, Pend = false, ACK = true, iPAN = true, DestAddr = Long, Vers. = 1, SrcAddr = Long
IPHC HC-06
TF = 3, NH = inline, HLIM = inline, CID = 1, SAC = stateful, SAM = 1, MCast = false, DAC = stateful, DAM = 2
Contexts: sci=0 dci=0
IPv6 TC = 0, FL = 0
From aaaa:0000:0000:0000:0212:7406:0006:0606 to aaaa:0000:0000:0000:0000:0000:0000:0001
UDP
Src Port: 8765, Dst Port: 5678

Payload (11 bytes)
000FA0F8 48656C6C 6F2032                    ....Hello 2
```

Figure 4.35 Network traffic information from the sinkhole attack scenario in the "Radio messages" output window.

When the simulation stopped, the generated pcap file was saved as "radiolog.pcap" within the ".../cooja/build" folder. Afterwards, the "IoT_Simul.sh" file, described in Section 3.4.1, created a) a new root folder named as "2021-10-29-23-23-49", and b) the "nettraffic" folder, inside the "2021-10-29-23-23-49" folder, where the "radiolog.pcap" file, copied from the ".../cooja/build" folder located in the Cooja Simulator, was saved as "sinkhole-radiolog.pcap". The "nettraffic" folder inside the root folder "2021-10-29-23-23-49" and the "sinkhole-radiolog.pcap" file in the "nettraffic" folder are shown in Figure 4.36.

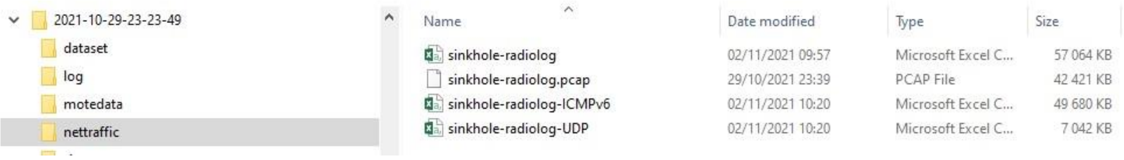


2021-10-29-23-23-49
dataset
log
motedata
nettraffic

Name	Date modified	Type	Size
sinkhole-radiolog.pcap	29/10/2021 23:39	PCAP File	42 421 KB

Figure 4.36 The "nettraffic" folder inside the root folder "2021-10-29-23-23-49" and the "sinkhole-radiolog.pcap" file.

Then, following the same process, as described in Section 3.4.1, we used Wireshark to extract the stored network traffic information from the "sinkhole-radiolog.pcap" file to the "sinkhole-radiolog.csv" file stored in the "nettraffic" folder as shown in Figure 4.37.



2021-10-29-23-23-49
dataset
log
motedata
nettraffic

Name	Date modified	Type	Size
sinkhole-radiolog	02/11/2021 09:57	Microsoft Excel C...	57 064 KB
sinkhole-radiolog.pcap	29/10/2021 23:39	PCAP File	42 421 KB
sinkhole-radiolog-ICMPv6	02/11/2021 10:20	Microsoft Excel C...	49 680 KB
sinkhole-radiolog-UDP	02/11/2021 10:20	Microsoft Excel C...	7 042 KB

Figure 4.37 The "nettraffic" folder inside the root folder "2021-10-29-23-23-49" and its included files.

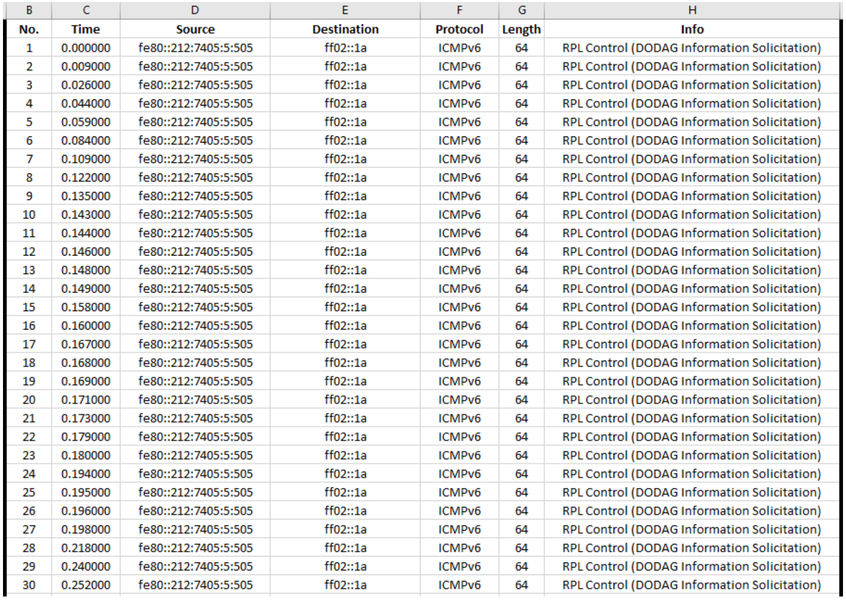
In the "nettraffic" folder, apart from the "sinkhole-radiolog.csv" file, we also used Wireshark, following the same process as in Section 3.4.1, to generate two more files (i.e., "sinkhole-radiolog-ICMPv6.csv" and "sinkhole-radiolog-UDP.csv") from the "sinkhole-radiolog.pcap" file.

4.4.3.2 Sinkhole Attack Network Traffic Dataset – Generated Results

The sinkhole attack network traffic dataset consists of the following csv files which are located in the "nettraffic" folder as described in Section 4.4.3.1: "sinkhole-radiolog.csv", "sinkhole-radiolog-ICMPv6.csv", and "sinkhole-radiolog-UDP.csv" files. In this Section, we present sets of records from these files.

4.4.3.2.1 "sinkhole-radiolog.csv"

The generated malicious "sinkhole-radiolog.csv" file consists of 463,581 records and its first 30 records (i.e., 1–30) are depicted below in Figure 4.38.

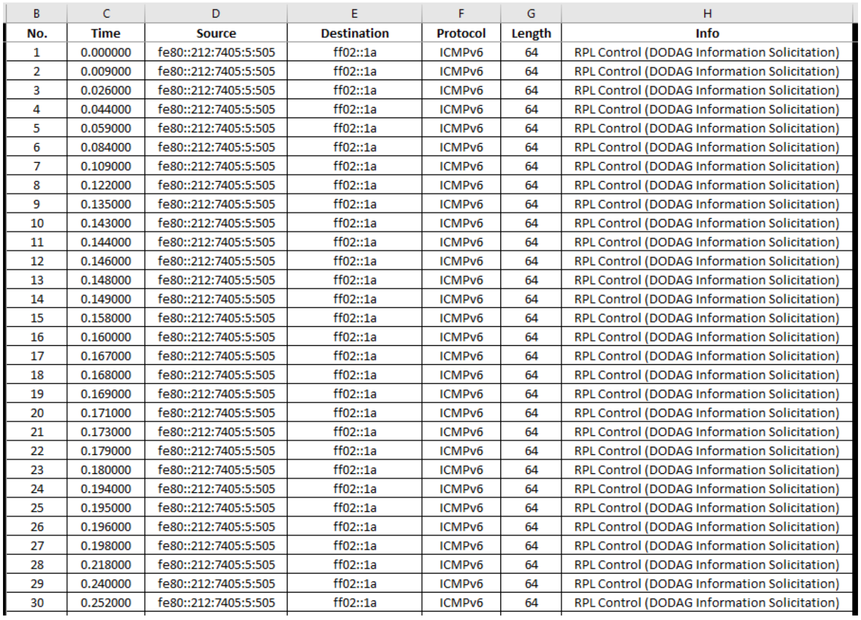


B	C	D	E	F	G	H
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
2	0.009000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
3	0.026000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
4	0.044000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
5	0.059000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
6	0.084000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
7	0.109000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
8	0.122000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
9	0.135000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
10	0.143000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
11	0.144000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
12	0.146000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
13	0.148000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
14	0.149000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
15	0.158000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
16	0.160000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
17	0.167000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
18	0.168000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
19	0.169000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
20	0.171000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
21	0.173000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
22	0.179000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
23	0.180000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
24	0.194000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
25	0.195000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
26	0.196000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
27	0.198000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
28	0.218000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
29	0.240000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
30	0.252000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)

Figure 4.38 Malicious "sinkhole-radiolog.csv"—1 to 30 records.

4.4.3.2.2 "sinkhole-radiolog-ICMPv6.csv"

The generated malicious "sinkhole-radiolog-ICMPv6.csv" file consists of 404,290 records and its first 30 records (i.e., 1–30) are depicted below in Figure 4.39.

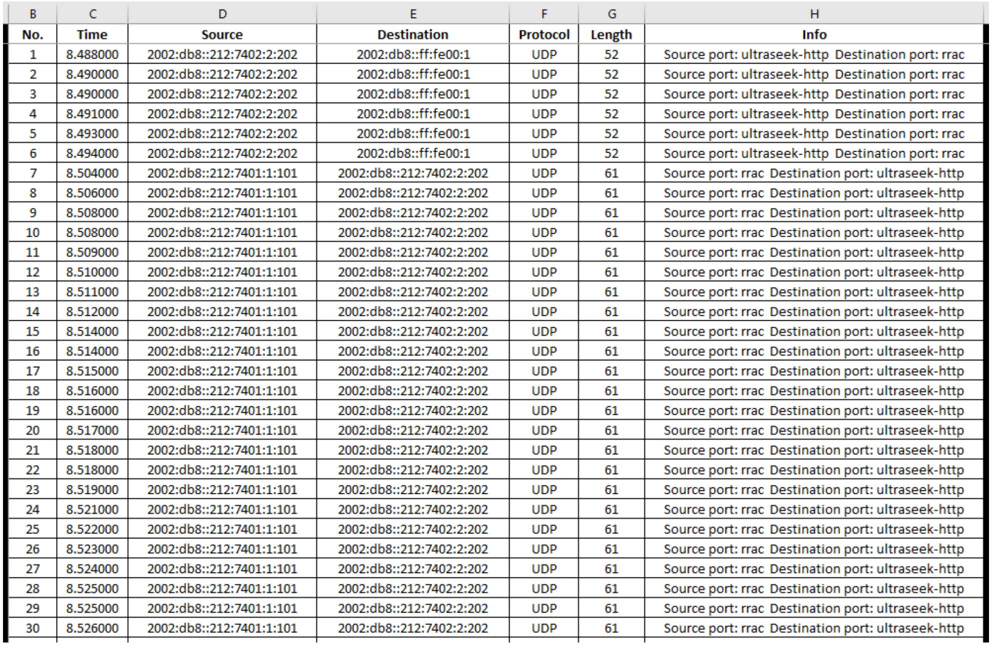


B	C	D	E	F	G	H
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
2	0.009000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
3	0.026000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
4	0.044000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
5	0.059000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
6	0.084000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
7	0.109000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
8	0.122000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
9	0.135000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
10	0.143000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
11	0.144000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
12	0.146000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
13	0.148000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
14	0.149000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
15	0.158000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
16	0.160000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
17	0.167000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
18	0.168000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
19	0.169000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
20	0.171000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
21	0.173000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
22	0.179000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
23	0.180000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
24	0.194000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
25	0.195000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
26	0.196000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
27	0.198000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
28	0.218000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
29	0.240000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
30	0.252000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)

Figure 4.39 Malicious “sinkhole-radiolog-ICMPv6.csv”—1 to 30 records.

4.4.3.2.3 “sinkhole-radiolog-UDP.csv”

The generated malicious “sinkhole-radiolog-UDP.csv” file consists of 52,750 records and its first 30 records (i.e., 1–30) are depicted below in Figure 4.40.



B	C	D	E	F	G	H
No.	Time	Source	Destination	Protocol	Length	Info
1	8.488000	2002:db8::212:7402:2:202	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
2	8.490000	2002:db8::212:7402:2:202	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
3	8.490000	2002:db8::212:7402:2:202	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
4	8.491000	2002:db8::212:7402:2:202	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
5	8.493000	2002:db8::212:7402:2:202	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
6	8.494000	2002:db8::212:7402:2:202	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
7	8.504000	2002:db8::212:7401:1:101	2002:db8::212:7402:2:202	UDP	61	Source port: rrac Destination port: ultraseek-http
8	8.506000	2002:db8::212:7401:1:101	2002:db8::212:7402:2:202	UDP	61	Source port: rrac Destination port: ultraseek-http
9	8.508000	2002:db8::212:7401:1:101	2002:db8::212:7402:2:202	UDP	61	Source port: rrac Destination port: ultraseek-http
10	8.508000	2002:db8::212:7401:1:101	2002:db8::212:7402:2:202	UDP	61	Source port: rrac Destination port: ultraseek-http
11	8.509000	2002:db8::212:7401:1:101	2002:db8::212:7402:2:202	UDP	61	Source port: rrac Destination port: ultraseek-http
12	8.510000	2002:db8::212:7401:1:101	2002:db8::212:7402:2:202	UDP	61	Source port: rrac Destination port: ultraseek-http
13	8.511000	2002:db8::212:7401:1:101	2002:db8::212:7402:2:202	UDP	61	Source port: rrac Destination port: ultraseek-http
14	8.512000	2002:db8::212:7401:1:101	2002:db8::212:7402:2:202	UDP	61	Source port: rrac Destination port: ultraseek-http
15	8.514000	2002:db8::212:7401:1:101	2002:db8::212:7402:2:202	UDP	61	Source port: rrac Destination port: ultraseek-http
16	8.514000	2002:db8::212:7401:1:101	2002:db8::212:7402:2:202	UDP	61	Source port: rrac Destination port: ultraseek-http
17	8.515000	2002:db8::212:7401:1:101	2002:db8::212:7402:2:202	UDP	61	Source port: rrac Destination port: ultraseek-http
18	8.516000	2002:db8::212:7401:1:101	2002:db8::212:7402:2:202	UDP	61	Source port: rrac Destination port: ultraseek-http
19	8.516000	2002:db8::212:7401:1:101	2002:db8::212:7402:2:202	UDP	61	Source port: rrac Destination port: ultraseek-http
20	8.517000	2002:db8::212:7401:1:101	2002:db8::212:7402:2:202	UDP	61	Source port: rrac Destination port: ultraseek-http
21	8.518000	2002:db8::212:7401:1:101	2002:db8::212:7402:2:202	UDP	61	Source port: rrac Destination port: ultraseek-http
22	8.518000	2002:db8::212:7401:1:101	2002:db8::212:7402:2:202	UDP	61	Source port: rrac Destination port: ultraseek-http
23	8.519000	2002:db8::212:7401:1:101	2002:db8::212:7402:2:202	UDP	61	Source port: rrac Destination port: ultraseek-http
24	8.521000	2002:db8::212:7401:1:101	2002:db8::212:7402:2:202	UDP	61	Source port: rrac Destination port: ultraseek-http
25	8.522000	2002:db8::212:7401:1:101	2002:db8::212:7402:2:202	UDP	61	Source port: rrac Destination port: ultraseek-http
26	8.523000	2002:db8::212:7401:1:101	2002:db8::212:7402:2:202	UDP	61	Source port: rrac Destination port: ultraseek-http
27	8.524000	2002:db8::212:7401:1:101	2002:db8::212:7402:2:202	UDP	61	Source port: rrac Destination port: ultraseek-http
28	8.525000	2002:db8::212:7401:1:101	2002:db8::212:7402:2:202	UDP	61	Source port: rrac Destination port: ultraseek-http
29	8.525000	2002:db8::212:7401:1:101	2002:db8::212:7402:2:202	UDP	61	Source port: rrac Destination port: ultraseek-http
30	8.526000	2002:db8::212:7401:1:101	2002:db8::212:7402:2:202	UDP	61	Source port: rrac Destination port: ultraseek-http

Figure 4.40 Malicious “sinkhole-radiolog-UDP.csv”—1 to 30 records.

4.5 Sleep Deprivation Attack Datasets

In this Section, we provide a detailed description of the approach followed to generate a set of malicious datasets by implementing a sleep deprivation attack scenario in the Cooja simulator, as shown in Figure 4.41.

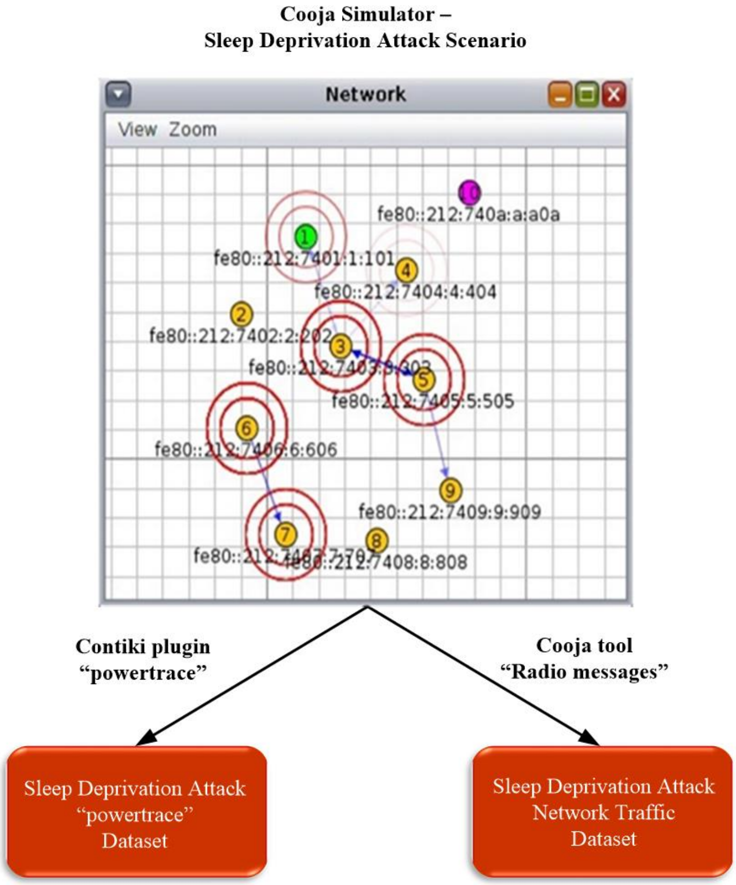
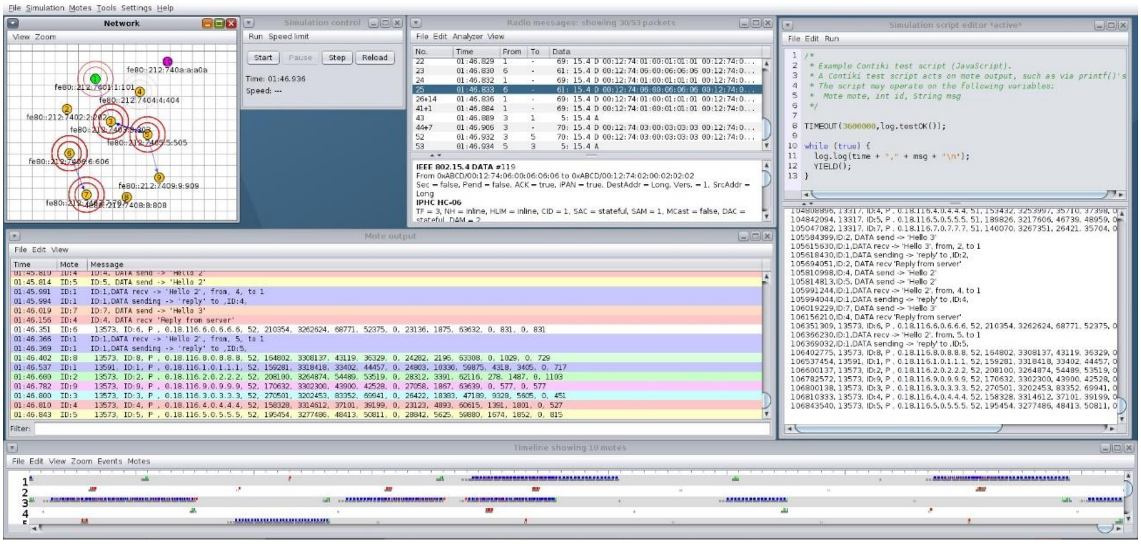


Figure 4.41 Sleep Deprivation Datasets generation by utilising the Cooja simulator.

4.5.1 Sleep Deprivation Attack Scenario – an example

The network topology of the simulated sleep deprivation attack scenario in the Cooja simulator environment consists of 8 yellow (benign) UDP-client motes (i.e., motes 2, 3, 4, 5, 6, 7, 8 and 9), the violet (malicious) UDP-client mote (i.e., mote 10) and the green (benign) UDP-server mote (i.e., mote 1) which is also the target of the attack through mote 4, as, as depicted in Figure 4.41. The simulation duration was set to 60 mins and the motes' outputs were printed out in the respective window (e.g., Mote output) while simulations run, as shown in Figure 4.42. Moreover, the 8 yellow (benign) UDP-client motes were configured to send text messages every 30 seconds, approximately, to the UDP-server mote that was configured to provide a corresponding response. On the other hand, the violet (malicious) UDP-client mote (i.e., mote 10) was compromised with malicious code, as shown in Figure 4.43 and Figure 4.44, to generate high UDP traffic (i.e., a dummy message every 40 ms, approximately) and send it to the target mote which is mote 4. The malicious mote was programmed to start 25 minutes later than the others allowing the network to work properly before the attack. Finally, it is noteworthy to say that similar to the benign network scenario, the UDP protocol was used at the Transport Layer, the IPv6 at the network layer, and the type of motes was the Tmote Sky in the sleep deprivation attack scenario.



```
No.    Time       From  To  Data
22     01:46.829  1     -   69: 15.4 D 00:12:74:01:00:01:01:01 00:12:74:0...
23     01:46.830  6     -   61: 15.4 D 00:12:74:06:00:06:06:06 00:12:74:0...
24     01:46.832  1     -   69: 15.4 D 00:12:74:01:00:01:01:01 00:12:74:0...
25     01:46.833  6     -   61: 15.4 D 00:12:74:06:00:06:06:06 00:12:74:0...
26+14  01:46.836  1     -   69: 15.4 D 00:12:74:01:00:01:01:01 00:12:74:0...
41+1   01:46.884  1     -   69: 15.4 D 00:12:74:01:00:01:01:01 00:12:74:0...
43     01:46.889  3     1    5: 15.4 A
44+7   01:46.906  3     -   70: 15.4 D 00:12:74:03:00:03:03:03 00:12:74:0...
52     01:46.932  3     5   70: 15.4 D 00:12:74:03:00:03:03:03 00:12:74:0...
53     01:46.934  5     3    5: 15.4 A
```

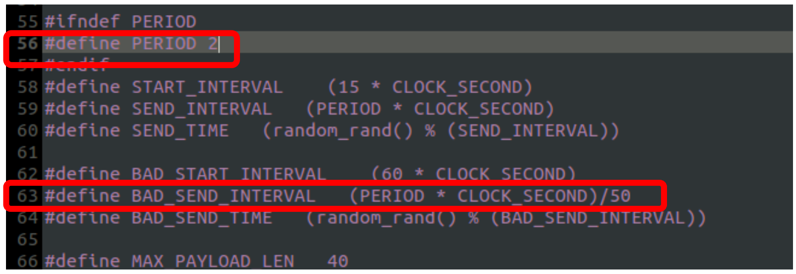
IEEE 802.15.4 DATA #119
From 0xABCD/00:12:74:06:00:06:06:06 to 0xABCD/00:12:74:02:00:02:02:02
Sec = false, Pend = false, ACK = true, iPAN = true, DestAddr = Long, Vers. = 1, SrcAddr = Long
IPHC HC-06
TF = 3, NH = inline, HLIM = inline, CID = 1, SAC = stateful, SAM = 1, MCast = false, DAC = stateful, DAM = 2

```
/*
 * Example Contiki test script (JavaScript).
 * A Contiki test script acts on mote output, such as via printf()'s
 * The script may operate on the following variables:
 *  Mote mote, int id, String msg
 */

TIMEOUT(3600000,log.testOK());

while (true) {
  log.log(time + "," + msg + "\n");
  YIELD();
}
```

```
104808896, 13317, ID:4, P , 0.18.116.4.0.4.4.4, 51, 153432, 3253997, 35710, 37398, 0
104842094, 13317, ID:5, P , 0.18.116.5.0.5.5.5, 51, 189826, 3217606, 46739, 48959, 0
105047082, 13317, ID:7, P , 0.18.116.7.0.7.7.7, 51, 140070, 3267351, 26421, 35704, 0
105584399,ID:2, DATA send -> 'Hello 3'
105615630,ID:1,DATA recv -> 'Hello 3', from, 2, to 1
105618430,ID:1,DATA sending -> 'reply' to ,ID:2,
105694051,ID:2, DATA recv 'Reply from server'
105810998,ID:4, DATA send -> 'Hello 2'
105814813,ID:5, DATA send -> 'Hello 2'
105991244,ID:1,DATA recv -> 'Hello 2', from, 4, to 1
105994044,ID:1,DATA sending -> 'reply' to ,ID:4,
106019229,ID:7, DATA send -> 'Hello 3'
106156210,ID:4, DATA recv 'Reply from server'
106351309, 13573, ID:6, P , 0.18.116.6.0.6.6.6, 52, 210354, 3262624, 68771, 52375, 0
106366230,ID:1,DATA recv -> 'Hello 2', from, 5, to 1
106369032,ID:1,DATA sending -> 'reply' to ,ID:5,
106402775, 13573, ID:8, P , 0.18.116.8.0.8.8.8, 52, 164802, 3308137, 43119, 36329, 0
106537454, 13591, ID:1, P , 0.18.116.1.0.1.1.1, 52, 159281, 3318418, 33402, 44457, 0
106600137, 13573, ID:2, P , 0.18.116.2.0.2.2.2, 52, 208100, 3264874, 54489, 53519, 0
106782572, 13573, ID:9, P , 0.18.116.9.0.9.9.9, 52, 170632, 3302300, 43900, 42528, 0
106800138, 13573, ID:3, P , 0.18.116.3.0.3.3.3, 52, 270501, 3202453, 83352, 69941, 0
106810333, 13573, ID:4, P , 0.18.116.4.0.4.4.4, 52, 158328, 3314612, 37101, 39199, 0
106843540, 13573, ID:5, P , 0.18.116.5.0.5.5.5, 52, 195454, 3277486, 48413, 50811, 0
```

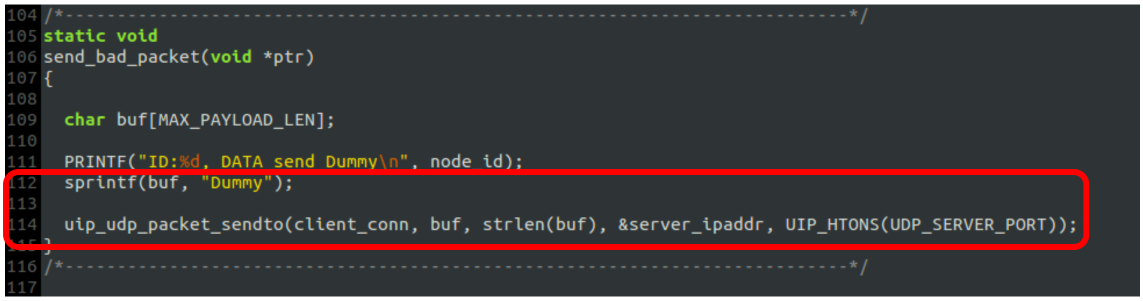
```
Time       Mote  Message
01:45.810  ID:4  ID:4, DATA send -> 'Hello 2'
01:45.814  ID:5  ID:5, DATA send -> 'Hello 2'
01:45.991  ID:1  ID:1,DATA recv -> 'Hello 2', from, 4, to 1
01:45.994  ID:1  ID:1,DATA sending -> 'reply' to ,ID:4,
01:46.019  ID:7  ID:7, DATA send -> 'Hello 3'
01:46.156  ID:4  ID:4, DATA recv 'Reply from server'
01:46.351  ID:6   13573, ID:6, P , 0.18.116.6.0.6.6.6, 52, 210354, 3262624, 68771, 52375, 0, 23136, 1875, 63632, 0, 831, 0, 831
01:46.366  ID:1  ID:1,DATA recv -> 'Hello 2', from, 5, to 1
01:46.369  ID:1  ID:1,DATA sending -> 'reply' to ,ID:5,
01:46.402  ID:8   13573, ID:8, P , 0.18.116.8.0.8.8.8, 52, 164802, 3308137, 43119, 36329, 0, 24282, 2196, 63308, 0, 1029, 0, 729
01:46.537  ID:1   13591, ID:1, P , 0.18.116.1.0.1.1.1, 52, 159281, 3318418, 33402, 44457, 0, 24803, 10330, 59875, 4318, 3405, 0, 717
01:46.600  ID:2   13573, ID:2, P , 0.18.116.2.0.2.2.2, 52, 208100, 3264874, 54489, 53519, 0, 28312, 3391, 62116, 278, 1487, 0, 1103
01:46.782  ID:9   13573, ID:9, P , 0.18.116.9.0.9.9.9, 52, 170632, 3302300, 43900, 42528, 0, 27058, 1867, 63639, 0, 577, 0, 577
01:46.800  ID:3   13573, ID:3, P , 0.18.116.3.0.3.3.3, 52, 270501, 3202453, 83352, 69941, 0, 26422, 18383, 47189, 9328, 5605, 0, 451
01:46.810  ID:4   13573, ID:4, P , 0.18.116.4.0.4.4.4, 52, 158328, 3314612, 37101, 39199, 0, 23123, 4893, 60615, 1391, 1801, 0, 527
01:46.843  ID:5   13573, ID:5, P , 0.18.116.5.0.5.5.5, 52, 195454, 3277486, 48413, 50811, 0, 28842, 5625, 59880, 1674, 1852, 0, 815
```

Figure 4.42 Cooja Simulator — Sleep Deprivation attack scenario — Motes' outputs.



```
#ifndef PERIOD
#define PERIOD 2
#endif

#define START_INTERVAL     (15 * CLOCK_SECOND)
#define SEND_INTERVAL    (PERIOD * CLOCK_SECOND)
#define SEND_TIME    (random_rand() % (SEND_INTERVAL))

#define BAD_START_INTERVAL     (60 * CLOCK_SECOND)
#define BAD_SEND_INTERVAL    (PERIOD * CLOCK_SECOND)/50
#define BAD_SEND_TIME    (random_rand() % (BAD_SEND_INTERVAL))

#define MAX_PAYLOAD_LEN   40
```

Figure 4.43 Malicious code in "udp-client-sleep_depr.c" to generate high UDP traffic (i.e., a dummy message every 40 ms, approximately)



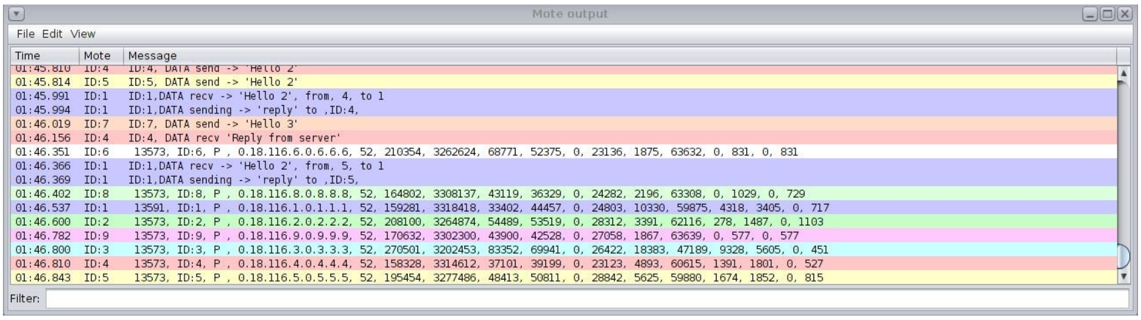
```
/*---------------------------------------------------------------------------*/
static void
send_bad_packet(void *ptr)
{

  char buf[MAX_PAYLOAD_LEN];

  PRINTF("ID:%d, DATA send Dummy\n", node_id);
  sprintf(buf, "Dummy");

  uip_udp_packet_sendto(client_conn, buf, strlen(buf), &server_ipaddr, UIP_HTONS(UDP_SERVER_PORT));
}
/*---------------------------------------------------------------------------*/
```

Figure 4.44 Malicious code in "udp-client-sleep_depr.c" to send out a dummy message

4.5.2 Sleep Deprivation Attack "powertrace" Dataset

4.5.2.1 Sleep Deprivation Attack "powertrace" Dataset – Generation Process

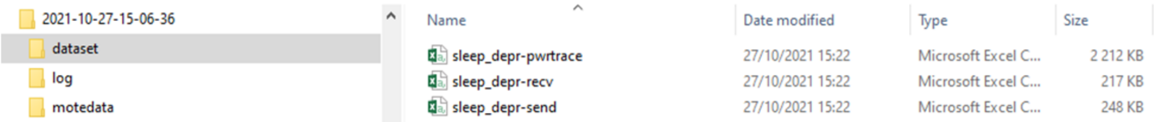
The approach followed for the "powertrace" dataset generation from the sleep deprivation attack scenario was similar to the approach followed for the "powertrace" dataset generation from the benign IoT network scenario in Section 3.3.1. In addition, the "powertrace" plugin was similarly enabled for collecting "powertrace" related features, summarised in Table 3, from the motes of the attack scenario every two seconds. In Figure 4.45, the depicted mote output window displays the captured "powertrace" information every two seconds and also the messages sent and received by each mote during the simulation time (60 mins).



```
Time       Mote   Message
01:45.810  ID:4   ID:4, DATA send -> 'Hello 2'
01:45.814  ID:5   ID:5, DATA send -> 'Hello 2'
01:45.991  ID:1   ID:1,DATA recv -> 'Hello 2', from, 4, to 1
01:45.994  ID:1   ID:1,DATA sending -> 'reply' to ,ID:4,
01:46.019  ID:7   ID:7, DATA send -> 'Hello 3'
01:46.156  ID:4   ID:4, DATA recv 'Reply from server'
01:46.351  ID:6    13573, ID:6, P , 0.18.116.6.0.6.6.6, 52, 210354, 3262624, 68771, 52375, 0, 23136, 1875, 63632, 0, 831, 0, 831
01:46.366  ID:1   ID:1,DATA recv -> 'Hello 2', from, 5, to 1
01:46.369  ID:1   ID:1,DATA sending -> 'reply' to ,ID:5,
01:46.402  ID:8    13573, ID:8, P , 0.18.116.8.0.8.8.8, 52, 164802, 3308137, 43119, 36329, 0, 24282, 2196, 63308, 0, 1029, 0, 729
01:46.537  ID:1    13591, ID:1, P , 0.18.116.1.0.1.1.1, 52, 159281, 3318418, 33402, 44457, 0, 24803, 10330, 59875, 4318, 3405, 0, 717
01:46.600  ID:2    13573, ID:2, P , 0.18.116.2.0.2.2.2, 52, 208100, 3264874, 54489, 53519, 0, 28312, 3391, 62116, 278, 1487, 0, 1103
01:46.782  ID:9    13573, ID:9, P , 0.18.116.9.0.9.9.9, 52, 170632, 3302300, 43900, 42528, 0, 27058, 1867, 63639, 0, 577, 0, 577
01:46.800  ID:3    13573, ID:3, P , 0.18.116.3.0.3.3.3, 52, 270501, 3202453, 83352, 69941, 0, 26422, 18383, 47189, 9328, 5605, 0, 451
01:46.810  ID:4    13573, ID:4, P , 0.18.116.4.0.4.4.4, 52, 158328, 3314612, 37101, 39199, 0, 23123, 4893, 60615, 1391, 1801, 0, 527
01:46.843  ID:5    13573, ID:5, P , 0.18.116.5.0.5.5.5, 52, 195454, 3277486, 48413, 50811, 0, 28842, 5625, 59880, 1674, 1852, 0, 815
```

Figure 4.45 Cooja Simulator— Sleep deprivation attack scenario — Mote output window.

When the timeout occurred, the simulation stopped, and all the captured information and prints were stored in the "COOJA.testlog" file. Afterwards, the "IoT_Simul.sh" file, described in Section 3.3.1, created a) a new root folder named as "2021-10-27-15-06-36", and b) the "log" folder, inside the "2021-10-27-15-06-36" folder, where the "COOJA.testlog" file was copied from the ".../cooja/build" folder located in the Cooja Simulator. Then, the "IoT_Simul.sh" file following the same process, as described in Section 3.3.1, extracted the required "powertrace" information from the "COOJA.testlog" file and saved it in the "sleep_depr-pwrtrace.csv" file in the "dataset" folder that was also created by the batch file inside the "2021-10-27-15-06-36" folder, as shown below in the left part of Figure 4.46. In the "dataset" folder, apart from the "sleep_depr-pwrtrace.csv" file, the "IoT_Simul.sh" file generated two more files (i.e., "sleep_depr-recv.csv" and "sleep_depr-send.csv"), following the same process as in Section 3.3.1. The "sleep_depr-recv.csv" file and the "sleep_depr-send.csv" file include the "received" and "sent" messages printed by the motes, respectively.



2021-10-27-15-06-36
dataset
log
motedata

Name	Date modified	Type	Size
sleep_depr-pwrtrace	27/10/2021 15:22	Microsoft Excel C...	2 212 KB
sleep_depr-recv	27/10/2021 15:22	Microsoft Excel C...	217 KB
sleep_depr-send	27/10/2021 15:22	Microsoft Excel C...	248 KB

Figure 4.46 Location of the generated "sleep_depr-pwrtrace.csv", "sleep_depr-recv.csv", and "sleep_depr-send.csv" files by the "IoT_Simul.sh" bash file.

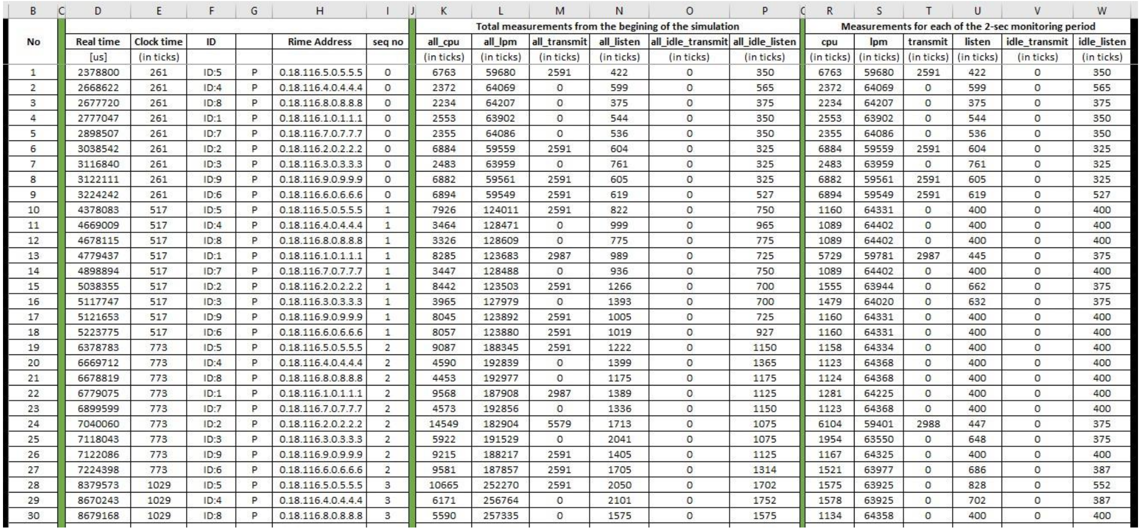
Finally, similar to the benign "powertrace" dataset generation approach in Section 3.3.1, the "IoT_Simul.sh" file extracted the information related to each mote from the "sleep_depr-pwrtrace.csv" file and generated one csv file for each mote with the corresponding information from the "sleep_depr-pwrtrace.csv" file. The generated ten csv files (i.e., "sleep_depr-mote1.csv", ..., "sleep_depr-mote10.csv") were stored in the "motedata" folder, created also by the "IoT_Simul.sh" file, as shown in the left part of Figure 4.46.

4.5.2.2 Sleep Deprivation Attack "powertrace" Dataset – Generated Results

The sleep deprivation attack "powertrace" dataset consists of the following csv files: "sleep_depr-pwrtrace.csv", "sleep_depr-mote1.csv", "sleep_depr-mote2.csv", "sleep_depr-mote3.csv", "sleep_depr-mote4.csv", "sleep_depr-mote5.csv", "sleep_depr-mote6.csv", "sleep_depr-mote7.csv", "sleep_depr-mote8.csv", "sleep_depr-mote9.csv", and "sleep_depr-mote10.csv". In this Section, we present sets of records from the "sleep_depr-pwrtrace.csv", and in Appendix 1 we present sets of records from "sleep_depr-mote1.csv", "sleep_depr-mote6.csv" and "sleep_depr-mote10.csv" files.

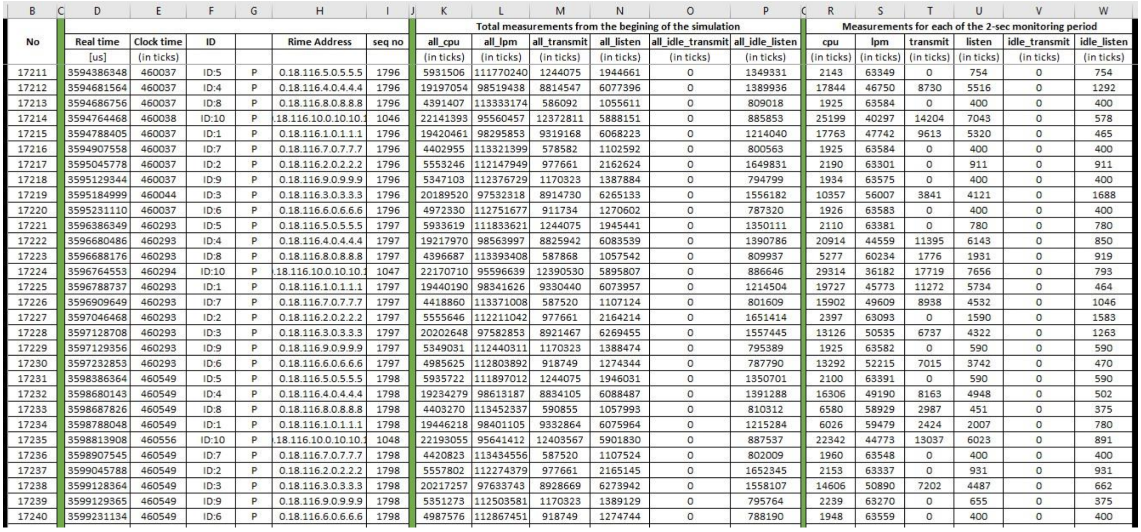
4.5.2.2.1 "sleep_depr-pwrtrace.csv"

The generated malicious "sleep_depr-pwrtrace.csv" file consists of 17,240 records and its first 30 records (i.e., 1–30) and its last 30 records (17,211–17,240) are depicted in Figure 4.47 and Figure 4.48, respectively.



B	D	E	F	G	H	I	K	L	M	N	O	P	R	S	T	U	V	W
No							Total measurements from the begining of the simulation						Measurements for each of the 2-sec monitoring period					
	Real time	Clock time	ID		Rime Address	seq no	all_cpu	all_lpm	all_transmit	all_listen	all_idle_transmit	all_idle_listen	cpu	lpm	transmit	listen	idle_transmit	idle_listen
	[us]	(in ticks)					(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)
1	2378800	261	ID:5	P	0.18.116.5.0.5.5.5	0	6763	59680	2591	422	0	350	6763	59680	2591	422	0	350
2	2668622	261	ID:4	P	0.18.116.4.0.4.4.4	0	2372	64069	0	599	0	565	2372	64069	0	599	0	565
3	2677720	261	ID:8	P	0.18.116.8.0.8.8.8	0	2234	64207	0	375	0	375	2234	64207	0	375	0	375
4	2777047	261	ID:1	P	0.18.116.1.0.1.1.1	0	2553	63902	0	544	0	350	2553	63902	0	544	0	350
5	2898507	261	ID:7	P	0.18.116.7.0.7.7.7	0	2355	64086	0	536	0	350	2355	64086	0	536	0	350
6	3038542	261	ID:2	P	0.18.116.2.0.2.2.2	0	6884	59559	2591	604	0	325	6884	59559	2591	604	0	325
7	3116840	261	ID:3	P	0.18.116.3.0.3.3.3	0	2483	63959	0	761	0	325	2483	63959	0	761	0	325
8	3122111	261	ID:9	P	0.18.116.9.0.9.9.9	0	6882	59561	2591	605	0	325	6882	59561	2591	605	0	325
9	3224242	261	ID:6	P	0.18.116.6.0.6.6.6	0	6894	59549	2591	619	0	527	6894	59549	2591	619	0	527
10	4378083	517	ID:5	P	0.18.116.5.0.5.5.5	1	7926	124011	2591	822	0	750	1160	64331	0	400	0	400
11	4669009	517	ID:4	P	0.18.116.4.0.4.4.4	1	3464	128471	0	999	0	965	1089	64402	0	400	0	400
12	4678115	517	ID:8	P	0.18.116.8.0.8.8.8	1	3326	128609	0	775	0	775	1089	64402	0	400	0	400
13	4779437	517	ID:1	P	0.18.116.1.0.1.1.1	1	8285	123683	2987	989	0	725	5729	59781	2987	445	0	375
14	4898894	517	ID:7	P	0.18.116.7.0.7.7.7	1	3447	128488	0	936	0	750	1089	64402	0	400	0	400
15	5038355	517	ID:2	P	0.18.116.2.0.2.2.2	1	8442	123503	2591	1266	0	700	1555	63944	0	662	0	375
16	5117747	517	ID:3	P	0.18.116.3.0.3.3.3	1	3965	127979	0	1393	0	700	1479	64020	0	632	0	375
17	5121653	517	ID:9	P	0.18.116.9.0.9.9.9	1	8045	123892	2591	1005	0	725	1160	64331	0	400	0	400
18	5223775	517	ID:6	P	0.18.116.6.0.6.6.6	1	8057	123880	2591	1019	0	927	1160	64331	0	400	0	400
19	6378783	773	ID:5	P	0.18.116.5.0.5.5.5	2	9087	188345	2591	1222	0	1150	1158	64334	0	400	0	400
20	6669712	773	ID:4	P	0.18.116.4.0.4.4.4	2	4590	192839	0	1399	0	1365	1123	64368	0	400	0	400
21	6678819	773	ID:8	P	0.18.116.8.0.8.8.8	2	4453	192977	0	1175	0	1175	1124	64368	0	400	0	400
22	6779075	773	ID:1	P	0.18.116.1.0.1.1.1	2	9568	187908	2987	1389	0	1125	1281	64225	0	400	0	400
23	6899599	773	ID:7	P	0.18.116.7.0.7.7.7	2	4573	192856	0	1336	0	1150	1123	64368	0	400	0	400
24	7040060	773	ID:2	P	0.18.116.2.0.2.2.2	2	14549	182904	5579	1713	0	1075	6104	59401	2988	447	0	375
25	7118043	773	ID:3	P	0.18.116.3.0.3.3.3	2	5922	191529	0	2041	0	1075	1954	63550	0	648	0	375
26	7122086	773	ID:9	P	0.18.116.9.0.9.9.9	2	9215	188217	2591	1405	0	1125	1167	64325	0	400	0	400
27	7224398	773	ID:6	P	0.18.116.6.0.6.6.6	2	9581	187857	2591	1705	0	1314	1521	63977	0	686	0	387
28	8379573	1029	ID:5	P	0.18.116.5.0.5.5.5	3	10665	252270	2591	2050	0	1702	1575	63925	0	828	0	552
29	8670243	1029	ID:4	P	0.18.116.4.0.4.4.4	3	6171	256764	0	2101	0	1752	1578	63925	0	702	0	387
30	8679168	1029	ID:8	P	0.18.116.8.0.8.8.8	3	5590	257335	0	1575	0	1575	1134	64358	0	400	0	400

Figure 4.47 Malicious "sleep_depr-pwrtrace.csv"—1 to 30 records.



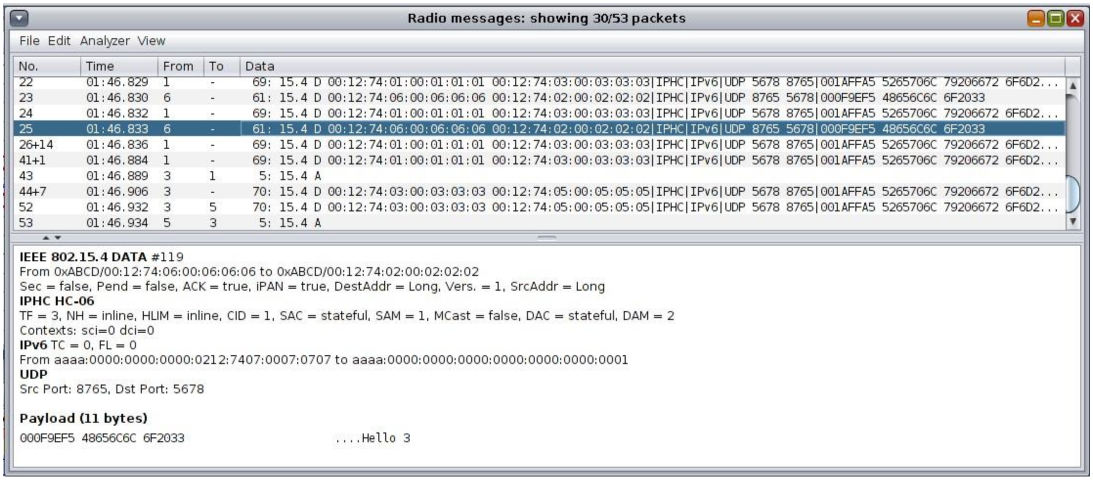
B	D	E	F	G	H	I	K	L	M	N	O	P	R	S	T	U	V	W
No							Total measurements from the begining of the simulation						Measurements for each of the 2-sec monitoring period					
	Real time	Clock time	ID		Rime Address	seq no	all_cpu	all_lpm	all_transmit	all_listen	all_idle_transmit	all_idle_listen	cpu	lpm	transmit	listen	idle_transmit	idle_listen
	[us]	(in ticks)					(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)
17211	3594386348	460037	ID:5	P	0.18.116.5.0.5.5.5	1796	5931506	111770240	1244075	1944661	0	1349331	2143	63349	0	754	0	754
17212	3594681564	460037	ID:4	P	0.18.116.4.0.4.4.4	1796	19197054	98519438	8814547	6077396	0	1389936	17844	46750	8730	5516	0	1292
17213	3594686756	460037	ID:8	P	0.18.116.8.0.8.8.8	1796	4391407	113333174	586092	1055611	0	809018	1925	63584	0	400	0	400
17214	3594764468	460038	ID:10	P	.18.116.10.0.10.10.1	1046	22141393	95560457	12372811	5888151	0	885853	25199	40297	14204	7043	0	578
17215	3594788405	460037	ID:1	P	0.18.116.1.0.1.1.1	1796	19420461	98295853	9319168	6068223	0	1214040	17763	47742	9613	5320	0	465
17216	3594907558	460037	ID:7	P	0.18.116.7.0.7.7.7	1796	4402955	113321399	578582	1102592	0	800563	1925	63584	0	400	0	400
17217	3595045778	460037	ID:2	P	0.18.116.2.0.2.2.2	1796	5553246	112147949	977661	2162624	0	1649831	2190	63301	0	911	0	911
17218	3595129344	460037	ID:9	P	0.18.116.9.0.9.9.9	1796	5347103	112376729	1170323	1387884	0	794799	1934	63575	0	400	0	400
17219	3595184999	460044	ID:3	P	0.18.116.3.0.3.3.3	1796	20189520	97532318	8914730	6265133	0	1556182	10357	56007	3841	4121	0	1688
17220	3595231110	460037	ID:6	P	0.18.116.6.0.6.6.6	1796	4972330	112751677	911734	1270602	0	787320	1926	63583	0	400	0	400
17221	3596386349	460293	ID:5	P	0.18.116.5.0.5.5.5	1797	5933619	111833621	1244075	1945441	0	1350111	2110	63381	0	780	0	780
17222	3596680486	460293	ID:4	P	0.18.116.4.0.4.4.4	1797	19217970	98563997	8825942	6083539	0	1390786	20914	44559	11395	6143	0	850
17223	3596688176	460293	ID:8	P	0.18.116.8.0.8.8.8	1797	4396687	113393408	587868	1057542	0	809937	5277	60234	1776	1931	0	919
17224	3596764553	460294	ID:10	P	.18.116.10.0.10.10.1	1047	22170710	95596639	12390530	5895807	0	886646	29314	36182	17719	7656	0	793
17225	3596788737	460293	ID:1	P	0.18.116.1.0.1.1.1	1797	19440190	98341626	9330440	6073957	0	1214504	19727	45773	11272	5734	0	464
17226	3596909649	460293	ID:7	P	0.18.116.7.0.7.7.7	1797	4418860	113371008	587520	1107124	0	801609	15902	49609	8938	4532	0	1046
17227	3597046468	460293	ID:2	P	0.18.116.2.0.2.2.2	1797	5555646	112211042	977661	2164214	0	1651414	2397	63093	0	1590	0	1583
17228	3597128708	460293	ID:3	P	0.18.116.3.0.3.3.3	1797	20202648	97582853	8921467	6269455	0	1557445	13126	50535	6737	4322	0	1263
17229	3597129356	460293	ID:9	P	0.18.116.9.0.9.9.9	1797	5349031	112440311	1170323	1388474	0	795389	1925	63582	0	590	0	590
17230	3597232853	460293	ID:6	P	0.18.116.6.0.6.6.6	1797	4985625	112803892	918749	1274344	0	787790	13292	52215	7015	3742	0	470
17231	3598386364	460549	ID:5	P	0.18.116.5.0.5.5.5	1798	5935722	111897012	1244075	1946031	0	1350701	2100	63391	0	590	0	590
17232	3598680143	460549	ID:4	P	0.18.116.4.0.4.4.4	1798	19234279	98613187	8834105	6088487	0	1391288	16306	49190	8163	4948	0	502
17233	3598687826	460549	ID:8	P	0.18.116.8.0.8.8.8	1798	4403270	113452337	590855	1057993	0	810312	6580	58929	2987	451	0	375
17234	3598788048	460549	ID:1	P	0.18.116.1.0.1.1.1	1798	19446218	98401105	9332864	6075964	0	1215284	6026	59479	2424	2007	0	780
17235	3598813908	460556	ID:10	P	.18.116.10.0.10.10.1	1048	22193055	95641412	12403567	5901830	0	887537	22342	44773	13037	6023	0	891
17236	3598907545	460549	ID:7	P	0.18.116.7.0.7.7.7	1798	4420823	113434556	587520	1107524	0	802009	1960	63548	0	400	0	400
17237	3599045788	460549	ID:2	P	0.18.116.2.0.2.2.2	1798	5557802	112274379	977661	2165145	0	1652345	2153	63337	0	931	0	931
17238	3599128364	460549	ID:3	P	0.18.116.3.0.3.3.3	1798	20217257	97633743	8928669	6273942	0	1558107	14606	50890	7202	4487	0	662
17239	3599129365	460549	ID:9	P	0.18.116.9.0.9.9.9	1798	5351273	112503581	1170323	1389129	0	795764	2239	63270	0	655	0	375
17240	3599231134	460549	ID:6	P	0.18.116.6.0.6.6.6	1798	4987576	112867451	918749	1274744	0	788190	1948	63559	0	400	0	400

Figure 4.48 Malicious "sleep_depr-pwrtrace.csv"—17,211 to 17,240 records.

4.5.3 Sleep Deprivation Attack Network Traffic Dataset

4.5.3.1 Sleep Deprivation Attack Network Traffic Dataset – Generation Process

The approach followed for the network traffic dataset generation from the sleep deprivation attack scenario was similar to the approach followed for the network traffic dataset generation from the benign IoT network scenario in Section 3.4.1. The “Radio messages” tool, provided by the Cooja simulator, was similarly used for collecting data related to the corresponding network traffic features (e.g., source/destination IPv6 address, packet size, and protocol) from the network of the attack scenario. During the simulation, the network traffic information was being shown in the top part of the “Radio messages” output window as depicted in the top part of Figure 4.49.

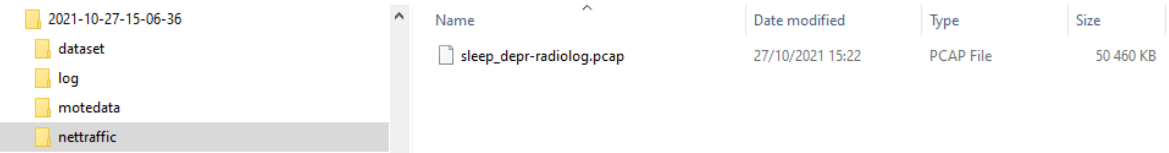


```
No.      Time       From  To   Data
22       01:46.829  1     -    69: 15.4 D 00:12:74:01:00:01:01:01 00:12:74:03:00:03:03:03|IPHC|IPv6|UDP 5678 8765|001AFFA5 5265706C 79206672 6F6D2...
23       01:46.830  6     -    61: 15.4 D 00:12:74:06:00:06:06:06 00:12:74:02:00:02:02:02|IPHC|IPv6|UDP 8765 5678|000F9EF5 48656C6C 6F2033
24       01:46.832  1     -    69: 15.4 D 00:12:74:01:00:01:01:01 00:12:74:03:00:03:03:03|IPHC|IPv6|UDP 5678 8765|001AFFA5 5265706C 79206672 6F6D2...
25       01:46.833  6     -    61: 15.4 D 00:12:74:06:00:06:06:06 00:12:74:02:00:02:02:02|IPHC|IPv6|UDP 8765 5678|000F9EF5 48656C6C 6F2033
26+14    01:46.836  1     -    69: 15.4 D 00:12:74:01:00:01:01:01 00:12:74:03:00:03:03:03|IPHC|IPv6|UDP 5678 8765|001AFFA5 5265706C 79206672 6F6D2...
41+1     01:46.884  1     -    69: 15.4 D 00:12:74:01:00:01:01:01 00:12:74:03:00:03:03:03|IPHC|IPv6|UDP 5678 8765|001AFFA5 5265706C 79206672 6F6D2...
43       01:46.889  3     1     5: 15.4 A
44+7     01:46.906  3     -    70: 15.4 D 00:12:74:03:00:03:03:03 00:12:74:05:00:05:05:05|IPHC|IPv6|UDP 5678 8765|001AFFA5 5265706C 79206672 6F6D2...
52       01:46.932  3     5    70: 15.4 D 00:12:74:03:00:03:03:03 00:12:74:05:00:05:05:05|IPHC|IPv6|UDP 5678 8765|001AFFA5 5265706C 79206672 6F6D2...
53       01:46.934  5     3     5: 15.4 A
```

```
IEEE 802.15.4 DATA #119
From 0xABCD/00:12:74:06:00:06:06:06 to 0xABCD/00:12:74:02:00:02:02:02
Sec = false, Pend = false, ACK = true, iPAN = true, DestAddr = Long, Vers. = 1, SrcAddr = Long
IPHC HC-06
TF = 3, NH = inline, HLIM = inline, CID = 1, SAC = stateful, SAM = 1, MCast = false, DAC = stateful, DAM = 2
Contexts: sci=0 dci=0
IPv6 TC = 0, FL = 0
From aaaa:0000:0000:0000:0212:7407:0007:0707 to aaaa:0000:0000:0000:0000:0000:0000:0001
UDP
Src Port: 8765, Dst Port: 5678

Payload (11 bytes)
000F9EF5 48656C6C 6F2033                    ....Hello 3
```

Figure 4.49 Network traffic information from the sleep deprivation attack scenario in the “Radio messages” output window.

When the simulation stopped, the generated pcap file was saved as “radiolog.pcap” within the “…/cooja/build” folder. Afterwards, the “IoT_Simul.sh” file, described in Section 3.4.1, created a) a new root folder named as “2021-10-27-15-06-36”, and b) the “nettraffic” folder, inside the “2021-10-27-15-06-36” folder, where the “radiolog.pcap” file, copied from the “…/cooja/build” folder located in the Cooja Simulator, was saved as “sleep_depr-radiolog.pcap”. The “nettraffic” folder inside the root folder “2021-10-27-15-06-36” and the “sleep_depr-radiolog.pcap” file in the “nettraffic” folder are shown in Figure 4.50.

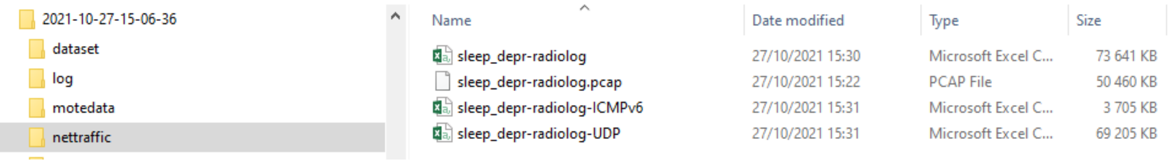


2021-10-27-15-06-36
- dataset
- log
- motedata
- nettraffic

Name	Date modified	Type	Size
sleep_depr-radiolog.pcap	27/10/2021 15:22	PCAP File	50 460 KB

Figure 4.50 The “nettraffic” folder inside the root folder “2021-10-27-15-06-36” and the “sleep_depr-radiolog.pcap” file.

Then, following the same process, as described in Section 3.4.1, we used Wireshark to extract the stored network traffic information from the “sleep_depr-radiolog.pcap” file to the “sleep_depr-radiolog.csv” file stored in the “nettraffic” folder as shown in Figure 4.51.



2021-10-27-15-06-36
- dataset
- log
- motedata
- nettraffic

Name	Date modified	Type	Size
sleep_depr-radiolog	27/10/2021 15:30	Microsoft Excel C...	73 641 KB
sleep_depr-radiolog.pcap	27/10/2021 15:22	PCAP File	50 460 KB
sleep_depr-radiolog-ICMPv6	27/10/2021 15:31	Microsoft Excel C...	3 705 KB
sleep_depr-radiolog-UDP	27/10/2021 15:31	Microsoft Excel C...	69 205 KB

Figure 4.51 The “nettraffic” folder inside the root folder “2021-10-27-15-06-36”and its included files.

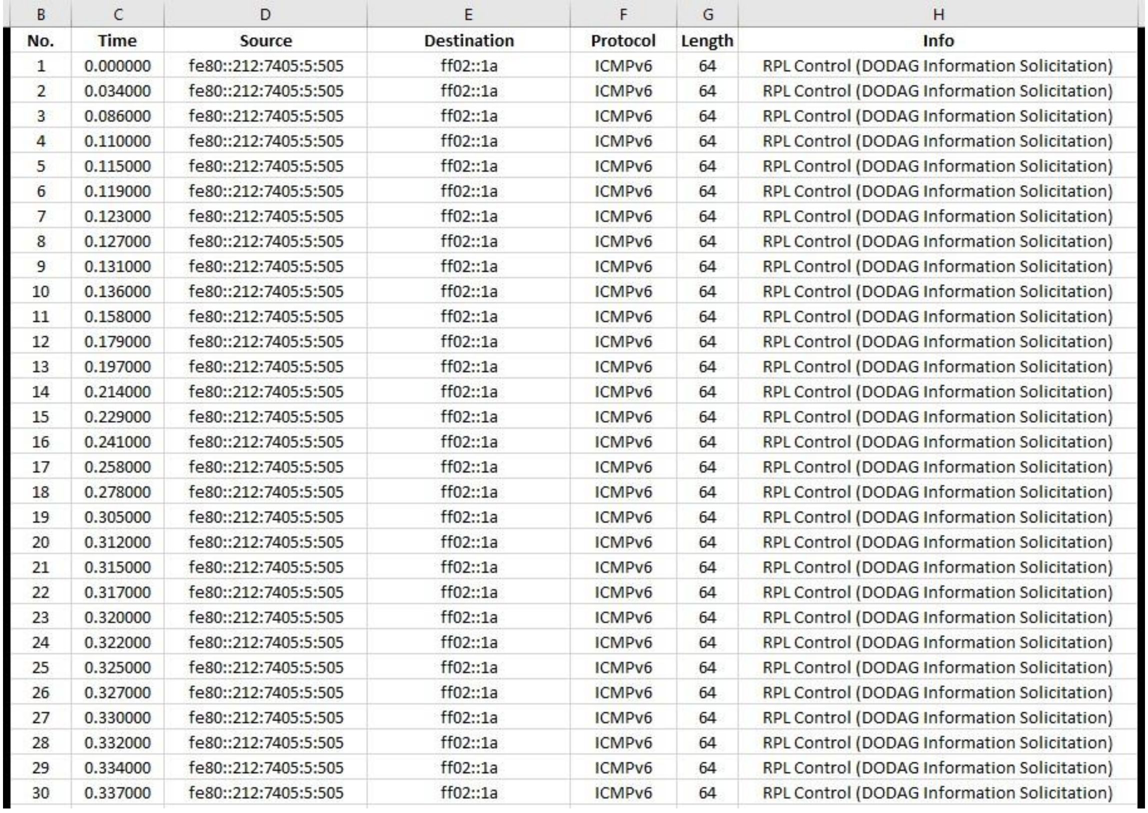
In the “nettraffic” folder, apart from the “sleep_depr-radiolog.csv” file, we also used Wireshark, following the same process as in Section 3.4.1, to generate two more files (i.e., “sleep-depr-radiolog-ICMPv6.csv” and “sleep_depr-radiolog-UDP.csv”) from the “sleep_depr-radiolog.pcap” file.

4.5.3.2 Sleep Deprivation Attack Network Traffic Dataset – Generated Results

The sleep deprivation attack network traffic dataset consists of the following csv files which are located in the “nettraffic” folder as described in Section 4.4.3.1: “sleep_depr-radiolog.csv”, “sleep-depr-radiolog-ICMPv6.csv”, and “sleep_depr-radiolog-UDP.csv” files. In this Section, we present sets of records from these files.

4.5.3.2.1 “sleep_depr-radiolog.csv”

The generated malicious “sleep_depr-radiolog.csv” file consists of 571,079 records and its first 30 records (i.e., 1–30) are depicted below in Figure 4.52.

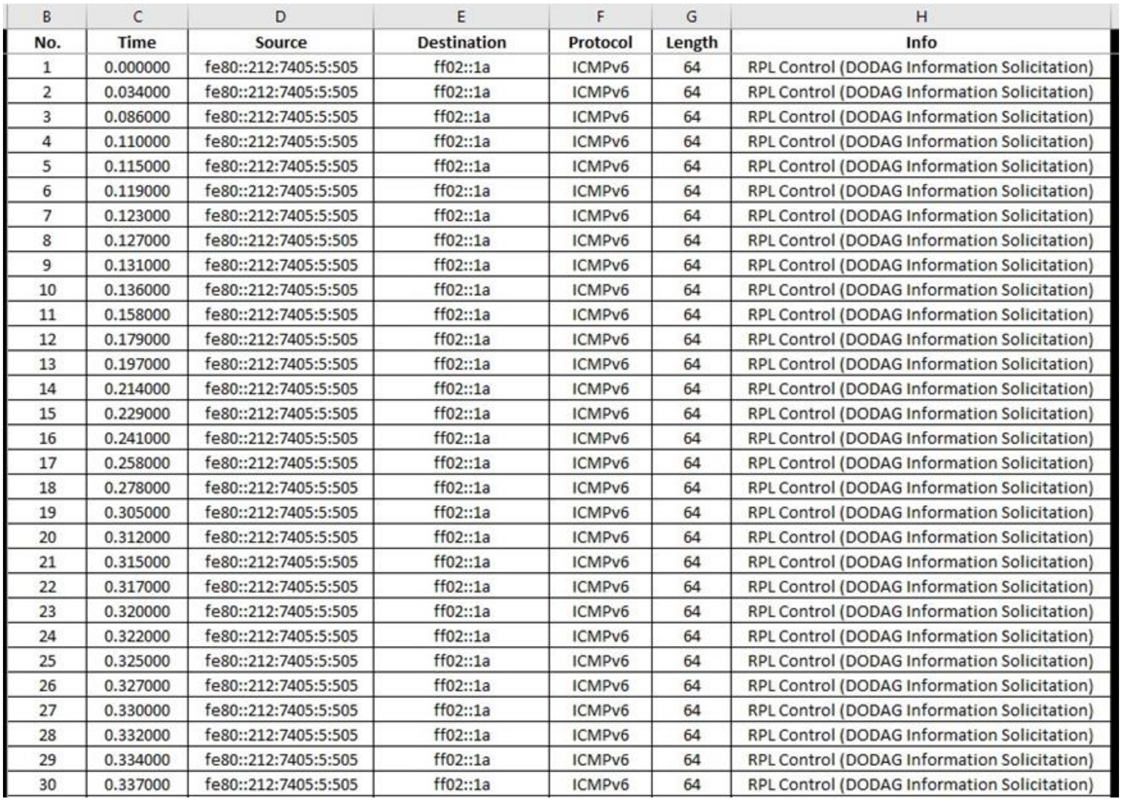


B	C	D	E	F	G	H
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
2	0.034000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
3	0.086000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
4	0.110000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
5	0.115000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
6	0.119000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
7	0.123000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
8	0.127000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
9	0.131000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
10	0.136000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
11	0.158000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
12	0.179000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
13	0.197000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
14	0.214000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
15	0.229000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
16	0.241000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
17	0.258000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
18	0.278000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
19	0.305000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
20	0.312000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
21	0.315000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
22	0.317000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
23	0.320000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
24	0.322000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
25	0.325000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
26	0.327000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
27	0.330000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
28	0.332000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
29	0.334000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
30	0.337000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)

Figure 4.52 Malicious “sleep_depr-radiolog.csv”—1 to 30 records.

4.5.3.2.2 “sleep-depr-radiolog-ICMPv6.csv”

The generated malicious “sleep_depr-radiolog-ICMPv6.csv” file consists of 30,338 records and its first 30 records (i.e., 1–30) are depicted below in Figure 4.53.

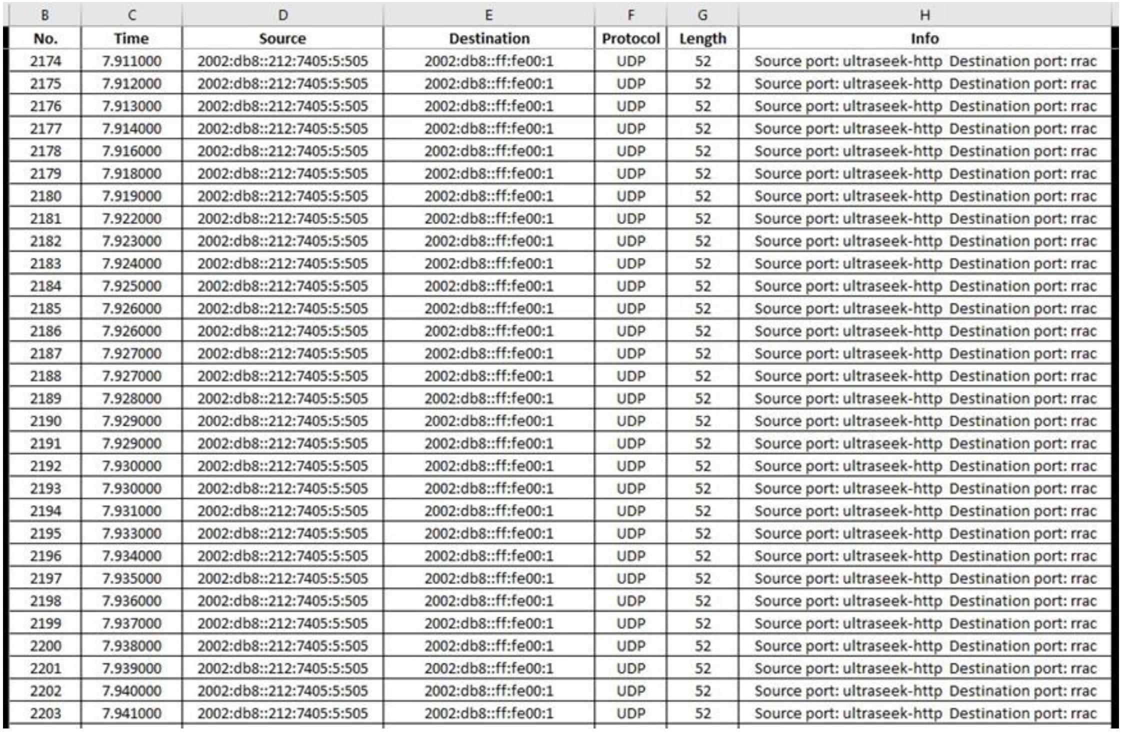


B	C	D	E	F	G	H
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
2	0.034000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
3	0.086000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
4	0.110000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
5	0.115000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
6	0.119000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
7	0.123000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
8	0.127000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
9	0.131000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
10	0.136000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
11	0.158000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
12	0.179000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
13	0.197000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
14	0.214000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
15	0.229000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
16	0.241000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
17	0.258000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
18	0.278000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
19	0.305000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
20	0.312000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
21	0.315000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
22	0.317000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
23	0.320000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
24	0.322000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
25	0.325000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
26	0.327000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
27	0.330000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
28	0.332000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
29	0.334000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
30	0.337000	fe80::212:7405:5:505	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)

Figure 4.53 Malicious "sleep_depr-radiolog-ICMPv6.csv"—1 to 30 records.

4.5.3.2.3 "sleep_depr-radiolog-UDP.csv"

The generated malicious "sleep_depr-radiolog-UDP.csv" file consists of 526,799 records and its first 30 records (i.e., 1–30) are depicted below in Figure 4.54.



B	C	D	E	F	G	H
No.	Time	Source	Destination	Protocol	Length	Info
2174	7.911000	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
2175	7.912000	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
2176	7.913000	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
2177	7.914000	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
2178	7.916000	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
2179	7.918000	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
2180	7.919000	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
2181	7.922000	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
2182	7.923000	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
2183	7.924000	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
2184	7.925000	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
2185	7.926000	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
2186	7.926000	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
2187	7.927000	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
2188	7.927000	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
2189	7.928000	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
2190	7.929000	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
2191	7.929000	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
2192	7.930000	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
2193	7.930000	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
2194	7.931000	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
2195	7.933000	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
2196	7.934000	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
2197	7.935000	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
2198	7.936000	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
2199	7.937000	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
2200	7.938000	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
2201	7.939000	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
2202	7.940000	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac
2203	7.941000	2002:db8::212:7405:5:505	2002:db8::ff:fe00:1	UDP	52	Source port: ultraseek-http Destination port: rrac

Figure 4.54 Malicious "sleep_depr-radiolog-UDP.csv"—1 to 30 records.

4.6 Summary

This Chapter provided a detailed description of the approach proposed to generate a set of malicious datasets from the following attack scenarios implemented in the Cooja simulator: i) **UDP flooding attack**, ii) **blackhole attack**, iii) **sinkhole attack**, and iv) **sleep deprivation attack**. Similar to the approach followed for the generation of the benign datasets in Section 3, the IoT-specific information from the simulated attack scenarios was captured from the Contiki plugin "powertrace" and the Cooja tool "Radio messages" in order to generate the corresponding "powertrace" and network traffic datasets for each of the simulated attack scenarios within csv files. The structure of the generated malicious IoT datasets from the above mentioned four attack scenarios is shown below in Figure 4.55.

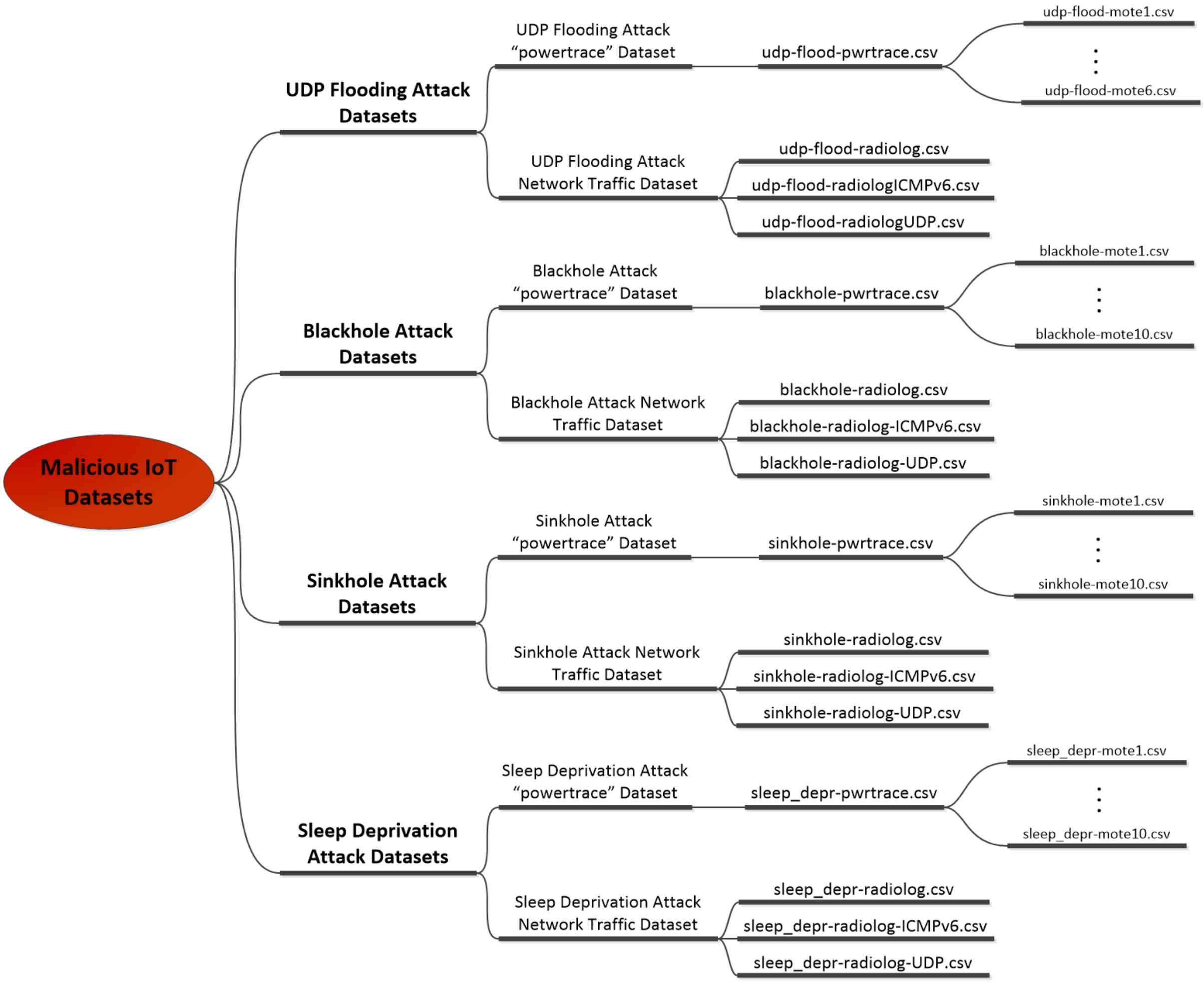

Malicious IoT Datasets
UDP Flooding Attack Datasets
UDP Flooding Attack "powertrace" Dataset
udp-flood-pwrtrace.csv
udp-flood-mote1.csv
⋮
udp-flood-mote6.csv
UDP Flooding Attack Network Traffic Dataset
udp-flood-radiolog.csv
udp-flood-radiologICMPv6.csv
udp-flood-radiologUDP.csv
Blackhole Attack Datasets
Blackhole Attack "powertrace" Dataset
blackhole-pwrtrace.csv
blackhole-mote1.csv
⋮
blackhole-mote10.csv
Blackhole Attack Network Traffic Dataset
blackhole-radiolog.csv
blackhole-radiolog-ICMPv6.csv
blackhole-radiolog-UDP.csv
Sinkhole Attack Datasets
Sinkhole Attack "powertrace" Dataset
sinkhole-pwrtrace.csv
sinkhole-mote1.csv
⋮
sinkhole-mote10.csv
Sinkhole Attack Network Traffic Dataset
sinkhole-radiolog.csv
sinkhole-radiolog-ICMPv6.csv
sinkhole-radiolog-UDP.csv
Sleep Deprivation Attack Datasets
Sleep Deprivation Attack "powertrace" Dataset
sleep_depr-pwrtrace.csv
sleep_depr-mote1.csv
⋮
sleep_depr-mote10.csv
Sleep Deprivation Attack Network Traffic Dataset
sleep_depr-radiolog.csv
sleep_depr-radiolog-ICMPv6.csv
sleep_depr-radiolog-UDP.csv

Figure 4.55 Generated Malicious IoT Datasets Structure

In principle, the proposed approach in this Chapter can be extended for generating malicious IoT datasets from ***j*** different scenarios of ***i*** different attack types, where each attack scenario, implemented in the Cooja simulator, may include ***n*** different motes. The generic structure of malicious IoT datasets generated according to the proposed approach is shown in Figure 4.56.

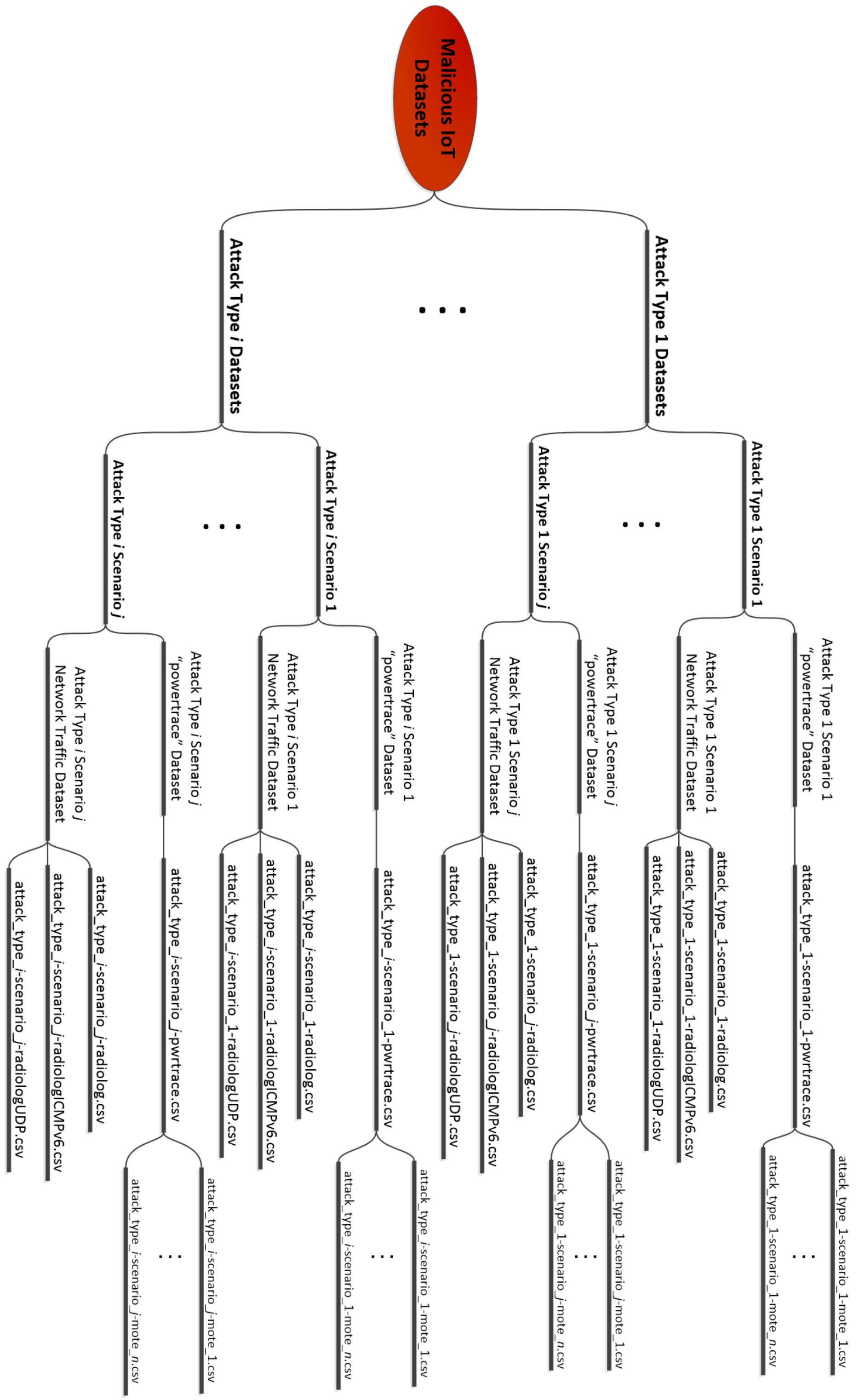

Malicious IoT Datasets
Attack Type 1 Datasets
. . .
Attack Type i Datasets
Attack Type 1 Scenario 1
. . .
Attack Type 1 Scenario j
Attack Type i Scenario 1
. . .
Attack Type i Scenario j
Attack Type 1 Scenario 1 "powertrace" Dataset
Attack Type 1 Scenario 1 Network Traffic Dataset
Attack Type 1 Scenario j "powertrace" Dataset
Attack Type 1 Scenario j Network Traffic Dataset
Attack Type i Scenario 1 "powertrace" Dataset
Attack Type i Scenario 1 Network Traffic Dataset
Attack Type i Scenario j "powertrace" Dataset
Attack Type i Scenario j Network Traffic Dataset
attack_type_1-scenario_1-pwrtrace.csv
attack_type_1-scenario_1-mote_1.csv
. . .
attack_type_1-scenario_1-mote_n.csv
attack_type_1-scenario_1-radiolog.csv
attack_type_1-scenario_1-radiologICMPv6.csv
attack_type_1-scenario_1-radiologUDP.csv
attack_type_1-scenario_j-pwrtrace.csv
attack_type_1-scenario_j-mote_1.csv
. . .
attack_type_1-scenario_j-mote_n.csv
attack_type_1-scenario_j-radiolog.csv
attack_type_1-scenario_j-radiologICMPv6.csv
attack_type_1-scenario_j-radiologUDP.csv
attack_type_i-scenario_1-pwrtrace.csv
attack_type_i-scenario_1-mote_1.csv
. . .
attack_type_i-scenario_1-mote_n.csv
attack_type_i-scenario_1-radiolog.csv
attack_type_i-scenario_1-radiologICMPv6.csv
attack_type_i-scenario_1-radiologUDP.csv
attack_type_i-scenario_j-pwrtrace.csv
attack_type_i-scenario_j-mote_1.csv
. . .
attack_type_i-scenario_j-mote_n.csv
attack_type_i-scenario_j-radiolog.csv
attack_type_i-scenario_j-radiologICMPv6.csv
attack_type_i-scenario_j-radiologUDP.csv

Figure 4.56 Malicious IoT Datasets – Generic Structure

Chapter 5 Datasets Analysis

5.1 Introduction

This Chapter is focused on the analysis of the generated benign “powertrace” and network traffic datasets, presented in Chapter 3, and the generated malicious “powertrace” and network traffic datasets, demonstrated in Chapter 4. The Chapter starts with the analysis of the malicious “powertrace” datasets to investigate whether their raw features can be important in the detection of anomalies in the network-level power profiling of low-power IoT devices due to UDP flooding attacks, blackhole attacks, sinkhole attacks, or sleep deprivation attacks. Next, the Chapter continues with investigating the extraction of new features, more informative and non-redundant, based on the raw features of the generated benign and malicious datasets. The new features are intended to constitute valuable features for anomaly-based detection of UDP flooding attacks, blackhole attacks, sinkhole attacks and sleep deprivation attacks in IoT networks. To this end, the total energy consumption of each mote is investigated as a valuable feature in Section 5.2.2. Last but not least, the generated benign and malicious network traffic datasets are also analysed in Section 5.3.1 to derive new features more informative in terms of the behaviour of the network traffic.

5.2 “powertrace” Datasets Analysis

5.2.1 Malicious “powertrace” Datasets Analysis – Feature Selection

The generated malicious “powertrace” datasets, presented in Chapter 4 include information about raw features (e.g., “all_cpu”, “all_transmit”, “all_listen”, “cpu”, “lpm”, etc.) that can be analysed to investigate whether they can be important in the detection of anomalies in the network-level power profiling of low-power IoT devices (i.e., motes) [107] due to one of the following attacks in the IoT network: UDP flooding attack, blackhole attack, sinkhole attack, and sleep deprivation attack. Towards this direction, the Mutual Information (MI) method is applied to measure the importance of the different features of each malicious “powertrace” dataset (i.e., “udp-flood-pwrtrace.csv”, “blackhole-pwrtrace.csv”, “sinkhole-pwrtrace.csv”, and “sleep_depr-pwrtrace.csv”) and identify the most significant ones. The MI method was selected as it is commonly used to measure the usefulness of a feature in discriminating the different classes in a dataset [108]. Before applying the MI method, all malicious “powertrace” datasets were pre-processed in the following way: the feature “Clock_time” was filtered out along with the features related to the simulation duration (i.e., “all_cpu”, “all_lpm”, “all_transmit”, “all_listen”, ”all_idle_transmit”, and “all_idle_listen” features) and the “seq no” feature. Besides that, the “P” feature was omitted, because it only has a fixed value throughout all of the collected records. Finally, the “ID” and “Rime Address” were also filtered out because it was observed that they led to overfitting.

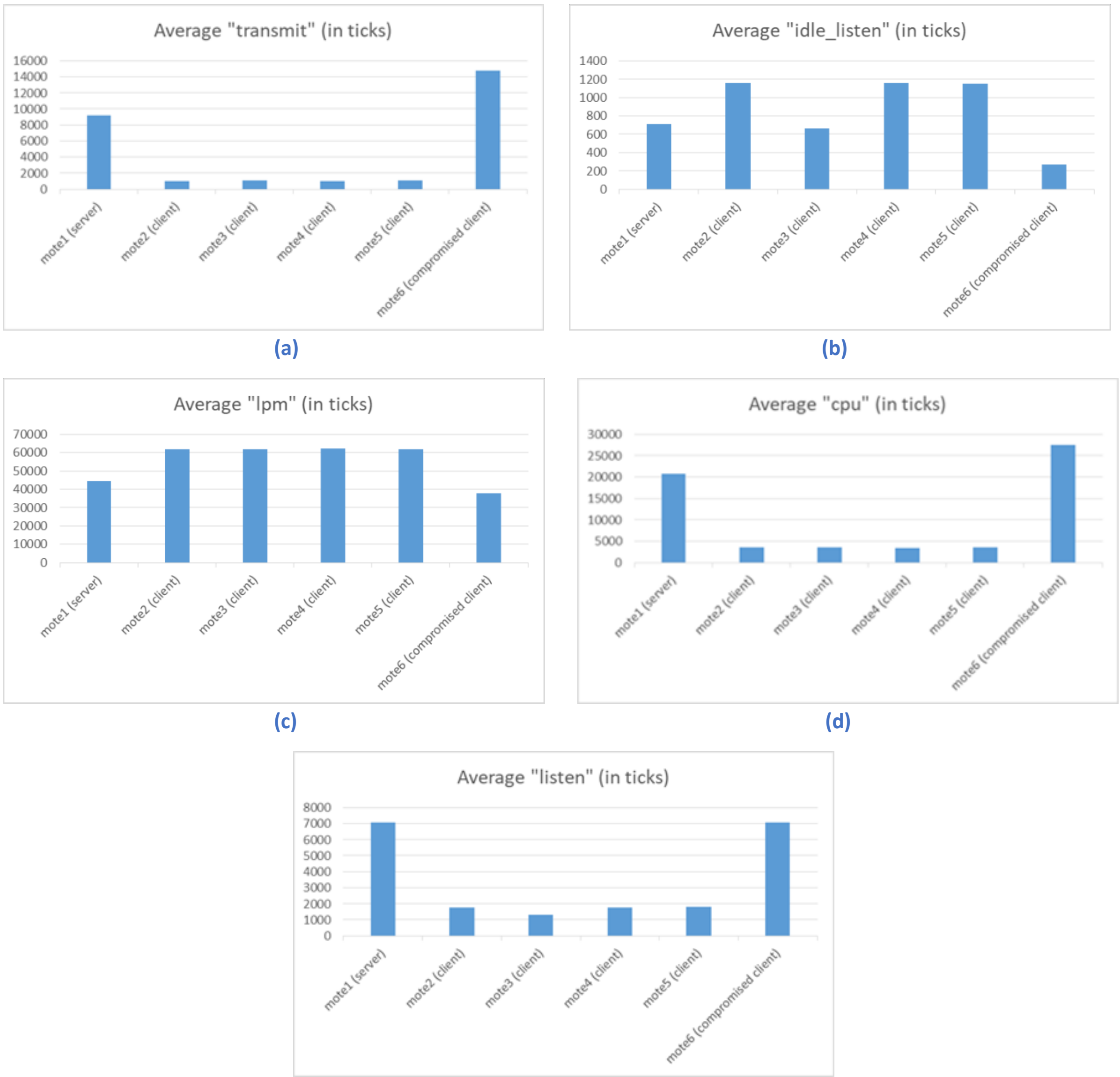
5.2.1.1 UDP Flooding Attack “powertrace” Dataset Analysis

The following features from the processed “udp-flood-pwrtrace.csv” file were the features whose importance was calculated based on the “label” feature (i.e., “0” for normal and “1” for malicious) by applying the MI method: “cpu”, “lpm”, “transmit”, “listen”, “idle_transmit” and “idle_listen”. The results, sorted by value in descending order, are shown below in Table 5.1, where the “idle_transmit” feature is the one with the least importance.

Feature	MI (in bits)
“transmit”	0.3571
“idle_listen”	0.3432
“lpm”	0.2669
“cpu”	0.2630
“listen”	0.1888
“idle_transmit”	0.0039

Table 5.1 Mutual Information – Features – “udp-flood-pwrtrace.csv”.

In addition, the average values of the first five features included in Table 5.1 for each mote were calculated and the results are shown below in Figure 5.1.



(a)

(b)

(c)

(d)

(e)

Figure 5.1 “udp-flood-pwrtrace.csv” - Average values (in ticks) for “transmit”, “idle_listen”, “lpm”, “cpu”, and “listen”.

Based on the results included in Figure 5.1, the following observations have been made:

- Mote 6 (i.e., compromised client) and mote 1 (i.e., UDP-server) have the highest average value for the "transmit" feature. Although this is expected for mote 1 as it the server in the IoT network scenario, it is not normal for a benign client. Nevertheless, it is expected for a compromised mote implementing a UDP flooding attack by transmitting many UDP packets to the target mote (i.e., mote 1 – UDP-server mote).
- Mote 6 (i.e., compromised client) and mote 1 (i.e., UDP-server) have the lowest average value for the "lpm" feature. Although this is expected for mote 1 as it the server in the IoT network scenario, it is not expected for a benign client. Nevertheless, it is expected for a compromised mote implementing a UDP flooding attack by generating and transmitting many UDP packets.
- Mote 6 (i.e., compromised client) and mote 1 (i.e., UDP-server) have the highest average value for the "cpu" feature. Although this is expected for mote 1 as it the server in the IoT network scenario, it is not expected for a benign client. However, it is expected for a compromised mote implementing a UDP flooding attack by generating and transmitting many UDP packets to the target mote (i.e., mote 1 – UDP-server mote).
- Mote 6 (i.e., compromised client) has the highest average value for the "listen" feature that is expected for the compromised client that we implemented to simulate a UDP flooding attack as it receives a high number of responses (i.e., a kind of acknowledgement packets sent back by the target-server) due the way the compromised mote was implemented.

Therefore, based on the information included in Table 5.1 and the above observations from Figure 5.1, the following conclusions are drawn: a) the "idle_transmit" feature can be omitted from the "udp-flood-pwrtrace.csv" dataset as its MI is close to zero, meaning that the "idle_transmit" feature provides very little information for the "label" feature (i.e., "0" for normal and "1" for malicious); and b) the following features can be valuable for anomaly-based detection of UDP flooding attacks in IoT networks as they can characterise the behaviour of the compromised node: "transmit", "idle_listen", "listen", "lpm", and "cpu".

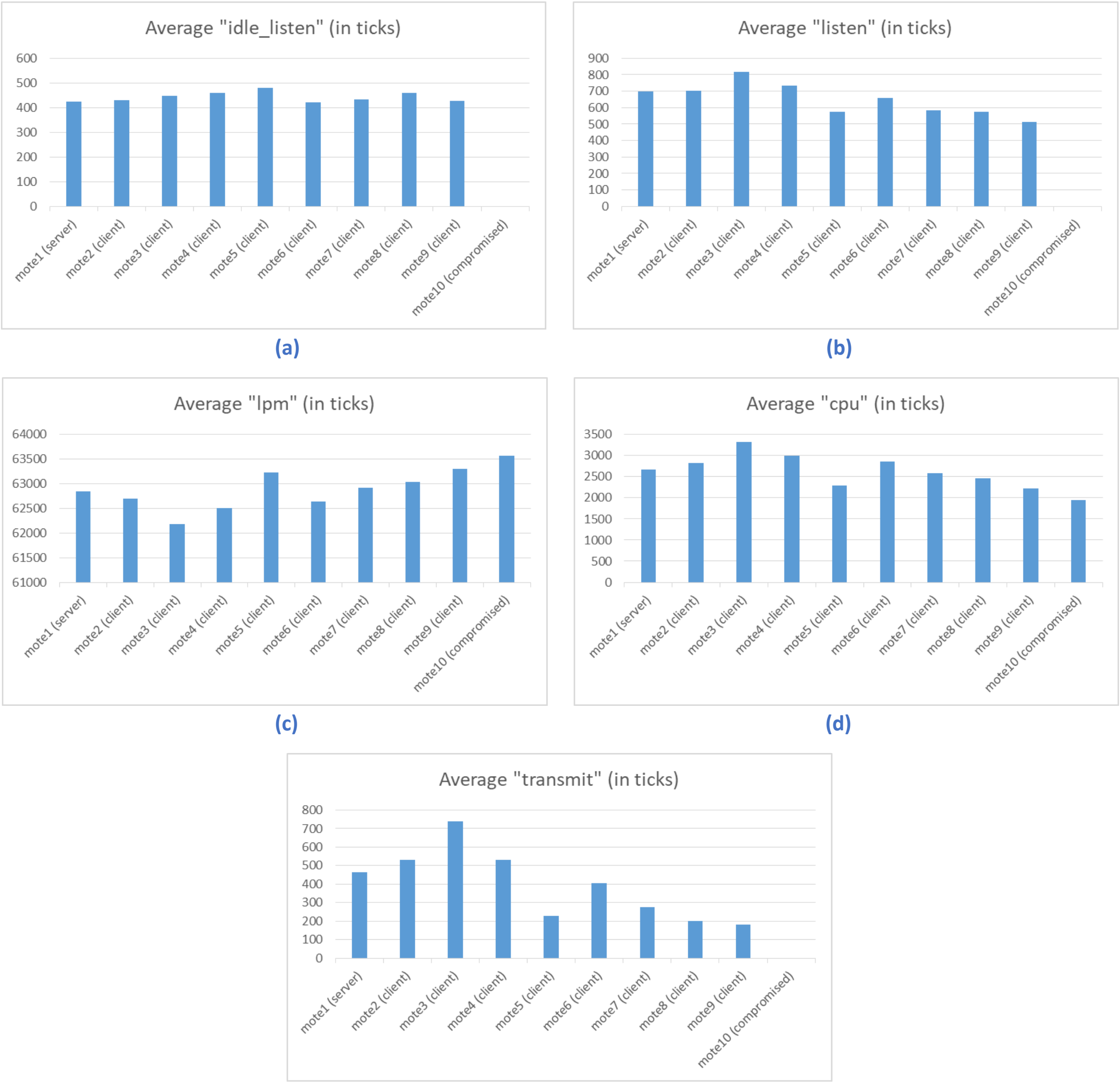
5.2.1.2 Blackhole Attack "powertrace" Dataset Analysis

The following features from the processed "blackhole-pwrtrace.csv" file were the features whose importance was calculated based on the "label" feature (i.e., "0" for normal and "1" for malicious) by applying the MI method: "cpu", "lpm", "transmit", "listen", "idle_transmit" and "idle_listen". The results, sorted by value in descending order, are shown below in 5.2, where the "idle_transmit" feature is the one with the least importance.

Feature	MI (in bits)
"idle_listen"	0.2217
"listen"	0.2214
"lpm"	0.1533
"cpu"	0.1475
"transmit"	0.0101
"idle_transmit"	0.0020

Table 5.2 Mutual Information – Features – "blackhole-pwrtrace.csv".

Furthermore, the average values of the first five features included in Table 5.2 for each mote were calculated and the results are presented below in Figure 5.2.



(a)

(b)

(c)

(d)

(e)

Figure 5.2 "blackhole-pwrtrace.csv" - Average values (in ticks) for "idle_listen", "listen", "lpm", "cpu", and "transmit".

Based on the results included in Figure 5.2, the following observations have been made:

- Mote 10 (i.e., compromised client) has the highest average value for the "lpm" feature which is expected for a compromised mote implementing a blackhole attack by dropping the packets that it has to forward.
- Mote 10 (i.e., compromised client) has the lowest average value for the "cpu" feature which is expected for a compromised mote implementing a blackhole attack by dropping the packets that it has to forward.
- Mote 10 (i.e., compromised client) has the lowest average value for the "transmit" feature (i.e., it is 0), which is not very common for benign client, but it is expected for a compromised mote dropping the packets that it has to forward in order to implement a blackhole attack.

- The average value for the “listen” and “idle_listen” features are zero because of the way we implemented the blackhole attack. Both features should be higher because of the behaviour of the compromised mote that implements a blackhole attack. In particular, we programmed the compromised mote to switch off, after 25 minutes from the beginning, not only the transmission feature in order to disrupt the communication chain but also the receiving feature. The fix of this issue in the blackhole implementation in the Cooja simulator is going to be part of our future work.

Consequently, based on the information included in Table .2 and the above observations derived from Figure 5.2, the following conclusions are drawn: a) the “idle_transmit” feature can be omitted from the “blackhole-pwrtrace.csv” dataset as its MI is close to 0, meaning that the “idle_transmit” feature provides very little information for the “label” feature (i.e., “0” for normal and “1” for malicious); and b) the following features can be valuable for anomaly-based detection of blackhole attacks in IoT networks as they can characterise the behaviour of the compromised mote: “lpm”, “cpu”, and “transmit”. The importance of the “listen” and “idle_listen” features that achieve the highest MI scores will be evaluated further in the near future, when the implementation issue is fixed, based also on their average values.

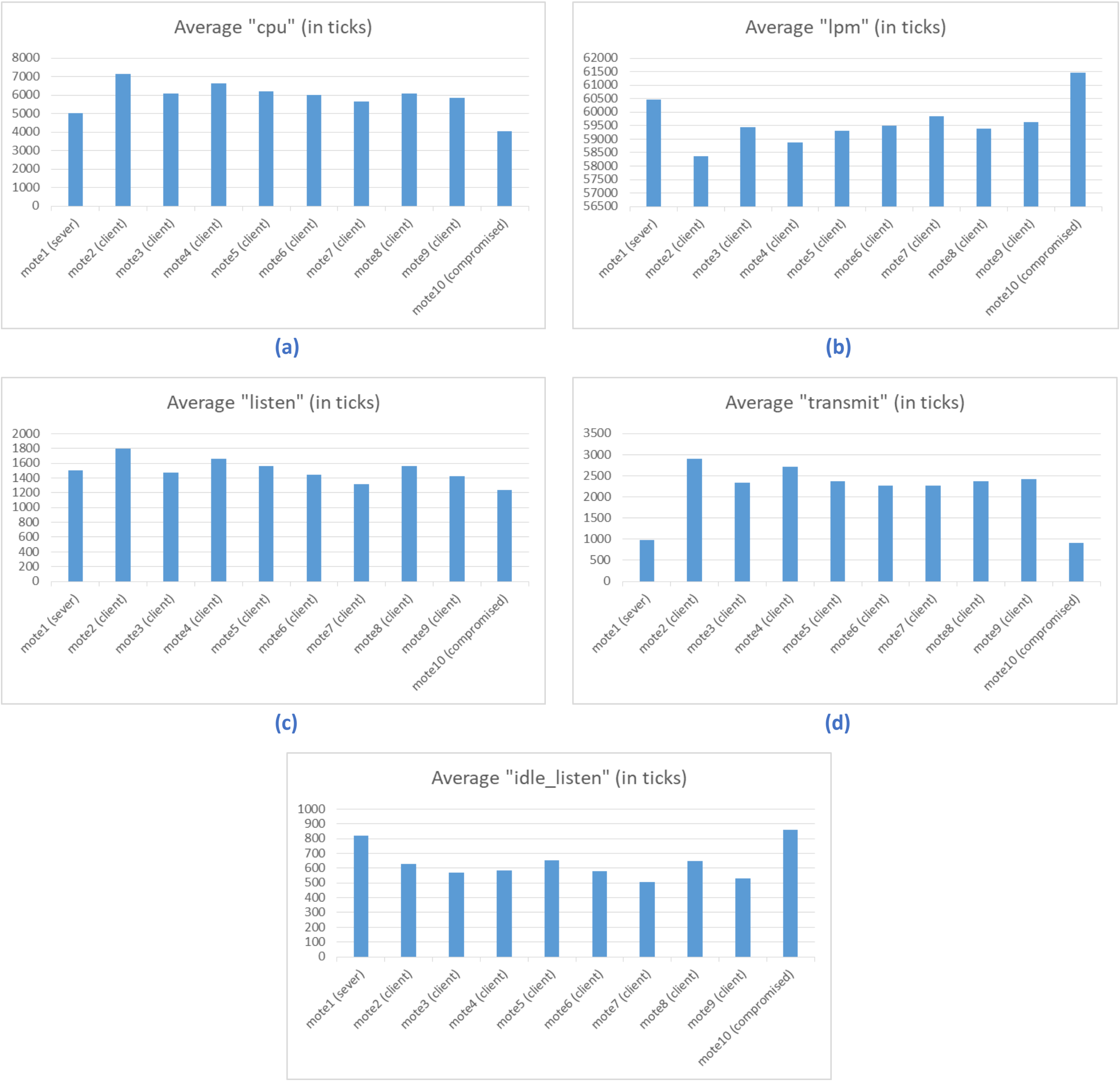
5.2.1.3 Sinkhole Attack “powertrace” Dataset Analysis

The following features from the processed “sinkhole-pwrtrace.csv” file were the features whose importance was calculated based on the “label” feature (i.e., “0” for normal and “1” for malicious) by applying the MI method: “cpu”, “lpm”, “transmit”, “listen”, “idle_transmit” and “idle_listen”. The results, sorted by value in descending order, are shown below in Table , where the “idle_transmit” feature is the one with the least importance.

Feature	MI (in bits)
“cpu”	0.1009
“lpm”	0.0899
“transmit”	0.0698
“listen”	0.0518
“idle_listen”	0.0174
“idle_transmit”	0.0000

Table 5.3. Mutual Information – Features – “sinkhole-pwrtrace.csv”.

Furthermore, the average values of the first five features included in Table 5.3 for each mote were calculated and the results are presented below in Figure 5.3.



(a)

(b)

(c)

(d)

(e)

Figure 5.3 “sinkhole-pwrtrace.csv” - Average values (in ticks) for “cpu”, “lpm”, “listen”, “transmit” and “idle_listen”.

Based on the results included in Figure 5.3, the following observations have been made:

- Mote 10 (i.e., compromised client) has the lowest average value for the “cpu” feature which is expected for a compromised mote implementing a sinkhole attack by dropping the received packets before they are processed and forwarded.
- Mote 10 (i.e., compromised client) has the highest average value for the “lpm” feature which is expected for a compromised mote implementing a sinkhole attack by dropping the received packets before they are processed and forwarded.
- Mote 10 (i.e., compromised client) has the lowest average value for the “transmit” feature, which is expected for a compromised mote that drops the received packets that it has to forward in order to implement a sinkhole attack. However, the average value of the transmit” feature is not zero because at the beginning of the simulation the compromised mote spends time to proclaim appealing false capabilities so that nearby motes will choose it as the forwarding mote in the routing process due to its very attractive false capabilities.
- Mote 10 (i.e., compromised client) has the highest average value for the “idle_listen” feature which is not very common for a benign client, but it is expected for a compromised

mote implementing a sinkhole attack as it should spend time listening to the medium to check if there is any packet in the air even though there is no packet being transmitted to it. It is worthwhile mentioning that the compromised mote that we programmed for the implementation of the sinkhole attack scenario in the Cooja simulator is a UDP-server mote and not a client mote in order to achieve the desired behaviour for the compromised mote that implements a sinkhole attack. This also explains the fact that its average value for the "idle_listen" feature is comparable, although higher, with the corresponding value of mote 1 which is the benign server of the sinkhole scenario.

Therefore, based on the information included in Table 5.3 and the above observations derived from Figure 5.3, the following conclusions are drawn: a) the "idle_transmit" feature can be omitted from the "sinkhole-pwrtrace.csv" dataset as its MI is 0, meaning that the "idle_transmit" feature provides zero information for the "label" feature (i.e., "0" for normal and "1" for malicious); and b) the following features can be valuable for anomaly-based detection of sinkhole attacks in IoT networks as they can characterise the behaviour of the compromised mote: "cpu", "lpm", "transmit", "idle_listen" and "listen".

5.2.1.4 Sleep Deprivation Attack "powertrace" Dataset Analysis

The following features from the processed "sleep_depr-pwrtrace.csv" file were the features whose importance was calculated based on the "label" feature (i.e., "0" for normal and "1" for malicious) by applying the MI method: "cpu", "lpm", "transmit", "listen", "idle_transmit" and "idle_listen". The results, sorted by value in descending order, are shown below in Table 5.4, where the "idle_transmit" feature is the one with the least importance.

Feature	MI (in bits)
"transmit"	0.1944
"cpu"	0.1200
"lpm"	0.1166
"listen"	0.0946
"idle_listen"	0.0859
"idle_transmit"	0.0024

Table 5.4 Mutual Information – Features – "sleep_depr-pwrtrace.csv".

In addition, the average values of the first five features included in Table 5.4 for each mote were calculated and the results are demonstrated below in Figure 5.4.

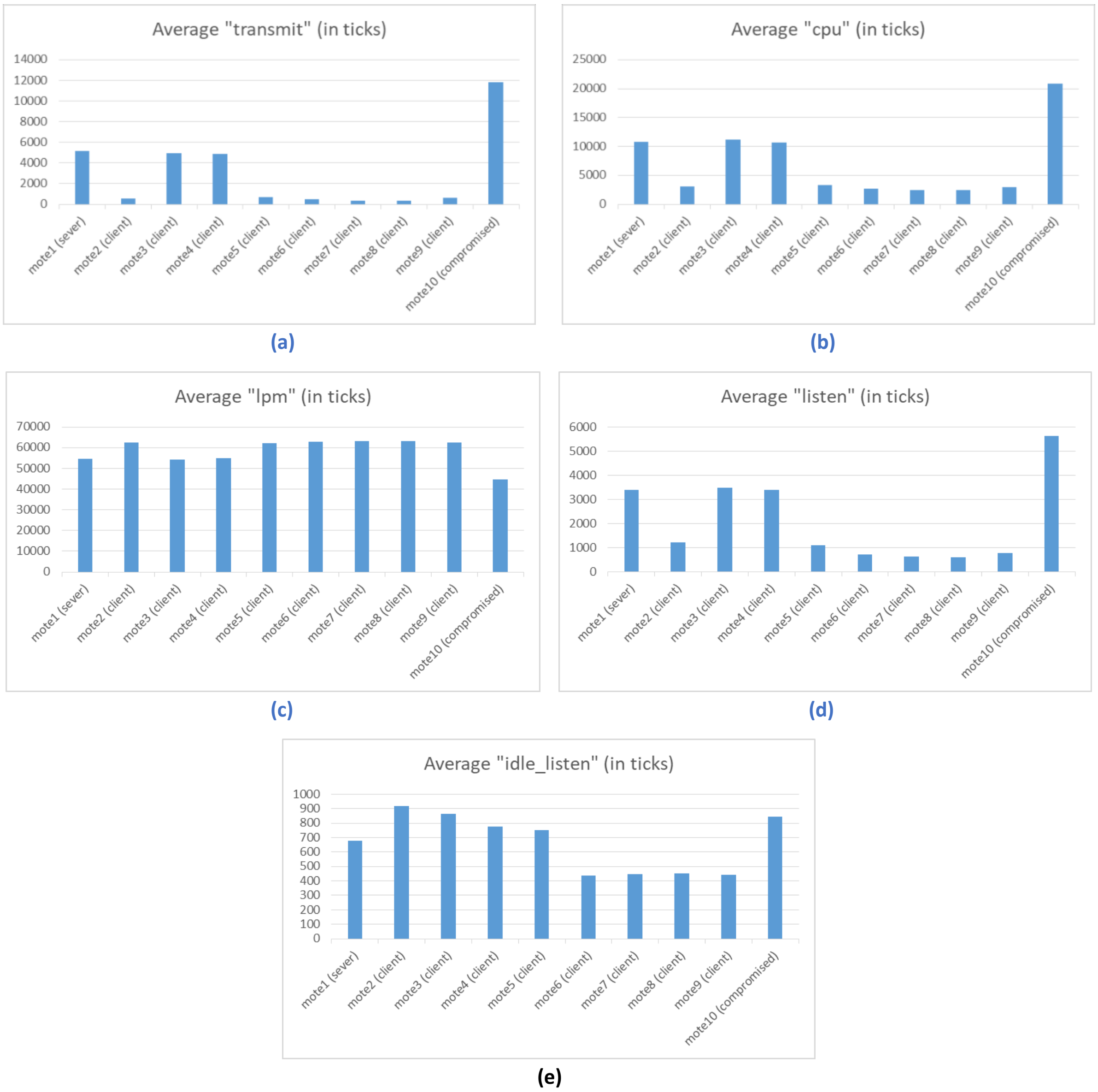


Figure 5.4 "sleep_depr-pwrtrace.csv" - Average values (in ticks) for "transmit", "cpu", "lpm", "listen", and "idle_listen".

Based on the results included in Figure 5.4, the following observations have been made:

- Mote 10 (i.e., compromised client) has the highest average value for the "transmit" feature which is expected for a compromised mote implementing a sleep deprivation attack by transmitting a high UDP traffic volume to the target mote (i.e., mote 4) which is the closest mote to the compromised mote.
- Mote 10 (i.e., compromised client) has the highest average value for the "cpu" feature which is expected for a compromised mote implementing a sleep deprivation attack by generating and transmitting many UDP packets to the target mote in order to break its programmed sleep routines and keep it continuously active until it is shut down due to a drained battery.
- Mote 10 (i.e., compromised client) has the lowest average value for the "lpm" feature which is expected for a compromised mote implementing a sleep deprivation attack by generating and transmitting many UDP packets to the target mote.
- Mote 10 (i.e., compromised client) has the highest average value for the "listen" feature that is expected for the compromised client that we implemented to simulate a sleep deprivation attack as it receives a high number of responses (i.e., a kind of acknowledgement packets

sent back by the server when it receives, via forwarding, the UDP packets sent by the compromised mote to mote4) due the way the compromised mote was implemented.

Therefore, based on the information included in Table 5.4 and the above observations derived from Figure 5.4, the following conclusions are drawn: a) the “idle_transmit” feature can be omitted from the “sleep_depr-pwrtrace.csv” dataset as its score for MI is close to zero, meaning that the “idle_transmit” feature provides very little information for the “label” feature (i.e., “0” for normal and “1” for malicious); and b) the following features can be valuable for anomaly-based detection of sleep deprivation attacks in IoT networks as they can characterise the behaviour of the compromised node: “transmit”, “cpu”, “lpm”, “listen”, and “idle_listen”.

5.2.2 Benign and Malicious “powertrace” Datasets Analysis – Feature Extraction

The generated malicious “powertrace” datasets, presented in Chapter 4 include information about raw features (e.g., “all_cpu”, “all_lpm”, “all_transmit”, “all_listen”, “all_idle_transmit”, “all_idle_listen”, “cpu”, “lpm”, “transmit”, “listen”, “idle_transmit”, “idle_listen” etc.) that can be used to derive new features more informative, in terms of the behaviour of each mote, and non-redundant. The new features are intended to constitute valuable features for training and evaluating AIDS for IoT networks. Towards this direction, the total energy consumption of each mote in an IoT network is investigated in this Section as a valuable feature for attack detection.

Based on [109] and [110], the total energy consumption of each mote, at the reading (i.e., record) *i*, is given by the sum of a) the energy consumption in the CPU state; b) the energy consumption in the LPM state; c) the energy consumption in the TX state; and d) the energy consumption in the RX state, at the reading (i.e., record) *i*, as shown in the equation below:

$$E_{total_i}(mJ) = E_{cpu_{total_i}} + E_{lpm_{total_i}} + E_{tx_{total_i}} + E_{rx_{total_i}} = \\ = (I_{cpu} \times V_{cpu} \times T_{cpu_i}) + (I_{lpm} \times V_{lpm} \times T_{lpm_i}) + (I_{tx} \times V_{tx} \times T_{tx_i}) \\ + (I_{rx} \times V_{rx} \times T_{rx_i}) \tag{5.1}$$

where

I_{cpu}: the nominal current in the CPU state;

I_{lpm}: the nominal current in the LPM state;

I_{tx}: the nominal current in the TX state;

I_{rx}: the nominal current in the RX state;

V_{cpu}: the nominal voltage in the CPU state;

V_{lpm}: the nominal voltage in the LPM state;

V_{tx}: the nominal voltage in the TX state;

V_{rx}: the nomnal voltage in the RX state;

$$T_{cpu_i} = \frac{cpu_i\ (\#\ ticks)}{RTIMER\_ARCH\_SECOND} = \frac{cpu_i\ (\#\ ticks)}{32{,}768}$$

$$T_{lpm_i} = \frac{lpm_i\ (\#\ ticks)}{RTIMER\_ARCH\_SECOND} = \frac{lpm_i\ (\#\ ticks)}{32{,}768}$$

$$T_{tx_i} = \frac{tx_i\ (\#\ ticks)}{RTIMER\_ARCH\_SECOND} = \frac{tx_i(\#\ ticks)}{32{,}768}$$

$$T_{rx_i} = \frac{rx\ _i(\#\ ticks)}{RTIMER\_ARCH\_SECOND} = \frac{rx_i\ (\#\ ticks)}{32{,}768}$$

Based on Equation (5.1) and Table 5.5 [105] that provides the typical operating conditions for a Tmote Sky mote, the total energy consumption, at the reading (i.e., record) *i*, is given by the Equation (5.2):

	MIN	NOM (Typical)	MAX	UNIT
Supply voltage	2.1	3.0	3.6	V
Supply voltage during flash memory programming	2.7	3.0	3.6	V
Operating free air temperature	-40		85	ºC
Current Consumption: MCU on, Radio RX		21.8	23	mA
Current Consumption: MCU on, Radio TX		19.5	21	mA
Current Consumption: MCU on, Radio off		1800	2400	µA
Current Consumption: MCU idle, Radio off		54.5	1200	µA
Current Consumption: MCU standby		5.1	21.0	µA

Table 5.5 Typical Operating Conditions for Tmote Sky motes [105].

$$E_{total_i}(mJ) = 1.8 \times 3 \times \left(\frac{cpu_i\ (\#\ ticks)}{32{,}768}\right) + 0.0545 \times 3 \times \left(\frac{lpm_i\ (\#\ ticks)}{32{,}768}\right) + 19.5 \times 3 \times \left(\frac{tx_i(\#\ ticks)}{32{,}768}\right) + 21.8 \times 3 \times \left(\frac{rx_i\ (\#\ ticks)}{32{,}768}\right) \quad (5.2)$$

5.2.2.1 Benign “powertrace” Dataset – Average Total Energy Consumption per Mote

Based on Equation (5.2) and the following features, from the generated benign “pwrtrace.csv” dataset in Section 3.3, for each mote: a) “all_cpu”; b) “all_lpm”; c) “all_transmit”; and d) “all_listen”, the average total energy consumption by each mote, during the simulation time (i.e., 60 min = 3600 sec) is shown below in Figure 5.5. The confidence interval has been considered to be the acquisition time which is 2 seconds.

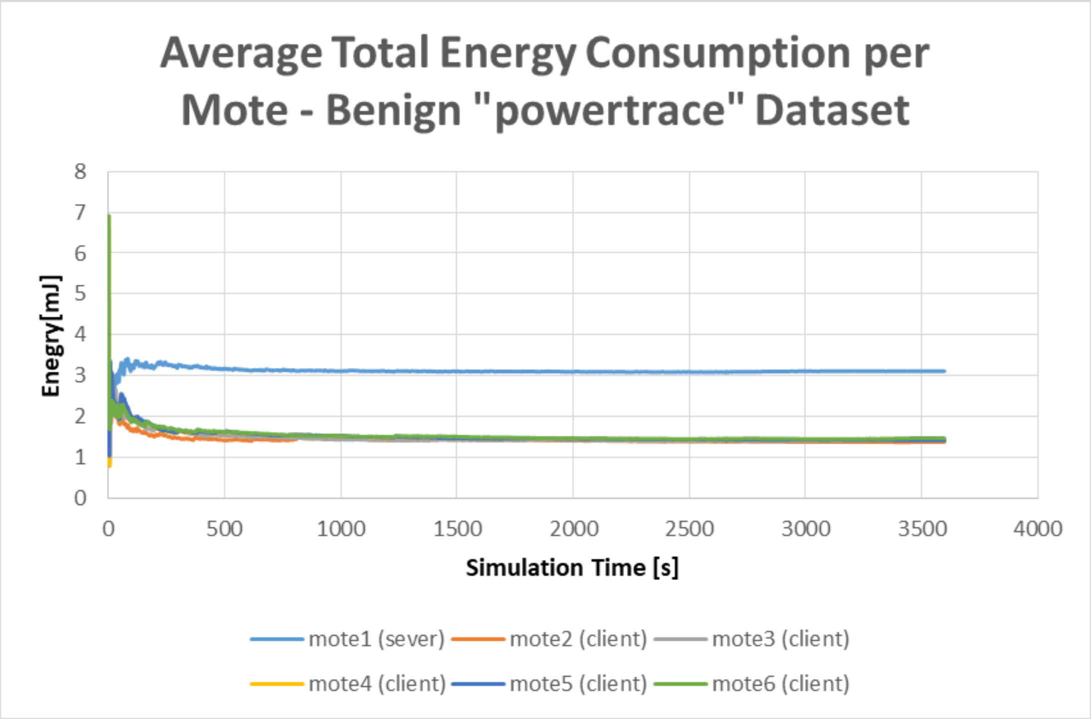


Figure 5.5 Average Total Energy Consumption per Mote – Benign "powertrace" Dataset.

5.2.2.2 UDP Flooding Attack "powertrace" Dataset – Average Total Energy Consumption per Mote

Based on Equation (5.2) and the following features, from the generated malicious "udp-flood-pwrtrace.csv" dataset in Section 4.2.2, for each mote: a) "all_cpu"; b) "all_lpm"; c) "all_transmit"; and d) "all_listen", the average total energy consumption per mote, during the simulation time (i.e., 60 min = 3600 sec) is shown below in Figure 5.6. The confidence interval has been considered to be the acquisition time which is 2 seconds.

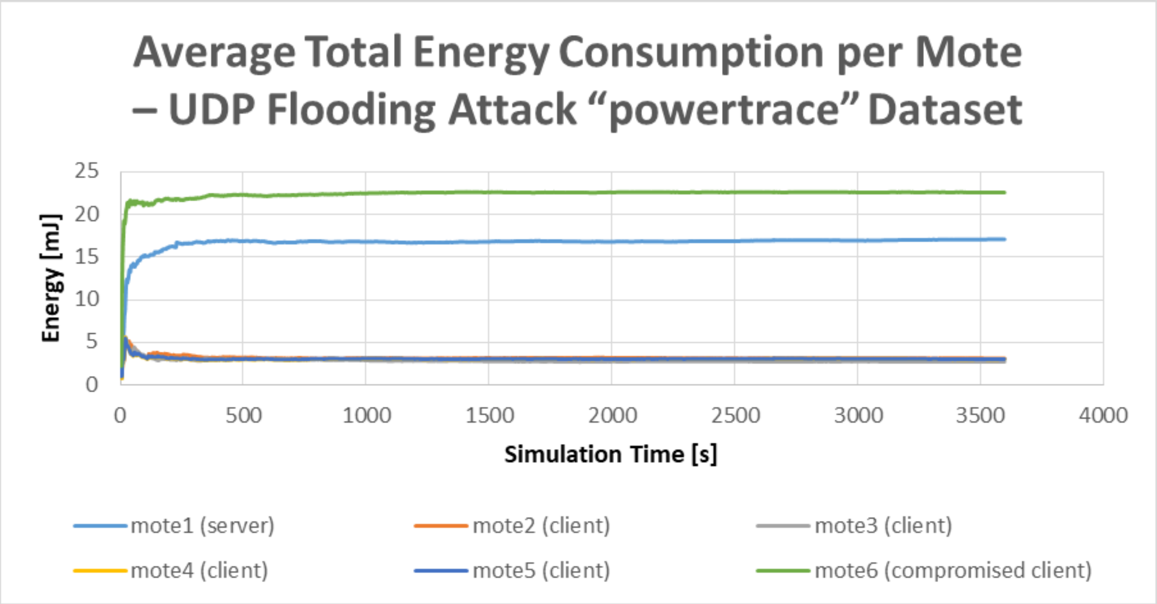


Figure 5.6 Average Total Energy Consumption per Mote – UDP Flooding Attack "powertrace" Dataset.

According to the results demonstrated in Figure 5.6, it is clear that the compromised mote (i.e., mote6) that carried out the UDP flooding attack consumed much more energy than any other benign motes (i.e., client or server) in the UDP flooding attack scenario as it generated and transmitted many UDP packets to the target server-mote (i.e., mote1). In addition, it is observed that the server-mote consumed a high level of energy as it received a high number of UDP packets from the compromised mote. In particular, the server-mote in the UDP flooding attack consumed much more energy than the energy it consumed in the benign scenario as demonstrated in Figure 5.6. These observations are also reflected in Figure 5.7 and Figure 5.8 demonstrating the average CPU energy consumption and the average Radio (i.e., TX+RX) energy consumption per mote, respectively.

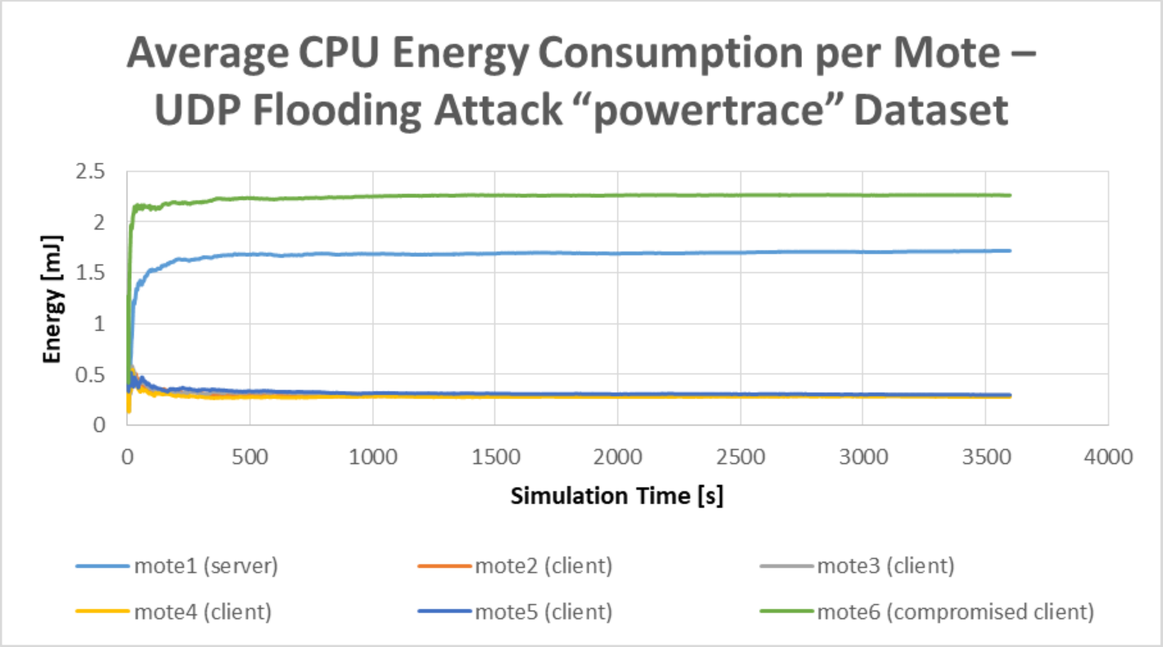


Figure 5.7 Average CPU Energy Consumption per Mote – UDP Flooding Attack "powertrace" Dataset.

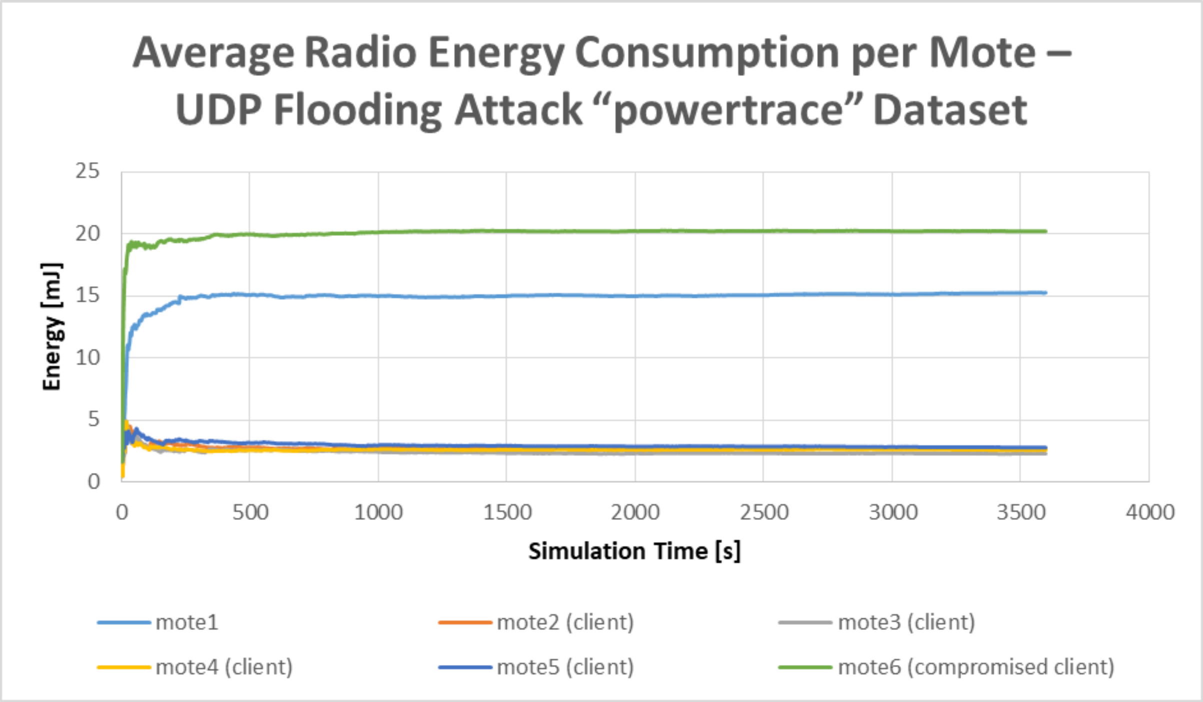


Figure 5.8 Average Radio (TX+RX) Energy Consumption per Mote – UDP Flooding Attack "powertrace" Dataset.

5.2.2.3 Blackhole Attack "powertrace" Dataset – Average Total Energy Consumption per Mote

Based on Equation (5.2) and the following features, from the generated malicious "blackhole-pwrtrace.csv" dataset in Section 4.3.2, for each mote: a) "all_cpu"; b) "all_lpm"; c) "all_transmit"; and d) "all_listen", the average total energy consumption per mote, during the simulation time (i.e., 60 min = 3600 sec) is shown below in Figure 5.9. The confidence interval has been considered to be the acquisition time which is 2 seconds.

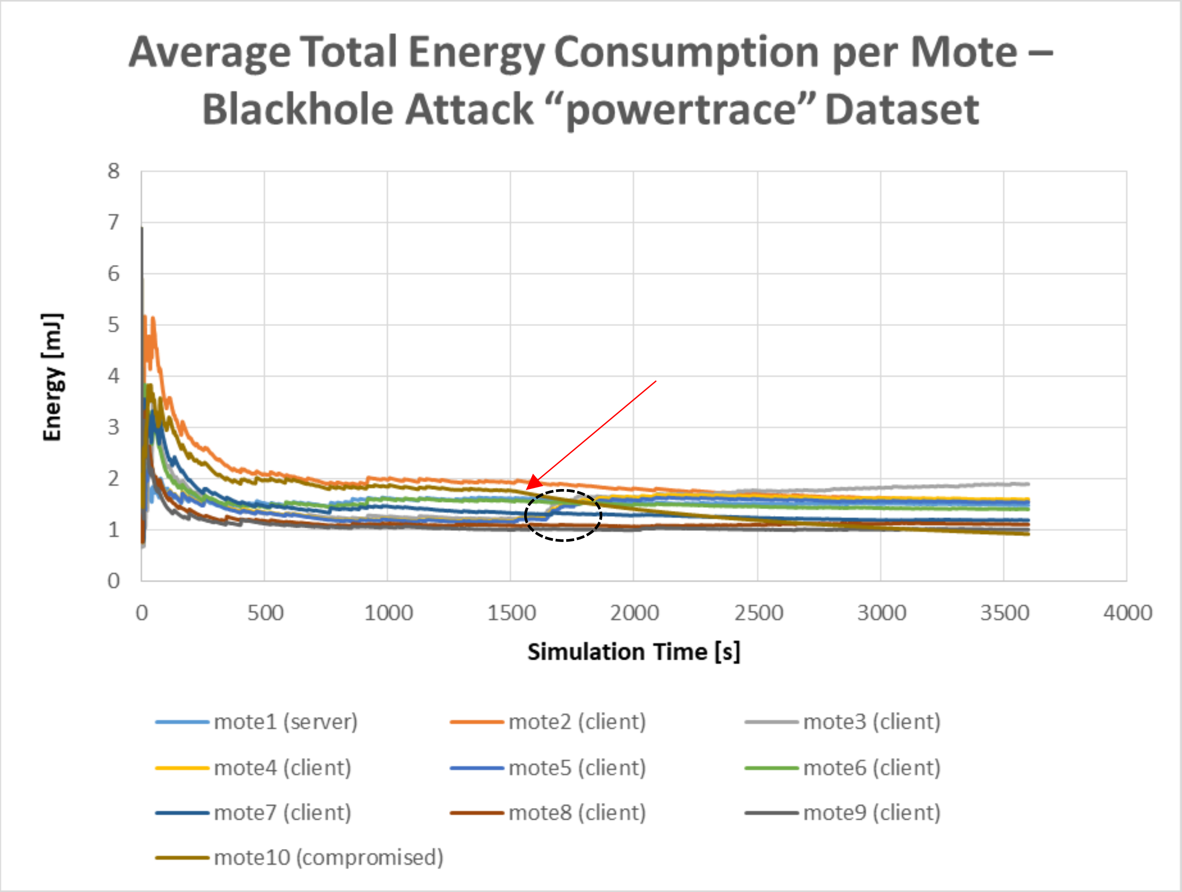


Figure 5.9 Average Total Energy Consumption per Mote – Blackhole Attack "powertrace" Dataset.

According to the results demonstrated in Figure 5.9, it is clear that the compromised mote (i.e., mote10) decreased its total energy consumption significantly (please see red arrow) as it was programmed to switch off, after 25 minutes (1,500 sec) from the beginning of the simulation, not only the transmission feature (TX) in order to disrupt the communication chain but also the receiving feature (RX). This observation is also clear in Figure 5.10 demonstrating the average Radio (i.e., TX+RX) energy consumption per mote. Furthermore, in Figure 5.9, it is shown that mote3, mote4, and mote5 increased their total energy consumption (see black dotted ellipse) because they increased their average radio energy consumption, as particularly depicted in Figure 5.10, as they were trying to re-establish connection with the mote-server due to the impact of the blackhole attack on the network.

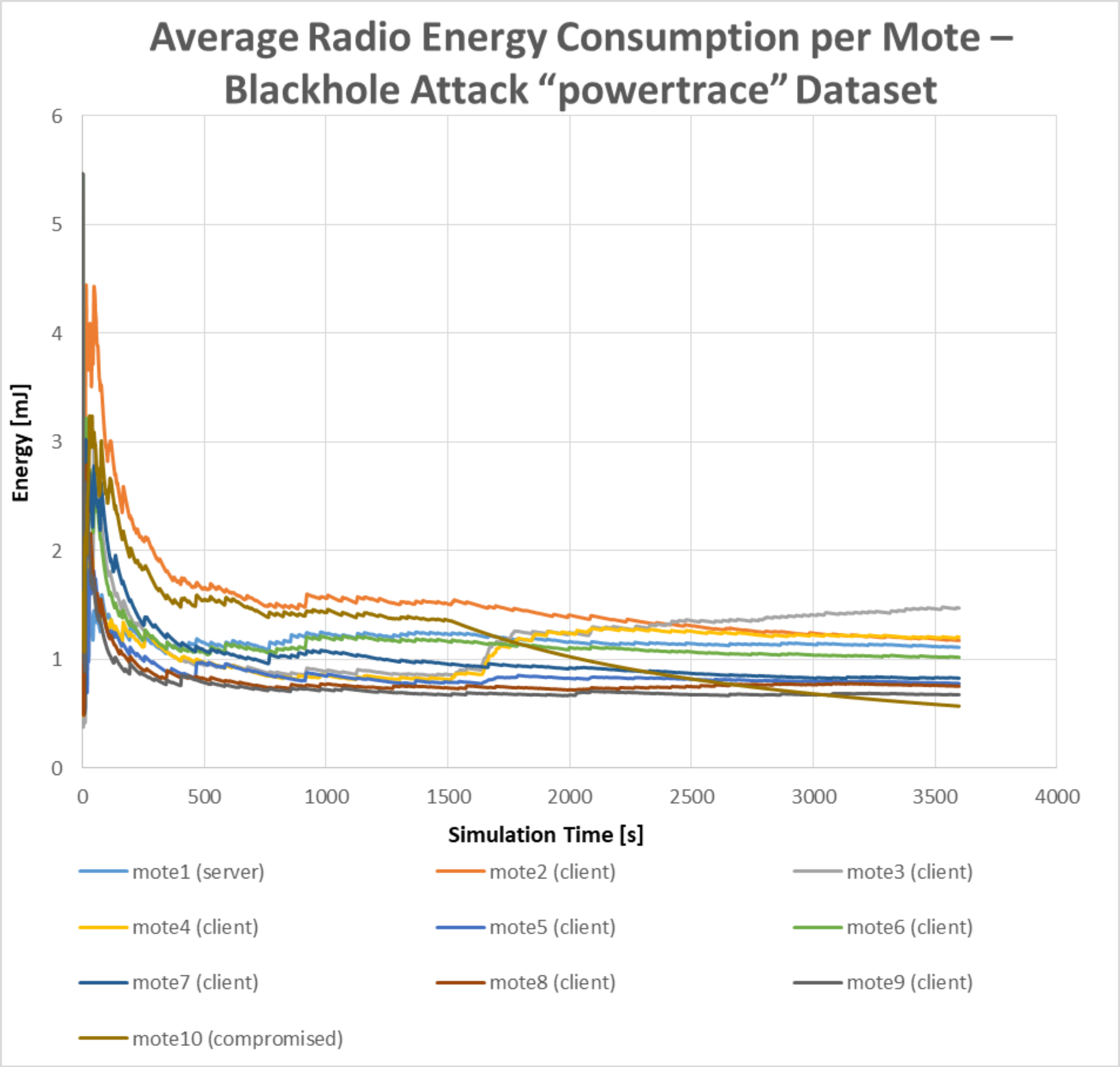


Figure 5.10 Average Radio (TX+RX) Energy Consumption per Mote – Blackhole Attack "powertrace" Dataset.

5.2.2.4 Sinkhole Attack "powertrace" Dataset – Average Total Energy Consumption per Mote

Based on Equation (5.2) and the following features, from the generated malicious "sinkhole-pwrtrace.csv" dataset in Section 4.4.2, for each mote: a) "all_cpu"; b) "all_lpm"; c) "all_transmit"; and d) "all_listen", the average total energy consumption per mote, during the simulation time (i.e., 60 min = 3600 sec) is shown below in Figure 5.11. The confidence interval has been considered to be the acquisition time which is 2 seconds.

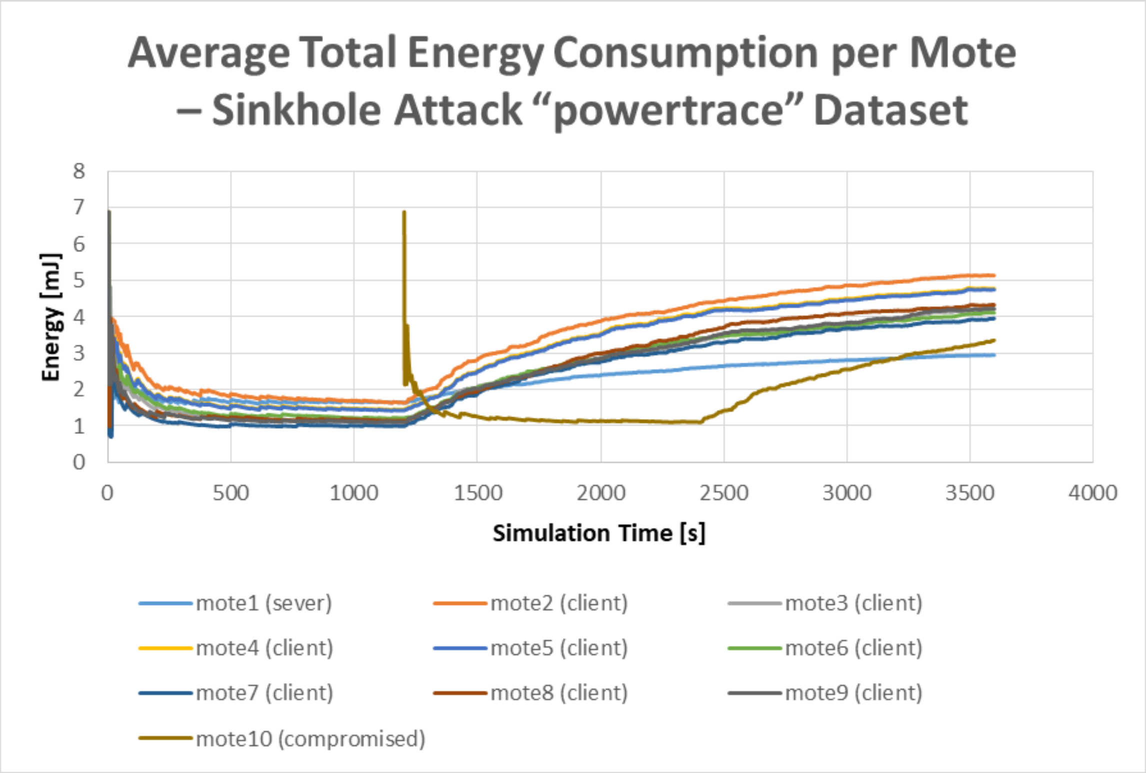


Figure 5.11 Average Total Energy Consumption per Mote – Sinkhole Attack “powertrace” Dataset.

Figure 5.11 shows that the compromised mote (i.e., mote10) that carried out the sinkhole attack consumed little total energy compared to the other benign motes (i.e., client or server) in the sinkhole scenario as it dropped the received packets before them being processed and forwarded. It is worthwhile noting that the spike of the energy consumption of the compromised mote at 1200th second was due to the fact that at that moment, the compromised mote was programmed to turn on as mentioned in section 4.4.1 (Sinkhole Attack Scenario – an example). This is also shown in detail in Figure 5.12. On the other hand, as also seen in Figure 5.13, all the other motes increased their energy consumption due to their efforts to respond to the impact of the sinkhole attack on the network.

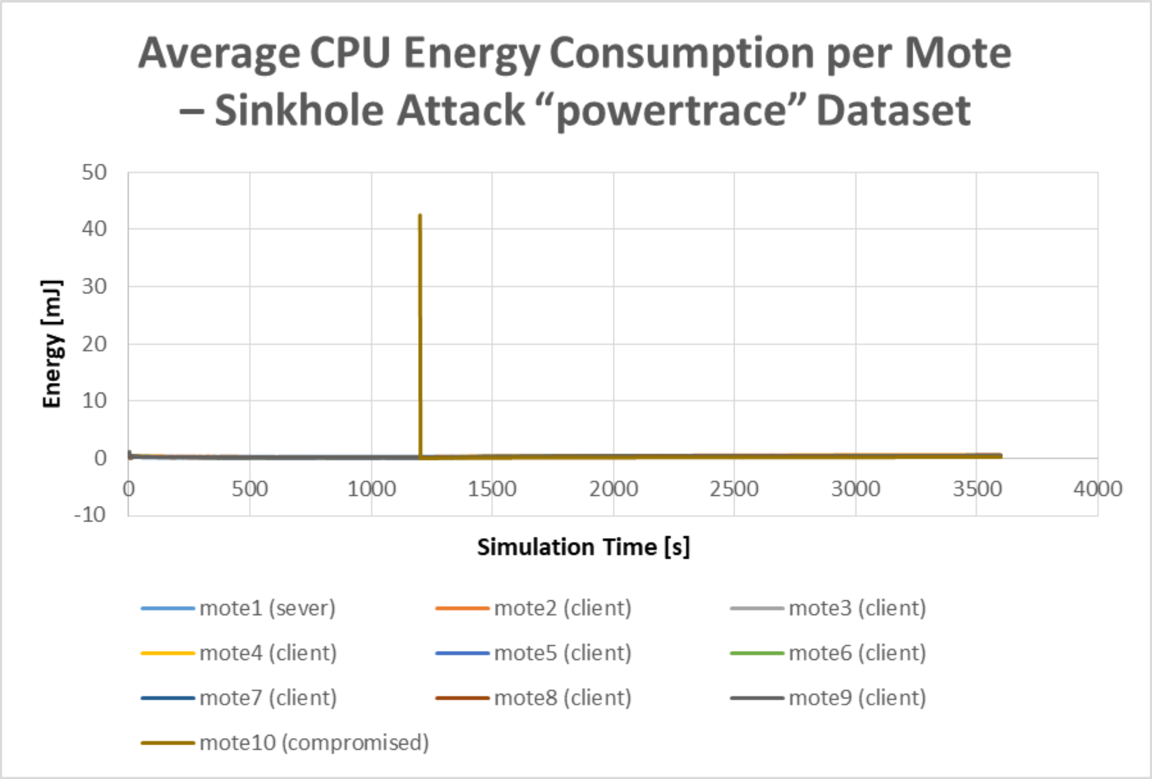


Figure 5.12Average CPU Energy Consumption per Mote – Sinkhole Attack "powertrace" Dataset.

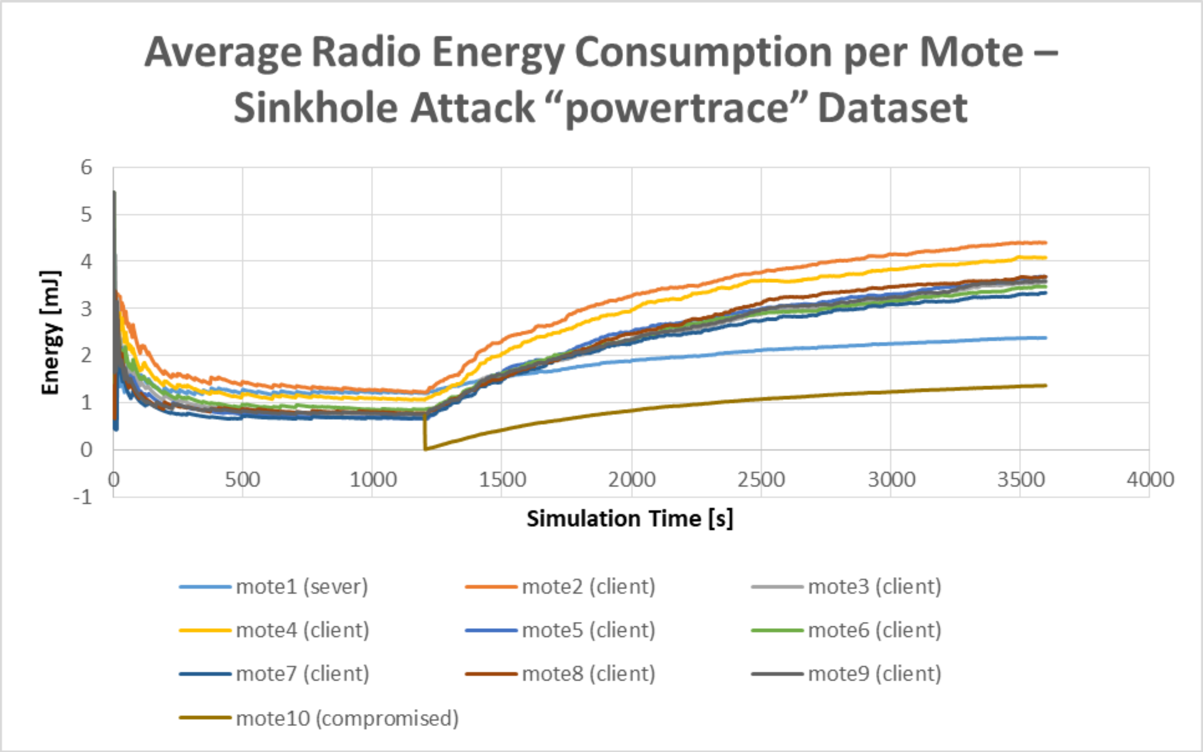


Figure 5.13 Average Radio Energy Consumption per Mote – Sinkhole Attack "powertrace" Dataset.

5.2.2.5 Sleep Deprivation Attack "powertrace" Dataset – Average Total Energy Consumption per Mote

Based on Equation (5.2) and the following features, from the generated malicious "sleep_depr-pwrtrace.csv" dataset in Section 4.5.2, for each mote: a) "all_cpu"; b) "all_lpm"; c) "all_transmit"; and d) "all_listen", the average total energy consumption per mote, during the simulation time (i.e., 60 min = 3600 sec) is shown below in Figure 5.14. The confidence interval has been considered to be the acquisition time which is 2 seconds.

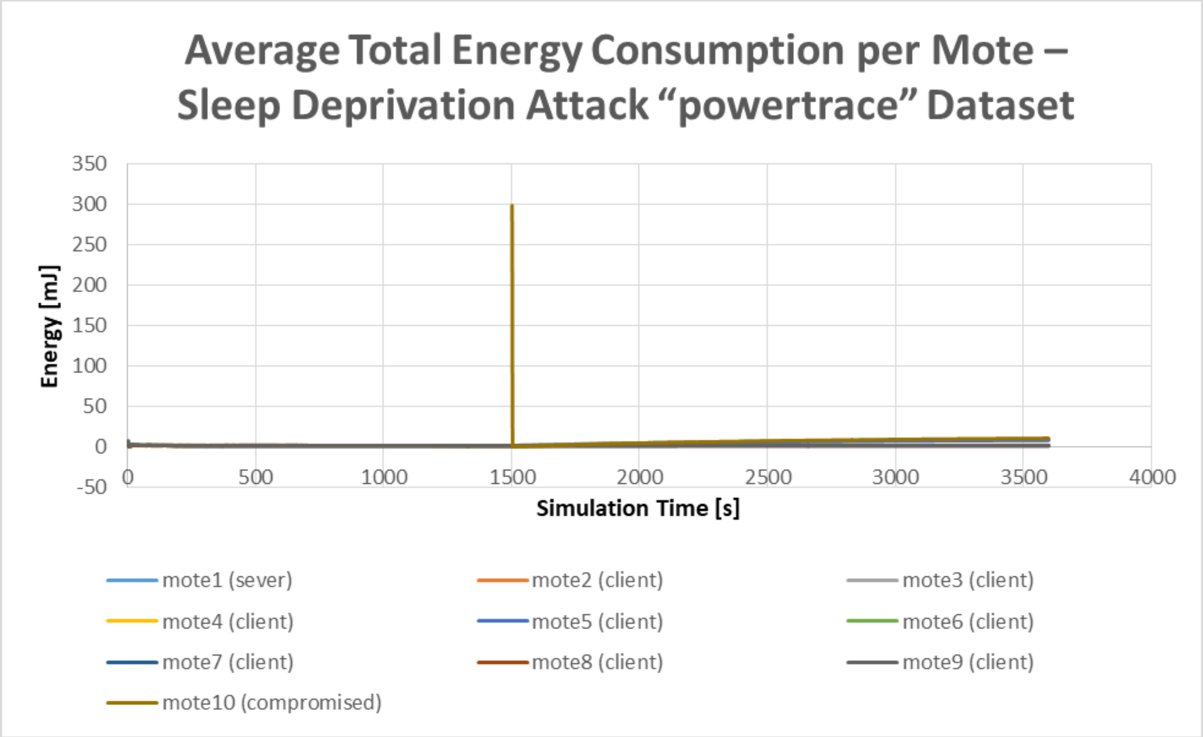


Figure 5.14 Average Total Energy Consumption per Mote – Sleep Deprivation Attack "powertrace" Dataset.

Figure 5.14 shows that the compromised mote (i.e., mote10) that carried out the sleep deprivation attack consumed more energy compared to the other benign motes (i.e., client or server) in the sleep deprivation scenario as it generated and transmitted many UDP packets to the target client-mote (i.e., mote4). Besides that, mote10 received a high number of responses (i.e., a kind of acknowledgement packets sent back by the server when it receives, via forwarding, the UDP packets sent by the compromised mote to mote4) due the way the compromised mote was implemented. It is worthwhile mentioning that the spike of the energy consumption of the compromised mote at 1500th second was due to the fact that at that moment, the compromised mote was programmed to turn on as mentioned in section 4.5.1 (Sleep Deprivation Attack Scenario – an example). This observation is also presented in detail in Figure . In addition, it is observed in Figure 5.16 that the server-mote (i.e., mote1) and the target client mote (i.e., mote4) consumed a high level of radio energy as they both received a high number of UDP packets from the compromised mote.

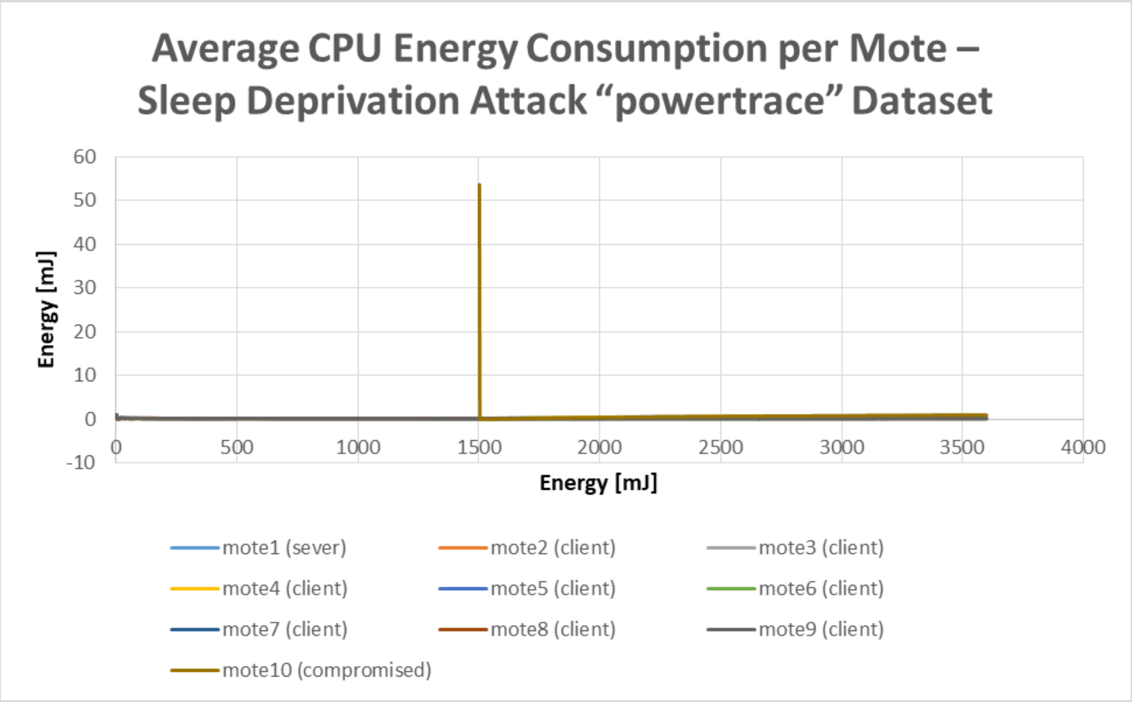


Figure 5.15 Average CPU Energy Consumption per Mote – Sleep Deprivation Attack "powertrace" Dataset.

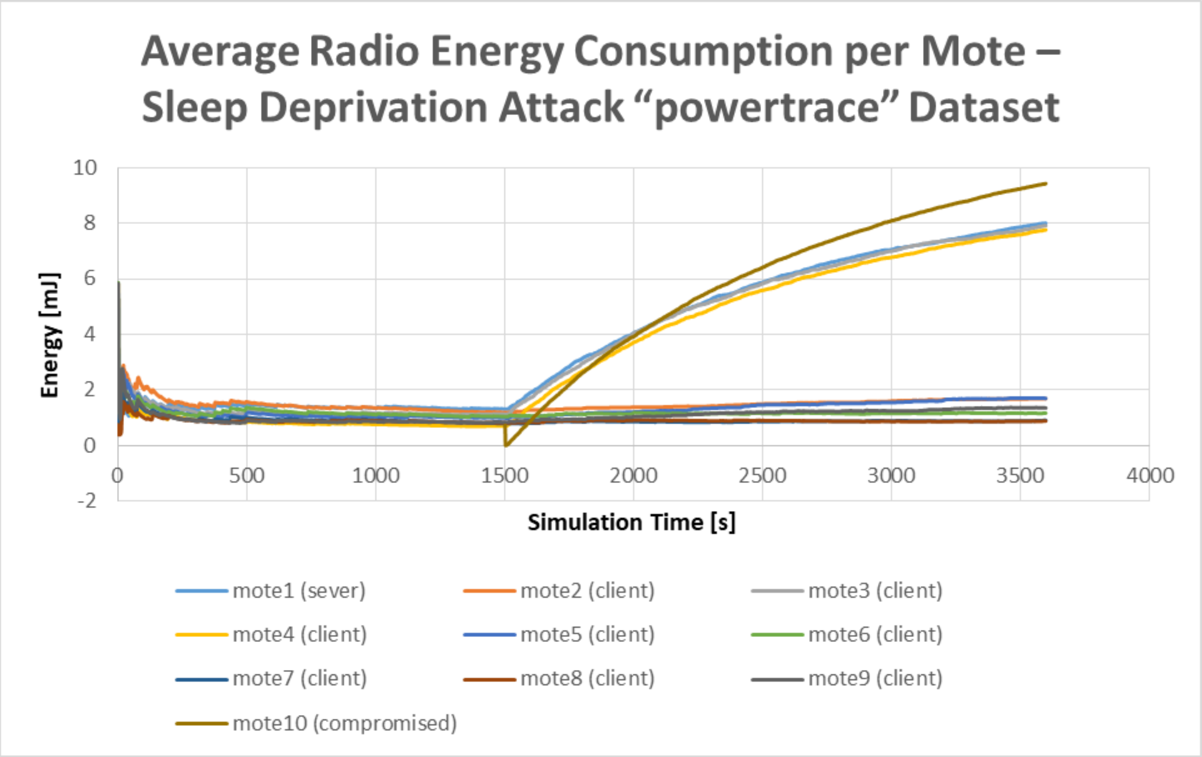


Figure 5.16 Average Radio Energy Consumption per Mote – Sleep Deprivation Attack "powertrace" Dataset.

5.3 Network Traffic Datasets Analysis

5.3.1 Benign and Malicious Network Traffic Datasets Analysis-Feature Extraction

The generated benign network traffic dataset (i.e., “radiolog.csv”), presented in Section 3.4, and the generated malicious network traffic datasets (i.e., “udp-flood-radiolog.csv”, “blackhole-radiolog.csv”, “sinkhole-radiolog.csv”, and “sleep_depr-radiolog.csv”), presented in Sections 4.2.3, 4.3.3, 4.4.3, and 4.5.3, include information about raw features, such as “source” address, “destination” address, “protocol”, and packet “length”, which can be used to derive new features more informative, in terms of the behaviour of the network traffic, and non-redundant. The new features are intended to constitute valuable features for training and evaluating AIDS for IoT networks. Towards this direction, the generated benign and malicious network traffic datasets are analysed in this Section in order to extract valuable features for anomaly-based detection of UDP flooding attacks, blackhole attacks, sinkhole attacks and sleep deprivation attacks in IoT networks.

5.3.1.1 Benign Network Traffic Dataset Analysis

From the generated benign “radiolog.csv” dataset in Section 3.4, Table 5.6 was extracted, demonstrating, in the last column, the percentage of the RPL packets overhead per mote[1] which is calculated as follows: the number of RPL packets per mote over the total number of exchanged packets within the network during the simulation time (i.e., 116,463 packets). The last row of Table 5.6 contains the total number of RPL packets (7,975), UDP packets (104,048) and other protocol packets (4,440) exchanged within the network, and the total RPL packets overhead which is equal to 6.85 %. The number of other packets for each mote is not shown because Wireshark cannot decode properly the information from the pcap file generated by Cooja.

Network Traffic and RPL Packets Overhead - Benign Network Traffic Dataset				
	Number of RPL Packets	Number of UDP Packets	Number of Other Packets[2]	RPL Packets Overhead [%]
Mote1	290	43,804	-	0.25
Mote2	1,982	11,621	-	1.70
Mote3	1,621	11,883	-	1.39
Mote4	1,604	11,827	-	1.38
Mote5	1,308	12,556	-	1.12
Mote6	1,170	12,357	-	1.00
Total	7,975	104,048	4,440	6.85

Table 5.6 Network Traffic and RPL Packets Overhead – Benign Network Traffic Dataset.

Based on the information included in Table 5.6, the calculated RPL packets overhead per mote and the total RPL packets overhead are depicted in Figure 5.17.

[1] For example, the calculated RPL packets overhead for mote1 is calculated as:

$$\frac{290}{116{,}463} \times 100\% = 0.25\ \%$$

[2] The number of other packets for each mote is not shown because Wireshark cannot decode properly the information from the pcap file generated by Cooja.

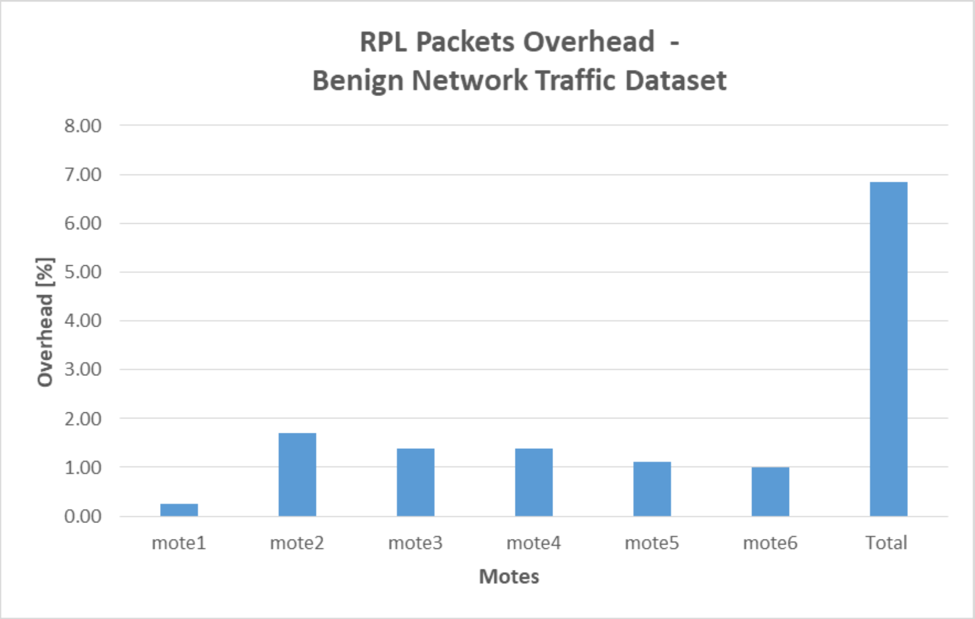


Figure 5.17 RPL Packets Overhead per Mote (%) and Total RPL Packets Overhead (%) – Benign Network Traffic Dataset.

5.3.1.2 UDP Flooding Attack Network Traffic Dataset Analysis

From the generated malicious "udp-flood-radiolog.csv" dataset in Section 4.2.3, Table 5.7 was extracted, demonstrating, in the last column, the percentage of the RPL packets overhead per mote which is calculated as follows: the number of RPL packets per mote over the total number of exchanged packets within the network during the simulation time (702,332 packets). The last row of Table 5.7 contains the total number of RPL packets (9,908), UDP packets (670,671), and other protocol packets (21,753) exchanged within the network, and the total RPL packets overhead which is equal to 1.41 %.

Network Traffic and RPL Packets Overhead – UDP Flooding Attack Network Traffic Dataset				
	Number of RPL Packets	Number of UDP Packets	Number of Other Packets[3]	RPL Packets Overhead [%]
Mote1	203	254,796	-	0.03
Mote2	2,228	28,953	-	0.32
Mote3	2,768	30,238	-	0.39
Mote4	1,976	27,260	-	0.28
Mote5	2,084	31,247	-	0.30
Mote6	6,490	298,177	-	0.09
Total	9,908	670,671	21,753	1.41

Table 5.7 Network Traffic and RPL Packets Overhead – UDP Flooding Attack Network Traffic Dataset.

Based on the information included in Table 5.7 , the calculated RPL packets overhead per mote and the total RPL packets overhead are depicted in Figure 5.18.

[3] The number of other packets for each mote is not shown because Wireshark cannot decode properly the information from the pcap file generated by Cooja.

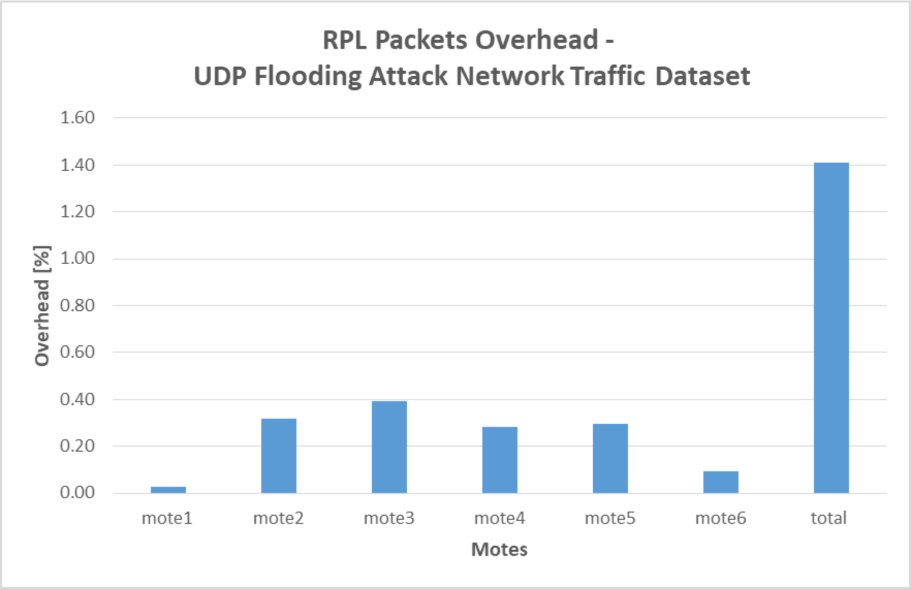


Figure 5.18 RPL Packets Overhead per Mote (%) and Total RPL Packets Overhead (%) – UDP Flooding Attack Network Traffic Dataset.

According to Figure , it is clear that the total RPL packets overhead in the UDP flooding attack scenario (1.41%) is much lower than the total RPL packets overhead in the benign scenario (6.85%) because of the huge amount of UDP packets transmitted by the compromised mote (i.e., mote6) to the target server-mote (i.e., mote1) in the attack scenario. For the same reason, the RPL packets overhead of mote6 in the UDP flooding attack scenario (0.09%) is much less than the corresponding overhead in the benign scenario (1%).

5.3.1.3 Blackhole Attack Network Traffic Dataset Analysis

From the generated malicious “blackhole-radiolog.csv” dataset in Section 4.3.3, Table 5.8 was extracted, demonstrating, in the last column, the percentage of the RPL packets overhead per mote which is calculated as follows: the number of RPL packets per mote over the total number of exchanged packets within the network during the simulation time (99,622 packets). The last row of Table 5.8 contains the total number of RPL packets (24,011), UDP packets (73,551), and other protocol packets (2,060) exchanged within the network, and the total RPL packets overhead which is equal to 24.10 %.

Network Traffic and RPL Packets Overhead – Blackhole Attack Network Traffic Dataset				
	Number of RPL Packets	Number of UDP Packets	Number of Other Packets[4]	RPL Packets Overhead [%]
Mote1	290	19,196	-	0,29
Mote2	4,292	3,821	-	4,31
Mote3	5,341	9,595	-	5,36
Mote4	3,849	10,910	-	3,86
Mote5	2,604	11,756	-	2,61
Mote6	1,433	1,948	-	1,44
Mote7	1,660	3,612	-	1,67
Mote8	1,264	3,779	-	1,27
Mote9	1,580	6,045	-	1,59
Mote10	1,698	2,889	-	1,70
Total	24,011	73,551	2,060	24,10

Table 5.8 Network Traffic and RPL Packets Overhead – Blackhole Attack Network Traffic Dataset.

Based on the information included in Table 5.8 , the calculated RPL packets overhead per mote and the total RPL packets overhead are depicted in Figure 5.19.

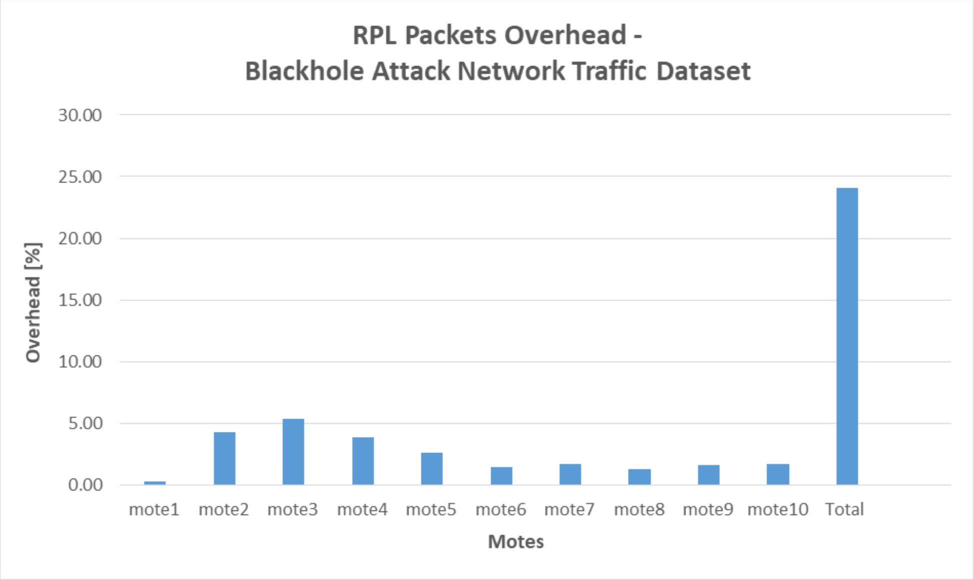


Figure 5.19 RPL Packets Overhead per Mote (%) and Total RPL Packets Overhead (%) – Blackhole Attack Network Traffic Dataset.

According to Figure 5.19, it is clear that the total RPL packets overhead in the blackhole attack scenario (24.10%) is much higher than the total RPL packets overhead in the benign scenario (6.85%) because of the large number of RPL packets transmitted by the motes in the attack scenario as they were trying to re-establish connection with the mote-server due to the impact of the blackhole attack on the network. On top of that, many UDP packets were dropped by the compromised mote (i.e. mote10) instead of being forwarded. It is worthwhile mentioning that we intend, as future work, to generate a network traffic dataset from a benign scenario with 10 motes (i.e., the current one includes 6) as the blackhole attack scenario so that we can get a more accurate value for the total

[4] The number of other packets for each mote is not shown because Wireshark cannot decode properly the information from the pcap file generated by Cooja.

RPL packets overhead in the benign scenario. However, it is expected that value of the total RPL packets overhead in a benign scenario with 10 motes and the same conditions as the benign scenario with 6 motes will be close to the value of the overhead in the scenario with the 6 motes because as the number of UDP packets will be increased due to the 4 more motes, the RPL packets transmitted in the network will be increased analogously.

5.3.1.4 Sinkhole Attack Network Traffic Dataset Analysis

From the generated malicious "sinkhole-radiolog.csv" dataset in Section 4.4.3, Table 5.9 was extracted, demonstrating, in the last column, the percentage of the RPL packets overhead per mote which is calculated as follows: the number of RPL packets per mote over the total number of exchanged packets within the network during the simulation time (463,581 packets). The last row of Table 5.9 contains the total number of RPL packets (404,290), UDP packets (52,750), and other protocol packets (6,541) exchanged within the network, and the total RPL packets overhead which is equal to 87.21 %.

Network Traffic and RPL Packets Overhead – Sinkhole Attack Network Traffic Dataset				
	Number of RPL Packets	Number of UDP Packets	Number of Other Packets[5]	RPL Packets Overhead [%]
Mote1	10,344	14,878	-	2.23
Mote2	56,427	4,130	-	12.17
Mote3	46,048	3,864	-	9.93
Mote4	52,087	5,279	-	11.24
Mote5	46,576	3,916	-	10.05
Mote6	43,657	4,643	-	9.42
Mote7	44,872	5,642	-	9.68
Mote8	46,974	4,282	-	10.13
Mote9	46,788	6,116	-	10.09
Mote10	10,517	0	-	2.27
Total	404,290	52,750	6,541	87.21

Table 5.9 Network Traffic and RPL Packets Overhead – Sinkhole Attack Network Traffic Dataset.

Based on the information included in Table 5.9 , the calculated RPL packets overhead per mote and the total RPL packets overhead are depicted in Figure 5.20.

[5] The number of other packets for each mote is not shown because Wireshark cannot decode properly the information from the pcap file generated by Cooja.

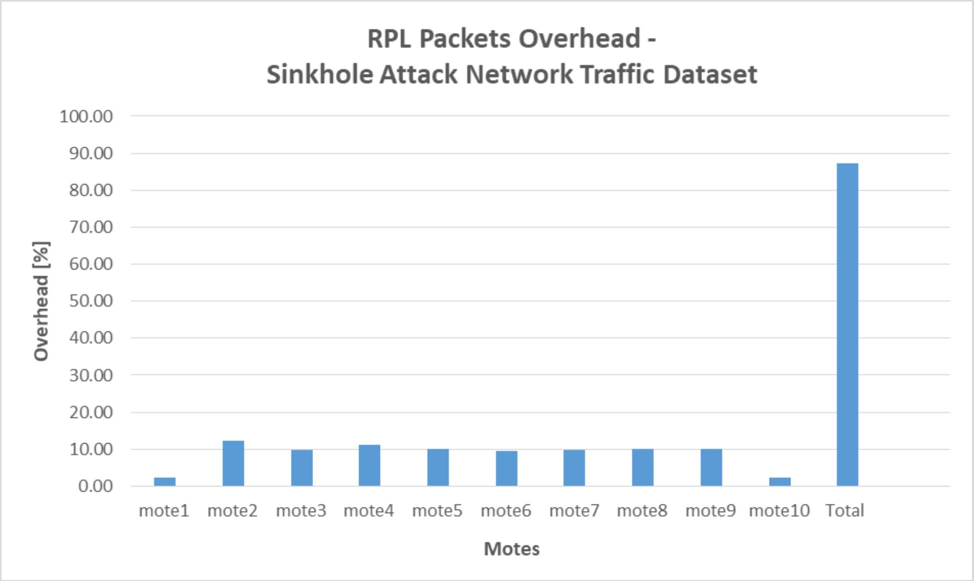


Figure 5.20 RPL Packets Overhead per Mote (%) and Total RPL Packets Overhead (%) – Sinkhole Attack Network Traffic Dataset.

According to Figure 5.20, it is clear that the total RPL packets overhead in the sinkhole attack scenario (87.21%) is significantly higher than the total RPL packets overhead in the benign scenario (6.85%) because of the huge number of RPL packets transmitted by the motes in the attack scenario as they were trying to respond to the impact of the sinkhole attack on the network. In addition, many UDP packets were dropped by the compromised mote (i.e. mote10) instead of being forwarded. It is worthwhile mentioning that we intend, as future work, to generate a network traffic dataset from a benign scenario with 10 motes (i.e., the current one includes 6) as the sinkhole attack scenario so that we can get a more accurate value for the total RPL packets overhead in the benign scenario. However, it is expected that value of the total RPL packets overhead in a benign scenario with 10 motes and the same conditions as the benign scenario with 6 motes will be close to the value of the overhead in the scenario with the 6 motes because as the number of UDP packets will be increased due to the 4 more motes, the RPL packets transmitted in the network will be increased analogously.

5.3.1.5 Sleep Deprivation Attack Network Traffic Dataset Analysis

From the generated malicious “sleep_depr-radiolog.csv” dataset in Section 4.5.3, was extracted, demonstrating, in the last column, the percentage of the RPL packets overhead per mote which is calculated as follows: the number of RPL packets per mote over the total number of exchanged packets within the network during the simulation time (571,079 packets). The last row of Table 5.10 contains the total number of RPL packets (30,338), UDP packets (526,799), and other protocol packets (13,942) exchanged within the network, and the total RPL packets overhead which is equal to 5.31 %.

Network Traffic and RPL Packets Overhead – Sleep Deprivation Attack Network Traffic Dataset				
	Number of RPL Packets	Number of UDP Packets	Number of Other Packets[6]	RPL Packets Overhead [%]
Mote1	261	237,640	-	0.05
Mote2	3,288	2,782	-	0.58
Mote3	2,709	3,075	-	0.47
Mote4	2,063	4,531	-	0.36
Mote5	5,550	4,256	-	0.97
Mote6	2,936	8,322	-	0.51
Mote7	2,617	9,595	-	0.46
Mote8	3,936	13,000	-	0.69
Mote9	6,248	10,708	-	1.09
Mote10	730	232,890	-	0.13
Total	30,338	526,799	13,942	5.31

Table 5.10 Network Traffic and RPL Packets Overhead – Sleep Deprivation Attack Network Traffic Dataset.

Based on the information included in Table 5.10 , the calculated RPL packets overhead per mote and the total RPL packets overhead are depicted in Figure 5.24.

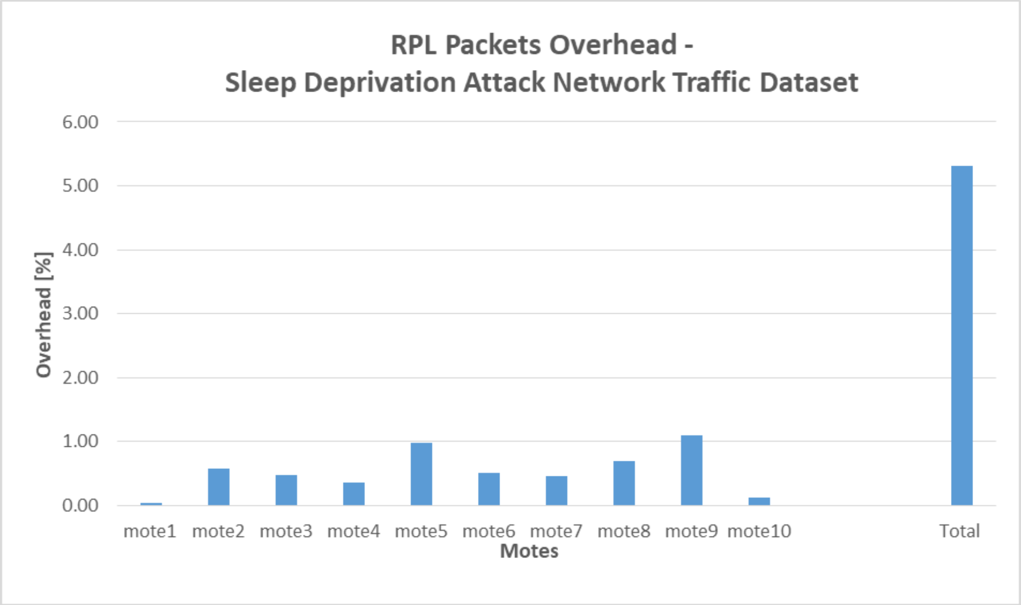


Figure 5.21 RPL Packets Overhead per Mote (%) and Total RPL Packets Overhead (%) – Sleep Deprivation Attack Network Traffic Dataset.

According to Figure 5.21, the total RPL packets overhead in the sleep deprivation attack scenario (5.31%) is lower than the total RPL packets overhead in the benign scenario (6.85%) because of the large number of UDP packets transmitted by the compromised mote (i.e., mote10) to the target client-mote (i.e., mote4). It is worthwhile mentioning that we intend, as future work, to generate a network traffic dataset from a benign scenario with 10 motes (i.e., the current one includes 6) as the sleep deprivation attack scenario so that we can get a more accurate value for the total RPL packets overhead in the benign scenario. However, it is expected that value of the total RPL packets

[6] The number of other packets for each mote is not shown because Wireshark cannot decode properly the information from the pcap file generated by Cooja.

overhead in a benign scenario with 10 motes and the same conditions as the benign scenario with 6 motes will be close to the value of the overhead in the scenario with the 6 motes because as the number of UDP packets will be increased due to the 4 more motes, the RPL packets transmitted in the network will be increased analogously.

5.4 Summary

This Chapter was focused on the analysis of the generated benign "powertrace" and network traffic datasets, presented in Chapter 3, and the generated malicious "powertrace" and network traffic datasets, demonstrated in Chapter 4. The Chapter started with the analysis of the malicious "powertrace" datasets to investigate whether their raw features can be important in the detection of anomalies in the network-level power profiling of low-power IoT devices (i.e., motes) due to UDP flooding attacks, blackhole attacks, sinkhole attacks, or sleep deprivation attacks. Towards this direction, all malicious "powertrace" datasets were pre-processed before applying the MI method to measure the importance of the different features of each malicious "powertrace" dataset (i.e., "udp-flood-pwrtrace.csv", "blackhole-pwrtrace.csv", "sinkhole-pwrtrace.csv", and "sleep_depr-pwrtrace.csv") and identify the most significant features. In addition, the average values of the most significant features, based on MI, were calculated. Based on the results and the observations in Section 5.2.1, the following 5 features have been identified as the most important for all malicious "powertrace" datasets: "transmit", "cpu", "lpm", "listen", and "idle_listen".

Next, the Chapter continued with investigating the extraction of new features, more informative and non-redundant, based on the raw features of the generated benign and malicious "powertrace" datasets and the generated benign and malicious network traffic datasets. To this end, the total energy consumption of each mote in an IoT network was investigated in Section 5.2.2 as a valuable feature for training and evaluating IoT AIDSs. According to the observations and conclusions in Section 5.2.2, the total energy consumption of each mote in an IoT network can play a valuable role in anomaly-based intrusion detection for the following types of attacks in IoT networks: UDP flooding attack, blackhole attack, sinkhole attack, and sleep deprivation attack. This is because any observation considerably deviating from the normal total energy consumption, and particularly the total CPU energy consumption and the total Radio (i.e., TX+RX) energy consumption per mote, can be considered as an anomalous behaviour, triggering alerts so that proper countermeasures can be taken to minimise the risk. On the other hand, the generated benign and malicious network traffic datasets were also analysed in Section 5.3.1 and the new feature that was extracted was the "RLP packets overhead". This new feature provides information about the number of RPL packets (per mote and total) transmitted over the total number of exchanged messages within the IoT network, indicating a blackhole or sinkhole attack when its value is high and a UDP flooding attack or sleep deprivation attack when its value is low.

Chapter 6 Datasets Validation

6.1 Introduction

This Chapter is focused on the validation of the generated malicious “powertrace” datasets, presented in Chapter 4, by applying different Machine Learning (ML) algorithms for IoT AIDSs to evaluate their performance on the generated malicious datasets. In particular, the following most popular ML algorithms for IoT AIDSs, reviewed in Section 2.3, were applied: naïve Bayes (NB), decision tree (DT), random forest (RF), logistic regression (LR), support vector machines (SVM), and k-nearest neighbor (KNN). Using five-fold cross validation, these algorithms were trained and tested over the same labelled dataset for each attack scenario. Furthermore, the following four traditional metrics, reviewed in Section 2.4, were used to evaluate the performance of the ML algorithms on the generated datasets when these algorithms are used for anomaly detection in IoT AIDSs: accuracy, precision, recall, and F1-score. In all experiments, the Python language (version 3.8.2) was used, along with the Scikit-Learn library [27] and a Python script created, utilizing specific functions of the Scikit-Learn library, to perform training and testing of the ML algorithms.

6.2 Dataset Pre-Processing

The pre-processing phase involved the removal of unnecessary features from the four malicious “powertrace” datasets (i.e., “udp-flood-pwrtrace.csv”, “blackhole-pwrtrace.csv”, “sinkhole-pwrtrace.csv”, and “sleep_depr-pwrtrace.csv”) and the addition of the “label” feature (i.e., “0” for normal and “1” for malicious) to all of them. In particular, the feature “Clock_time” was filtered out along with the features related to the simulation time (i.e., “sim time”) and the simulation duration (i.e., “all_cpu”, “all_lpm”, “all_transmit”, “all_listen”, ”all_idle_transmit”, “all_idle_listen”) and the “seq no” feature. Besides that, the “P” feature was omitted, because it only has a fixed value throughout all of the collected records of the malicious “powertrace” datasets. Moreover the “ID” and “Rime Address” were also filtered out because it was observed that they led to overfitting. Last but not least, the “idle_transmit” feature was filtered out as well, because it had the lowest calculated importance, based on the “label” feature, by applying the MI method for all malicious “powertrace” datasets. After the pre-processing phase, the new labelled malicious “powertrace” datasets were named as “udp-flood-pwrtrace_label.csv”, “blackhole-pwrtrace_label.csv”, “sinkhole-pwrtrace_label.csv”, and “sleep_depr-pwrtrace_label.csv”, and contained the following features: “cpu”, “lpm”, “transmit”, “listen”, and “idle_listen”.

6.3 Normalisation

The data normalization step was performed to the numeric values of each feature. If the values of a feature are significantly larger compared to the values of other features, this may lead to inaccurate results. Thus, data normalisation helps to ensure that features with significantly large values do not outweigh features with smaller values. To achieve this, all of the features’ values are scaled within the range of [0.0, 1.0] by performing a min–max normalization process on each feature. This normalization process is described by the following equation:

$$z = (x - x_{min})/(x_{max} - x_{min}) \qquad (6.1)$$

where z is the normalized value (i.e., after scaling), x is the value before scaling, and x_{max} and x_{min} are the maximum and minimum values of the feature, respectively.

6.4 Training and Testing of ML Algorithms on the Malicious "Powertrace" Datasets

The selected ML algorithms were trained and tested separately over the four labelled malicious "powertrace" datasets: "udp-flood-pwrtrace_label.csv", "blackhole-pwrtrace_label.csv", "sinkhole-pwrtrace_label.csv", and "sleep_depr-pwrtrace_label.csv". Initially, each of the four datasets was split into two parts: the train part and the test part. The train part consisted of 80% of the dataset and the ML algorithms were trained and evaluated with this part. On the other hand, the test part consisted of 20% of the dataset and was held back for further evaluation of the models with unseen data. The percentage split of 80% train–20% test was determined according to [72] as the best ratio to avoid the overfitting problem. After that, the training process of each ML algorithm over each dataset was performed using the five-fold cross validation method. According to this method, the training dataset was divided into five subsets of equal size and the records of each subset were randomly selected. The training process was repeated five times. Each time, four out of the five subsets were utilized for the training of the ML algorithm and the remaining subset was used for validation. The final performance results were produced by averaging the results of the five folds [72]. Table 6.1 presents a summary of the set hyperparameters of each of the six ML algorithms.

ML Algorithm	Hyperparameters
Decision Tree (DT)	• The Gini index was used to select tree nodes. • Minimum samples per leaf node set to 10
Naïve Bayes (NB)	• The Gaussian variant of the NB algorithm was used.
Logistic Regression (LR)	-
Random Forest (RF)	• The Gini index was used to select tree nodes. • The minimum number of samples per leaf node was set to 10. • The random forest consisted of 10 decision trees.
K-Nearest Neighbour (KNN)	• The value of K was set to 5. • The Euclidean distance was set as the distance metric.
Support Vector Machine (SVM)	• The Gaussian radial basis function (RBF) was set as the kernel function.

Table 6.1 Summary of the hyperparameters of each selected ML algorithm.

6.5 Performance Evaluation Results

6.5.1 “udp-flood-pwrtrace_label.csv” Dataset

The selected ML algorithms were trained and tested on the “udp-flood-pwrtrace_label.csv” dataset for binary classification, using the five-fold cross validation method. The performance of the selected ML algorithms was evaluated by the evaluation metrics of accuracy, precision, recall, and F1-score. The numerical results of the evaluation metrics for the selected ML algorithms, when applied to the “udp-flood-pwrtrace_label.csv”, are shown in Table 6.2 and Figure 6.1.

ML Algorithm	Accuracy	Precision	Recall	F1-score
Decision Tree (DT)	0.9818	0.9509	0.9396	0.9451
Naïve Bayes (NB)	0.9148	0.6774	0.9354	0.7855
Logistic Regression (LR)	0.9742	0.9333	0.9104	0.9216
Random Forest (RF)	0.9885	0.9739	0.9569	0.9653
K-Nearest Neighbor (KNN)	0.9931	0.9853	0.9729	0.9790
Support Vector Machine (SVM)	0.9890	0.9773	0.9562	0.9666

Table 6.2 Evaluation metrics for binary classification for the “udp-flood-pwrtrace_label.csv” dataset.

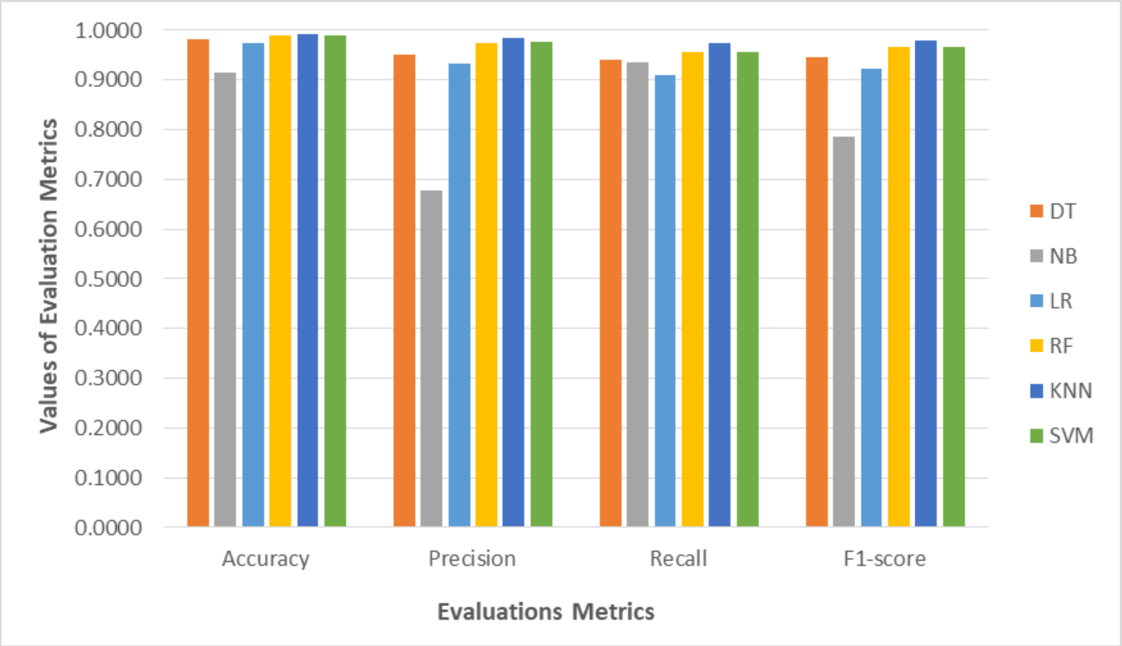


Figure 6.1 Evaluation metrics for binary classification for the “udp-flood-pwrtrace_label.csv” dataset.

It is observed that the KNN, SVM and RF algorithms demonstrate an almost perfect accuracy score (i.e., around 0.99), followed by the DT and LR (i.e., close to 0.98). The same trend can be seen in the precision, recall, and F1-score, as the KNN, SVM and RF algorithms show high values between 0.95 – 0-99, while the DT and LR classifiers demonstrate values between 0.91-0.96. On the other hand, although the NB achieves accuracy and recall higher than 0.91, it shows the lowest precision of 0.6774 and the lowest F1-score of 0.7855.

6.5.2 “blackhole-pwrtrace_label.csv” Dataset

The selected ML algorithms were trained and tested on the “blackhole-pwrtrace_label.csv” dataset for binary classification, using the five-fold cross validation method. The performance of the selected ML algorithms was evaluated by the evaluation metrics of accuracy, precision, recall, and F1-score. The numerical results of the evaluation metrics for the selected ML algorithms, when applied to the “blackhole-pwrtrace_label.csv”, are shown in Table 6.3 and Figure 6.2.

ML Algorithm	Accuracy	Precision	Recall	F1-score
Decision Tree (DT)	1.0000	1.0000	1.0000	1.0000
Naïve Bayes (NB)	0.9999	1.0000	0.9976	0.9988
Logistic Regression (LR)	1.0000	1.0000	1.0000	1.0000
Random Forest (RF)	1.0000	1.0000	1.0000	1.0000
K-Nearest Neighbor (KNN)	1.0000	1.0000	1.0000	1.0000
Support Vector Machine (SVM)	1.0000	1.0000	1.0000	1.0000

Table 6.3 Evaluation metrics for binary classification for the “blackhole-pwrtrace_label.csv” dataset.

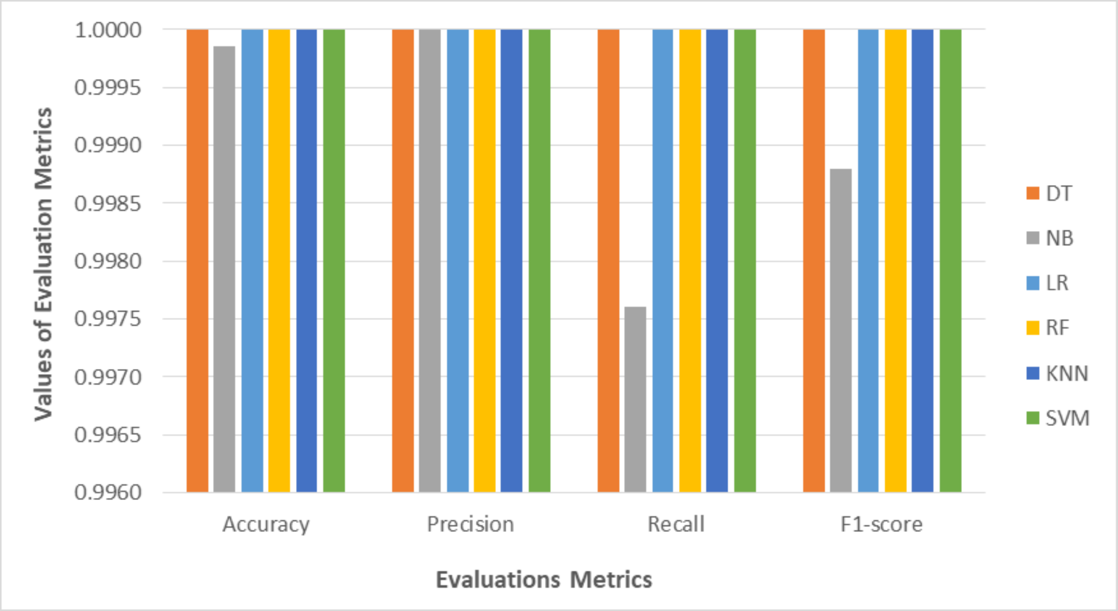


Figure 6.2 Evaluation metrics for binary classification for the “blackhole-pwrtrace_label.csv” dataset.

It can be easily observed that the KNN, RF, SVM, DT and LR algorithms achieve perfect accuracy, precision, recall, and F1-score, while the NB algorithm achieves an almost perfect performance.

6.5.3 “sinkhole-pwrtrace_label.csv” Dataset

The selected ML algorithms were trained and tested on the “sinkhole-pwrtrace_label.csv” dataset for binary classification, using the five-fold cross validation method. The performance of the selected ML algorithms was evaluated by the evaluation metrics of accuracy, precision, recall, and F1-score. The numerical results of the evaluation metrics for the selected ML algorithms, when applied to the “sinkhole-pwrtrace_label.csv”, are shown in Table 6.4 and Figure 6.3.

ML Algorithm	Accuracy	Precision	Recall	F1-score
Decision Tree (DT)	0.9517	0.6836	0.5578	0.6138
Naïve Bayes (NB)	0.9062	0.0414	0.1277	0.0625
Logistic Regression (LR)	0.9304	0.0667	0.0010	0.0021
Random Forest (RF)	0.9545	0.7560	0.5005	0.6017
K-Nearest Neighbor (KNN)	0.9367	0.5630	0.4035	0.4685
Support Vector Machine (SVM)	0.9311	0.0000	0.0000	0.0000

Table 6.4 Evaluation metrics for binary classification for the “sinkhole-pwrtrace_label.csv” dataset.

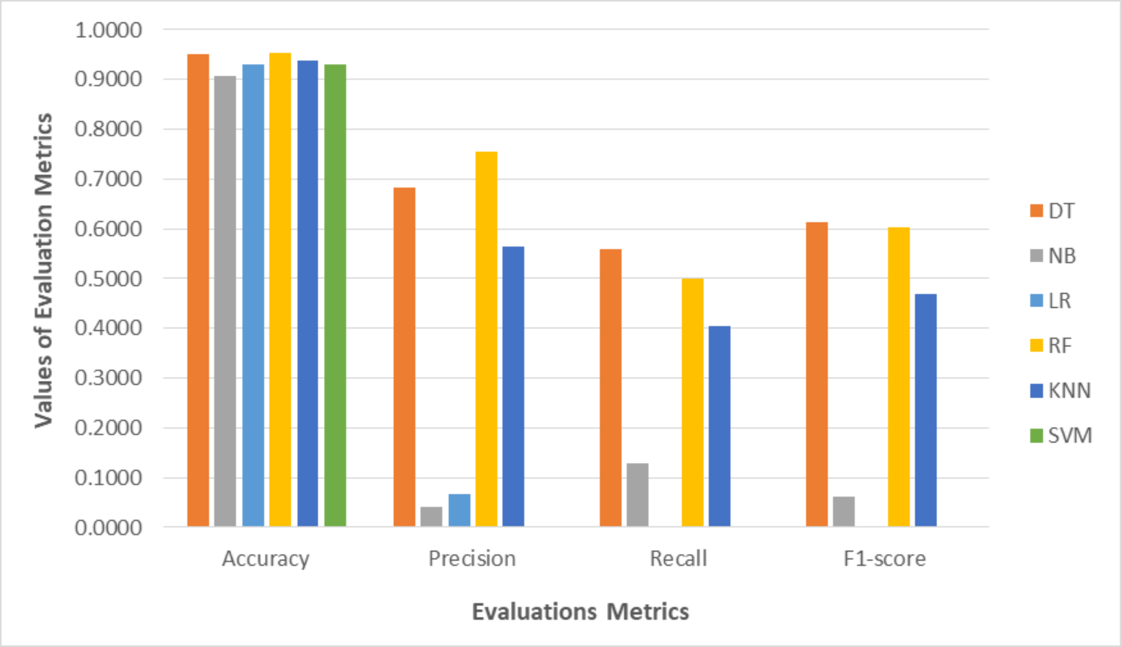


Figure 6.3 Evaluation metrics for binary classification for the “sinkhole-pwrtrace_label.csv” dataset.

It is observed that all algorithms demonstrate high accuracy, with the lowest accuracy (0.9062) being achieved by NB and the highest (0.9545) being achieved by the RF classifier, followed by the DT and KNN. On the other hand, in principle, the performance of all algorithms in terms of precision, recall, and F1-score is very low. The highest precision of 0.7560 was achieved by the RF and the highest recall and F1-score by the DT, 0.5578 and 0.6138, respectively. Moreover, it is worthwhile mentioning that the SVM shows the lowest precision, recall, and F1-score of 0. This is because the precision was ill-defined (i.e., division by zero).

6.5.4 “sleep_depr-pwrtrace_label.csv” Dataset

The selected ML algorithms were trained and tested on the “sleep_depr-pwrtrace_label.csv” dataset for binary classification, using the five-fold cross validation method. The performance of the selected ML algorithms was evaluated by the evaluation metrics of accuracy, precision, recall, and F1-score. The numerical results of the evaluation metrics for the selected ML algorithms, when applied to the “sleep_depr-pwrtrace_label.csv”, are shown in Table 6.5 and Figure 6.4.

ML Algorithm	Accuracy	Precision	Recall	F1-score
Decision Tree (DT)	0.9739	0.8143	0.7402	0.7749
Naïve Bayes (NB)	0.9034	0.3506	0.6403	0.4476
Logistic Regression (LR)	0.9478	0.6552	0.3075	0.4173
Random Forest (RF)	0.9766	0.8559	0.7402	0.7937
K-Nearest Neighbor (KNN)	0.9759	0.8439	0.7401	0.7885
Support Vector Machine (SVM)	0.9393	0.5000	0.0036	0.0071

Table 6.5 Evaluation metrics for binary classification for the “sleep_depr-pwrtrace_label.csv” dataset.

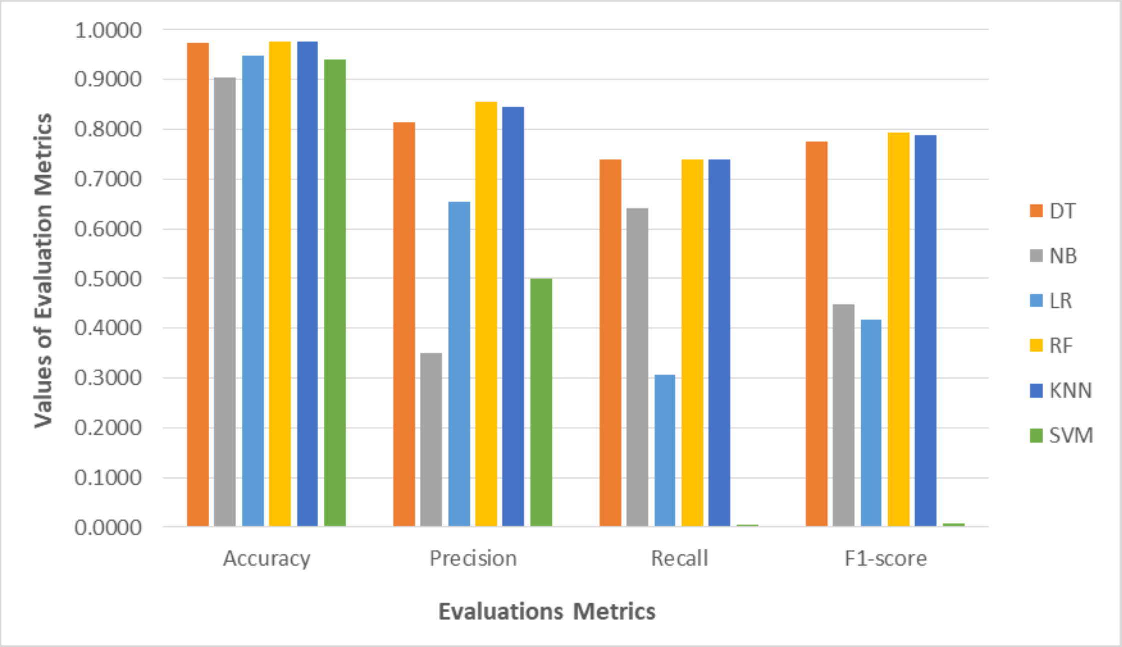


Figure 6.4 Evaluation metrics for binary classification for the “sleep_depr-pwrtrace_label.csv” dataset.

It is observed that all algorithms demonstrate high accuracy, with the lowest accuracy (0.9034) being achieved by NB and the highest (0.9766) being achieved by the RF classifier, followed by the KNN (0.9759) and DT (0.9739). In terms of precision, the top three values were achieved by the RF, KNN and DT, and the lowest by the SVM. Furthermore, the RF, KNN, and DT outperform the other algorithms in terms of recall and F1-score.

6.5 Summary

This Chapter was focused on the validation of the generated malicious “powertrace” datasets, presented in Chapter 4, by applying the following most popular ML algorithms for IoT AIDS to evaluate their performance on the generated malicious datasets: naïve Bayes (NB), decision tree (DT), random forest (RF), logistic regression (LR), support vector machines (SVM), and k-nearest neighbour (KNN). Using five-fold cross validation, these algorithms were trained and tested over the same labelled dataset for each attack scenario. Furthermore, the traditional metrics of accuracy, precision, recall, and F1-score were used to evaluate the performance of the ML algorithms on the generated datasets. The evaluations results demonstrated that the RF, KNN, and DT algorithms presented very high values regarding accuracy (between 0.93 and 1.0) and outperform the other algorithms regarding precision, recall and F1-score for all malicious datasets. In particular, it is worthwhile mentioning that the RF, KNN, and DT algorithms achieved precision between 0.84 and 1.0 for the “udp-flood-pwrtrace_label.csv”, “blackhole-pwrtrace_label.csv”, and the “sleep_depr-pwrtrace_label.csv”. In principle, the evaluation results demonstrated that the generated malicious datasets can be used for training and testing effectively ML algorithms for IoT AIDSs.

This page intentionally left blank.

Chapter 7 Conclusion and Future Work

7.1 Conclusions

The focus of this PhD research work was on the generation of new labelled IoT datasets that will be publicly available to the research community and include the following required information so as to be considered as benchmark IoT datasets for training and evaluating Machine Learning models for IoT AIDSs: (a) **information reflecting multiple benign and attack scenarios from current IoT network environments**, (b) **sensor measurement data**, (c) **network-related information (e.g., packet-level information) from IoT networks**, and (d) **information related to the behaviour of the IoT devices deployed within IoT networks**. It is worthwhile mentionioning that the new labelled IoT datasets were generated by implementing various benign IoT network scenarios and IoT network attack scenarios in the Cooja simulator which is the companion network simulator of the open source Contiki Operating System (OS) which is one of the most popular OSs for resource constrained IoT devices. To the best of our knowledge, this is the first time that the Cooja simulator is used, in a systematic way, to generate benchmark IoT datasets. The new labelled IoT datasets generated by the Cooja simulator are not to be considered as a replacement of datasets captured from real IoT networks or real IoT testbeds, but instead to be considered as complementary datasets that will contribute to fill the current gap of the scarcity of benchmark datasets for training and evaluating Machine Learning models for IoT AIDSs. Furthermore, the generated datasets were analysed to select important raw features for the detection of anomalies as well as extract new features, more informative and non-redundant, based on the raw features. Finally, different Machine Learning (ML) algorithms for IoT AIDSs were applied to evaluate their performance on the generated malicious datasets and validate that the generated malicious datasets can be used for training and testing effectively ML algorithms for IoT AIDSs.

The main contribution of this PhD research work is summarised as follows.

- Generation of a set of benign IoT datasets from a benign IoT network scenario implemented in the Cooja simulator. The generated datasets constitute the benign IoT datasets for the simulated benign IoT network scenario. Furthermore, a detailed description of the approach proposed to generate the set of benign IoT datasets has also been provided. In addition, it is worthwhile mentioning that the proposed approach can be extended for generating benign IoT datasets from ***j*** different benign scenarios, where each scenario, implemented in the Cooja simulator, may include ***n*** different motes. The generic structure of the benign IoT datasets generated according to the proposed approach has been provided and constitutes a roadmap for generating more and richer benign IoT datasets.
- Generation of a set of malicious datasets from the following attack scenarios implemented in the Cooja simulator: i) UDP flooding attack, ii) blackhole attack, iii) sinkhole attack, and iv) sleep deprivation attack. The generated datasets constitute the malicious IoT datasets for the simulated IoT attack scenarios. Moreover, a detailed description of the approach proposed to generate the set of the malicious IoT datasets has also been given. On top of that, it is important to highlight that the proposed approach can be extended for generating malicious IoT datasets from ***j*** different attack scenarios of ***i*** different attack types, where each attack scenario, implemented in the Cooja simulator, may include ***n*** different motes. The generic structure of the malicious IoT datasets generated according to the proposed approach has been provided and constitutes a roadmap for generating more and richer malicious IoT datasets.

- Analysis of the malicious “powertrace” datasets to investigate whether their raw features can be important in the detection of anomalies in the network-level power profiling of low-power IoT devices (i.e., motes) due to UDP flooding attacks, blackhole attacks, sinkhole attacks, or sleep deprivation attacks. According to the analysis, the following 5 features have been identified as the most important for all malicious “powertrace” datasets: “transmit”, “cpu”, “lpm”, “listen”, and “idle_listen”.

- Extraction of new features, more informative and non-redundant, based on the raw features of the generated benign and malicious “powertrace” datasets. To this end, the total energy consumption of each mote in an IoT network was investigated as a valuable feature for training and evaluating IoT AIDSs. According to the observations and conclusions, the total energy consumption of each mote in an IoT network can play a valuable role in anomaly-based intrusion detection for the following types of attacks in IoT networks: UDP flooding attack, blackhole attack, sinkhole attack, and sleep deprivation attack. This is because any observation considerably deviating from the normal total energy consumption, and particularly the total CPU energy consumption and the total Radio energy consumption per mote, can be considered as an anomalous behaviour, triggering alerts so that proper countermeasures can be taken to minimise the risk.

- Extraction of new features, more informative and non-redundant, based on the raw features of the generated benign and malicious network traffic datasets. The generated benign and malicious network traffic datasets were analysed and the new feature that was extracted was the “RPL packets overhead”. This new feature provides information about the number of RPL packets (per mote and total) transmitted over the total number of exchanged messages within the IoT network, indicating a blackhole or sinkhole attack when its value is high and a UDP flooding attack or sleep deprivation attack when its value is low.

- Validation of the generated malicious “powertrace” datasets by applying the following most popular ML algorithms for IoT AIDS to evaluate their performance on the generated malicious datasets: naïve Bayes (NB), decision tree (DT), random forest (RF), logistic regression (LR), support vector machines (SVM), and k-nearest neighbour (KNN). Using five-fold cross validation, these algorithms were trained and tested over the same labelled dataset for each attack scenario. Furthermore, the traditional metrics of accuracy, precision, recall, and F1-score were used to evaluate the performance of the ML algorithms on the generated datasets. The evaluations results demonstrated that the generated malicious datasets can be used for training and testing effectively ML algorithms for IoT AIDSs.

7.2 Future Work

This thesis laid the foundation for future research efforts towards the generation of rich benchmark IoT datasets for effective training and evaluation of different ML models for IoT AIDSs by implementing various benign IoT network scenarios and IoT network attack scenarios in the Cooja simulator. In this context, considering the generic structure of the benign IoT datasets, proposed in Chapter 3, as a roadmap for generating more and richer benign IoT datasets, we plan to continue generating more benign IoT datasets from a wide spectrum of different benign IoT scenarios, where each scenario, implemented in the Cooja simulator, will include a different number of motes. Furthermore, considering the generic structure of the malicious IoT datasets, proposed in Chapter 4, as a roadmap for generating more and richer malicious IoT datasets, we will continue generating more malicious IoT datasets from several different IoT attack scenarios of different attack types, where each attack scenario, implemented in the Cooja simulator, will include a different number of motes. In particular, additional attack scenarios for each of the four attack types considered in this PhD work (i.e., UDP flooding attack, blackhole attack, sinkhole attack, and sleep deprivation attack) can be implemented in the Cooja simulator, examining with different number of motes and configuring different topologies. Following the research methodology defined in this PhD work, the newly implemented scenarios will contribute to more and richer IoT datasets.

Besides that, different feature selection techniques will be applied on the generated IoT datasets to identify those raw features that are important in the detection of anomalies in IoT networks and devices deployed in these networks due to IoT attacks. On top of that, we will continue with the extraction of new features, more informative and non-redundant, based on the raw features of the generated benign and malicious "powertrace" datasets, and the generated benign and malicious network traffic datasets. The target is to identify and/or extract a rich set of very informative and non-redundant features that will allow not only the detection of anomalies due to IoT attacks but also the identification of the type of the attack causing the detected anomalies. Last but not least, the validation of the generated datasets by applying different ML algorithms to evaluate their performance on the generated datasets, based on the original (i.e., raw) set of features, the subset of the selected features, and/or the new extracted features, is of utmost importance. In fact, it constitutes the essential final step where the performance evaluation results will indicate whether or not the generated datasets meet the requirements of benchmark IoT datasets for training and evaluation of various ML models for IoT AIDSs.

This page intentionally left blank.

References

[1] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, "A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1125–1142, Oct. 2017.

[2] M. binti Mohamad Noor and W. H. Hassan, "Current research on Internet of Things (IoT) security: A survey," *Comput. Networks*, vol. 148, pp. 283–294, Jan. 2019.

[3] V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal, and B. Sikdar, "A Survey on IoT Security: Application Areas, Security Threats, and Solution Architectures," *IEEE Access*, vol. 7. Institute of Electrical and Electronics Engineers Inc., pp. 82721–82743, 2019.

[4] Y. Lu and L. Da Xu, "Internet of things (IoT) cybersecurity research: A review of current research topics," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2103–2115, Apr. 2019.

[5] S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Porisini, "Security, privacy and trust in Internet of things: The road ahead," *Computer Networks*, vol. 76. Elsevier B.V., pp. 146–164, 15-Jan-2015.

[6] I. Makhdoom, M. Abolhasan, J. Lipman, R. P. Liu, and W. Ni, "Anatomy of Threats to the Internet of Things," *IEEE Commun. Surv. Tutorials*, vol. 21, no. 2, pp. 1636–1675, Apr. 2019.

[7] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, "Industrial Internet of Things: Challenges, Opportunities, and Directions," *IEEE Trans. Ind. Informatics*, vol. 14, no. 11, pp. 4724–4734, Nov. 2018.

[8] P. P. Ray, "A survey on Internet of Things architectures," *Journal of King Saud University - Computer and Information Sciences*, vol. 30, no. 3. King Saud bin Abdulaziz University, pp. 291–319, 01-Jul-2018.

[9] E. L. C. Macedo *et al.*, "On the security aspects of Internet of Things: A systematic literature review," *Journal of Communications and Networks*, vol. 21, no. 5. Korean Institute of Communication Sciences, pp. 444–457, 01-Oct-2019.

[10] B. B. Zarpelão, R. S. Miani, C. T. Kawakani, and S. C. de Alvarenga, "A survey of intrusion detection in Internet of Things," *Journal of Network and Computer Applications*, vol. 84. Academic Press, pp. 25–37, 15-Apr-2017.

[11] K. A. P. da Costa, J. P. Papa, C. O. Lisboa, R. Munoz, and V. H. C. de Albuquerque, "Internet of Things: A survey on machine learning-based intrusion detection approaches," *Comput. Networks*, vol. 151, pp. 147–157, Mar. 2019.

[12] A. Khraisat and A. Alazab, "A critical review of intrusion detection systems in the internet of things: techniques, deployment strategy, validation strategy, attacks, public datasets and challenges," *Cybersecurity*, vol. 4, no. 1, pp. 1–27, Dec. 2021.

[13] N. Chaabouni, M. Mosbah, A. Zemmari, C. Sauvignac, and P. Faruki, "Network Intrusion Detection for IoT Security Based on Learning Techniques," *IEEE Commun. Surv. Tutorials*, vol. 21, no. 3, pp. 2671–2701, Jul. 2019.

[14] A. Alsaedi, N. Moustafa, Z. Tari, A. Mahmood, and A. Anwar, "TON_IoT Telemetry Dataset: A New Generation Dataset of IoT and IIoT for Data-Driven Intrusion Detection Systems," *IEEE Access*, vol. 8, pp. 165130–165150, Sep. 2020.

[15] T. M. Booij, I. Chiscop, E. Meeuwissen, N. Moustafa, and F. T. H. den Hartog, "ToN_IoT: The

Role of Heterogeneity and the Need for Standardization of Features and Attack Types in IoT Network Intrusion Datasets," *IEEE Internet Things J.*, 2021.

[16] "KDD Cup 1999 Data." [Online]. Available: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

[17] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *IEEE Symposium on Computational Intelligence for Security and Defense Applications, CISDA 2009*, 2009, pp. 1–6.

[18] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *2015 Military Communications and Information Systems Conference, MilCIS 2015*, 2015, pp. 1–6.

[19] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization," in *ICISSP2018*, 2018, pp. 108–116.

[20] S. Suthaharan, M. Alzahrani, S. Rajasegarar, C. Leckie, and M. Palaniswami, "Labelled data collection for anomaly detection in wireless sensor networks," in *Proceedings of the 2010 6th International Conference on Intelligent Sensors, Sensor Networks and Information Processing, ISSNIP 2010*, 2010, pp. 269–274.

[21] A. Sivanathan *et al.*, "Classifying IoT Devices in Smart Environments Using Network Traffic Characteristics," *IEEE Trans. Mob. Comput.*, vol. 18, no. 8, pp. 1745–1759, Aug. 2019.

[22] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset," *Futur. Gener. Comput. Syst.*, vol. 100, pp. 779–796, Nov. 2019.

[23] A. Hamza, H. H. Gharakheili, T. A. Benson, and V. Sivaraman, "Detecting Volumetric Attacks on IoT Devices via SDN-Based Monitoring of MUD Activity," in *SOSR 2019 - Proceedings of the 2019 ACM Symposium on SDN Research*, 2019, pp. 36–48.

[24] M. Chernyshev, Z. Baig, O. Bello, and S. Zeadally, "Internet of things (IoT): Research, simulators, and testbeds," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 1637–1647, Jun. 2018.

[25] F. Österlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt, "Cross-level sensor network simulation with COOJA," in *Proceedings - Conference on Local Computer Networks, LCN*, 2006, pp. 641–648.

[26] I. Essop, J. C. Ribeiro, M. Papaioannou, G. Zachos, G. Mantas, and J. Rodriguez, "Generating Datasets for Anomaly-Based Intrusion Detection Systems in IoT and Industrial IoT Networks," *Sensors*, vol. 21, no. 4, 2021.

[27] "scikit-learn: machine learning in Python — scikit-learn 1.0.1 documentation." [Online]. Available: https://scikit-learn.org/stable/. [Accessed: 19-Dec-2021].

[28] ITU-T, "Y.4101 : Common requirements and capabilities of a gateway for Internet of things applications," 2017.

[29] ENISA, "Baseline Security Recommendations for IoT — ENISA," 2017.

[30] W. Stallings and L. Brown, *Computer Security: Principles and Practice*, 4th ed. Pearson Education, 2018.

[31] ITU-T, "Recommendation ITU-T Y.2060 'Overview of the Internet of Things,'" 2012.

[32] ITU, "The Internet of Things - Executive Summary," 2005.

[33] D. Evans, "The Internet of Things: How the Next Evolution of the Internet Is Changing Everything," 2011.

[34] ENISA, "GUIDELINES FOR SECURING THE INTERNET OF THINGS: Secure supply chain for IoT," 2020.

[35] R. Mahmoud, T. Yousuf, F. Aloul, and I. Zualkernan, "Internet of things (IoT) security: Current status, challenges and prospective measures," in *2015 10th International Conference for Internet Technology and Secured Transactions, ICITST 2015*, 2016, pp. 336–341.

[36] Q. Qi and F. Tao, "A Smart Manufacturing Service System Based on Edge Computing, Fog Computing, and Cloud Computing," *IEEE Access*, vol. 7, pp. 86769–86777, 2019.

[37] U. Satija, B. Ramkumar, and S. M. Manikandan, "Real-Time Signal Quality-Aware ECG Telemetry System for IoT-Based Health Care Monitoring," *IEEE Internet Things J.*, vol. 4, no. 3, pp. 815–823, Jun. 2017.

[38] R. K. Pathinarupothi, P. Durga, and E. S. Rangan, "IoT-based smart edge for global health: Remote monitoring with severity detection and alerts transmission," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2449–2462, Apr. 2019.

[39] D. Mocrii, Y. Chen, and P. Musilek, "IoT-based smart homes: A review of system architecture, software, communications, privacy and security," *Internet of Things (Netherlands)*, vol. 1–2. Elsevier B.V., pp. 81–98, 01-Sep-2018.

[40] C. C. Sobin, "A Survey on Architecture, Protocols and Challenges in IoT," *Wireless Personal Communications*, vol. 112, no. 3. Springer, pp. 1383–1429, 01-Jun-2020.

[41] S. Balaji, K. Nathani, and R. Santhakumar, "IoT Technology, Applications and Challenges: A Contemporary Survey," *Wireless Personal Communications*, vol. 108, no. 1. Springer New York LLC, pp. 363–388, 15-Sep-2019.

[42] W. Kassab and K. A. Darabkh, "A–Z survey of Internet of Things: Architectures, protocols, applications, recent advances, future directions and recommendations," *J. Netw. Comput. Appl.*, vol. 163, p. 102663, Aug. 2020.

[43] A. R. Al-Ali, I. A. Zualkernan, M. Rashid, R. Gupta, and M. Alikarar, "A smart home energy management system using IoT and big data analytics approach," *IEEE Trans. Consum. Electron.*, vol. 63, no. 4, pp. 426–434, Nov. 2017.

[44] M. Alaa, A. A. Zaidan, B. B. Zaidan, M. Talal, and M. L. M. Kiah, "A review of smart home applications based on Internet of Things," *Journal of Network and Computer Applications*, vol. 97. Academic Press, pp. 48–65, 01-Nov-2017.

[45] K. Saravanan, E. G. Julie, and Y. H. Robinson, "Smart cities & IoT: Evolution of applications, architectures & technologies, present scenarios & future dream," in *Intelligent Systems Reference Library*, vol. 154, Springer Science and Business Media Deutschland GmbH, 2019, pp. 135–151.

[46] V. Agarwal and S. Sharma, "IoT Based Smart Transport Management System," in *Communications in Computer and Information Science*, 2021, vol. 1394 CCIS, pp. 207–216.

[47] S. Sicari, A. Rizzardi, and A. Coen-Porisini, "Smart transport and logistics: A Node-RED implementation," *Internet Technol. Lett.*, vol. 2, no. 2, p. e88, Mar. 2019.

[48] A. Čolaković and M. Hadžialić, "Internet of Things (IoT): A review of enabling technologies, challenges, and open research issues," *Computer Networks*, vol. 144. Elsevier B.V., pp. 17–39, 24-Oct-2018.

[49] N. Neshenko, E. Bou-Harb, J. Crichigno, G. Kaddoum, and N. Ghani, “Demystifying IoT Security: An Exhaustive Survey on IoT Vulnerabilities and a First Empirical Look on Internet-Scale IoT Exploitations,” *IEEE Commun. Surv. Tutorials*, vol. 21, no. 3, pp. 2702–2733, 2019.

[50] P. Anand, Y. Singh, A. Selwal, M. Alazab, S. Tanwar, and N. Kumar, “IoT vulnerability assessment for sustainable computing: Threats, current solutions, and open challenges,” *IEEE Access*, vol. 8, pp. 168825–168853, 2020.

[51] X. Liang and Y. Kim, “A Survey on Security Attacks and Solutions in the IoT Network,” in *2021 IEEE 11th Annual Computing and Communication Workshop and Conference, CCWC 2021*, 2021, pp. 853–859.

[52] M. A. Ferrag, L. Maglaras, A. Argyriou, D. Kosmanos, and H. Janicke, “Security for 4G and 5G cellular networks: A survey of existing authentication and privacy-preserving schemes,” *Journal of Network and Computer Applications*, vol. 101. Academic Press, pp. 55–82, 01-Jan-2018.

[53] K. Zhao and L. Ge, “A survey on the internet of things security,” in *Proceedings - 9th International Conference on Computational Intelligence and Security, CIS 2013*, 2013, pp. 663–667.

[54] X. Yang *et al.*, “Towards a Low-Cost Remote Memory Attestation for the Smart Grid,” *Sensors*, vol. 15, no. 8, pp. 20799–20824, Aug. 2015.

[55] J. Lin, W. Yu, and X. Yang, “Towards Multistep Electricity Prices in Smart Grid Electricity Markets,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 1, pp. 286–302, Jan. 2016.

[56] G. Gomez, F. J. Lopez-Martinez, D. Morales-Jimenez, and M. R. McKay, “On the equivalence between interference and eavesdropping in wireless communications,” *IEEE Trans. Veh. Technol.*, vol. 64, no. 12, pp. 5935–5940, Dec. 2015.

[57] N. Zhao, F. R. Yu, M. Li, and V. C. M. Leung, “Anti-eavesdropping schemes for interference alignment (IA)-based wireless networks,” in *IEEE Transactions on Wireless Communications*, 2016, vol. 15, no. 8, pp. 5719–5732.

[58] S. Hoffmann and G. Bumiller, “Identification and Simulation of a Denial-of-Sleep Attack on Open Metering System,” in *Proceedings of 2019 IEEE PES Innovative Smart Grid Technologies Europe, ISGT-Europe 2019*, 2019.

[59] J. Newsome, E. Shi, D. Song, and A. Perrig, “The Sybil attack in sensor networks: analysis & defenses - IEEE Conference Publication,” in *Third International Symposium on Information Processing in Sensor Networks, 2004. IPSN 2004*, 2004.

[60] K. Zhang, X. Liang, R. Lu, and X. Shen, “Sybil attacks and their defenses in the internet of things,” *IEEE Internet Things J.*, vol. 1, no. 5, pp. 372–383, Oct. 2014.

[61] K. N. Qureshi, S. S. Rana, A. Ahmed, and G. Jeon, “A novel and secure attacks detection framework for smart cities industrial internet of things,” *Sustain. Cities Soc.*, vol. 61, p. 102343, Oct. 2020.

[62] A. Mayzaud, R. Badonnel, and I. Chrisment, “A Taxonomy of Attacks in RPL-based Internet of Things,” *Int. J. Netw. Secur.*, vol. 18, no. 3, pp. 459–473, 2016.

[63] I. Butun, P. Osterberg, and H. Song, “Security of the Internet of Things: Vulnerabilities, Attacks, and Countermeasures,” *IEEE Commun. Surv. Tutorials*, vol. 22, no. 1, pp. 616–644, Jan. 2020.

[64] M. El-hajj, A. Fadlallah, M. Chamoun, and A. Serhrouchni, “A Survey of Internet of Things (IoT)

Authentication Schemes," *Sensors*, vol. 19, no. 5, p. 1141, Mar. 2019.

[65] M. Frustaci, P. Pace, G. Aloi, and G. Fortino, "Evaluating critical security issues of the IoT world: Present and future challenges," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2483–2495, 2018.

[66] ITU-T, "Y.2066 : Common requirements of the Internet of things," 2014.

[67] J. Asharf, N. Moustafa, H. Khurshid, E. Debie, W. Haider, and A. Wahab, "A Review of Intrusion Detection Systems Using Machine and Deep Learning in Internet of Things: Challenges, Solutions and Future Directions," *Electronics*, vol. 9, no. 7, p. 1177, Jul. 2020.

[68] G. D'Agostini, "A multidimensional unfolding method based on Bayes' theorem," *Nucl. Inst. Methods Phys. Res. A*, vol. 362, no. 2–3, pp. 487–498, Aug. 1995.

[69] I. Witten, E. Frank, and M. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*. Elsevier, 2011.

[70] S. B. Kotsiantis, "Decision trees: A recent overview," *Artificial Intelligence Review*, vol. 39, no. 4. Springer, pp. 261–283, 29-Apr-2013.

[71] G. Zachos, I. Essop, G. Mantas, K. Porfyrakis, J. C. Ribeiro, and J. Rodriguez, "An Anomaly-Based Intrusion Detection System for Internet of Medical Things Networks," *Electronics*, vol. 10, no. 21, p. 2562, Oct. 2021.

[72] A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems.* O'Reilly Media, Inc., 2019.

[73] L. Guangjun, S. Nazir, H. U. Khan, and A. U. Haq, "Spam Detection Approach for Secure Mobile Message Communication Using Machine Learning Algorithms," *Secur. Commun. Networks*, vol. 2020, 2020.

[74] Q. Liu, P. Li, W. Zhao, W. Cai, S. Yu, and V. C. M. Leung, "A survey on security threats and defensive techniques of machine learning: A data driven view," *IEEE Access*, vol. 6. Institute of Electrical and Electronics Engineers Inc., pp. 12103–12117, 12-Feb-2018.

[75] L. Huraj, T. Horak, P. Strelec, and P. Tanuska, "Mitigation against DDoS Attacks on an IoT-Based Production Line Using Machine Learning," *Appl. Sci.*, vol. 11, no. 4, p. 1847, Feb. 2021.

[76] J. S. Pan, F. Fan, S. C. Chu, H. Q. Zhao, and G. Y. Liu, "A Lightweight Intelligent Intrusion Detection Model for Wireless Sensor Networks," *Secur. Commun. Networks*, vol. 2021, 2021.

[77] F. Alam, R. Mehmood, I. Katib, and A. Albeshri, "Analysis of Eight Data Mining Algorithms for Smarter Internet of Things (IoT)," in *Procedia Computer Science*, 2016, vol. 58, pp. 437–442.

[78] P. Illy, G. Kaddoum, C. M. Moreira, K. Kaur, and S. Garg, "Securing Fog-to-Things Environment Using Intrusion Detection System Based On Ensemble Learning," in *IEEE Wireless Communications and Networking Conference, WCNC*, 2019, vol. 2019-April, pp. 1–7.

[79] M. Woźniak, M. Graña, and E. Corchado, "A survey of multiple classifier systems as hybrid systems," *Inf. Fusion*, vol. 16, no. 1, pp. 3–17, Mar. 2014.

[80] P. Domingos, "A few useful things to know about machine learning," *Commun. ACM*, vol. 55, no. 10, pp. 78–87, 2012.

[81] L. E. A. Santana, L. Silva, A. M. P. Canuto, F. Pintro, and K. O. Vale, "A comparative analysis of genetic algorithm and ant colony optimization to select attributes for an heterogeneous ensemble of classifiers," in *2010 IEEE World Congress on Computational Intelligence, WCCI*

2010 - 2010 IEEE Congress on Evolutionary Computation, CEC 2010, 2010.

[82] B. A. Tama and S. Lim, "Ensemble learning for intrusion detection systems: A systematic mapping study and cross-benchmark evaluation," *Computer Science Review*, vol. 39. Elsevier Ireland Ltd, p. 100357, 01-Feb-2021.

[83] Y. Zhou, G. Cheng, S. Jiang, and M. Dai, "Building an efficient intrusion detection system based on feature selection and ensemble classifier," *Comput. Networks*, vol. 174, p. 107247, Jun. 2020.

[84] M. S. Abirami, U. Yash, and S. Singh, "Building an Ensemble Learning Based Algorithm for Improving Intrusion Detection System," in *Advances in Intelligent Systems and Computing*, 2020, vol. 1056, pp. 635–649.

[85] H. H. W. J. Bosman, G. Iacca, A. Tejada, H. J. Wörtche, and A. Liotta, "Ensembles of incremental learners to detect anomalies in ad hoc sensor networks," *Ad Hoc Networks*, vol. 35, pp. 14–36, Dec. 2015.

[86] Z. Jin, J. Shang, Q. Zhu, C. Ling, W. Xie, and B. Qiang, "RFRSF: Employee Turnover Prediction Based on Random Forests and Survival Analysis," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2020, vol. 12343 LNCS, pp. 503–515.

[87] R. Doshi, N. Apthorpe, and N. Feamster, "Machine learning DDoS detection for consumer internet of things devices," in *Proceedings - 2018 IEEE Symposium on Security and Privacy Workshops, SPW 2018*, 2018, pp. 29–35.

[88] R. E. Schapire, "Explaining adaboost," in *Empirical Inference: Festschrift in Honor of Vladimir N. Vapnik*, Springer Berlin Heidelberg, 2013, pp. 37–52.

[89] A. Shahraki, M. Abbasi, and Ø. Haugen, "Boosting algorithms for network intrusion detection: A comparative evaluation of Real AdaBoost, Gentle AdaBoost and Modest AdaBoost," *Eng. Appl. Artif. Intell.*, vol. 94, p. 103770, Sep. 2020.

[90] A. Yulianto, P. Sukarno, and N. A. Suwastika, "Improving AdaBoost-based Intrusion Detection System (IDS) Performance on CIC IDS 2017 Dataset," in *Journal of Physics: Conference Series*, 2019, vol. 1192, no. 1, p. 012018.

[91] M. Mazini, B. Shirazi, and I. Mahdavi, "Anomaly network-based intrusion detection system using a reliable hybrid artificial bee colony and AdaBoost algorithms," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 31, no. 4, pp. 541–553, Oct. 2019.

[92] N. Moustafa, J. Hu, and J. Slay, "A holistic review of Network Anomaly Detection Systems: A comprehensive survey," *J. Netw. Comput. Appl.*, vol. 128, pp. 33–55, Feb. 2019.

[93] A. Verma and V. Ranga, "Machine Learning Based Intrusion Detection Systems for IoT Applications," *Wirel. Pers. Commun.*, vol. 111, no. 4, pp. 2287–2310, Apr. 2020.

[94] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "Network anomaly detection: Methods, systems and tools," *IEEE Commun. Surv. Tutorials*, vol. 16, no. 1, pp. 303–336, Mar. 2014.

[95] G. Zachos, I. Essop, G. Mantas, K. Porfyrakis, J. C. Ribeiro, and J. Rodriguez, "Generating IoT Edge Network Datasets based on the TON_IoT Telemetry Dataset," in *2021 IEEE 26th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, 2021, pp. 1–6.

[96] "Node-RED." [Online]. Available: https://nodered.org/. [Accessed: 10-Dec-2021].

[97] "ToN_IoT datasets | IEEE DataPort." [Online]. Available: https://ieee-dataport.org/documents/toniot-datasets. [Accessed: 10-Dec-2021].

[98] "What is VMware NSX? | Network Security Virtualization Platform." [Online]. Available: https://www.vmware.com/products/nsx.html. [Accessed: 10-Dec-2021].

[99] I. Stojmenovic and S. Wen, "The Fog computing paradigm: Scenarios and security issues," in *2014 Federated Conference on Computer Science and Information Systems, FedCSIS 2014*, 2014, pp. 1–8.

[100] "Kali Linux | Penetration Testing and Ethical Hacking Linux Distribution." [Online]. Available: https://www.kali.org/. [Accessed: 10-Dec-2021].

[101] "HiveMQ - Enterprise ready MQTT to move your IoT data." [Online]. Available: https://www.hivemq.com/. [Accessed: 10-Dec-2021].

[102] "Home of Acunetix Art." [Online]. Available: http://testphp.vulnweb.com/. [Accessed: 10-Dec-2021].

[103] "IoT Hub | Microsoft Azure." [Online]. Available: https://azure.microsoft.com/en-au/services/iot-hub/. [Accessed: 10-Dec-2021].

[104] "Serverless Computing - AWS Lambda - Amazon Web Services." [Online]. Available: https://aws.amazon.com/lambda/. [Accessed: 10-Dec-2021].

[105] Moteiv Corporation, "tmote sky - Ultra low power IEEE 802.15.4 compliant wireless sensor module," 2006. [Online]. Available: http://www.crew-project.eu/sites/default/files/tmote-sky-datasheet.pdf [Accessed: 10-Dec-2021].

[106] "Wireshark · Go Deep." [Online]. Available: https://www.wireshark.org/. [Accessed: 28-Nov-2020].

[107] A. Dunkels, J. Eriksson, N. Finne, and N. Tsiftes, "Powertrace: Network-level Power Profiling for Low-power Wireless Networks," Kista, Sweden, 2011.

[108] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Comput. Electr. Eng.*, vol. 40, no. 1, pp. 16–28, Jan. 2014.

[109] M. Amirinasab Nasab, S. Shamshirband, A. Chronopoulos, A. Mosavi, and N. Nabipour, "Energy-Efficient Method for Wireless Sensor Networks Low-Power Radio Operation in Internet of Things," *Electronics*, vol. 9, no. 2, pp. 2–19, Feb. 2020.

[110] A. Bandekar and A. Y. Javaid, "Cyber-attack Mitigation and Impact Analysis for Low-power IoT Devices," in *2017 IEEE 7th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems, CYBER 2017*, 2018, pp. 1631–1636.

Full List of Publications

International Peer Reviewed Journal papers (4)

1. Zachos, Georgios, **Essop, Ismael**, Mantas, Georgios, Porfyrakis, Kyriakos, Ribeiro, Jose, Rodriguez, Jonathan (2021), An anomaly-based intrusion detection system for internet of medical things networks. Electronics, 10: 2562 (21) 2079-9292 (Online) (doi: https://doi.org/10.3390/electronics10212562).
2. **Essop, Ismael**, Ribeiro, José C., Papaioannou, Maria, Zachos, Georgios, Mantas, Georgios, Rodriguez, Jonathan (2021), Generating datasets for anomaly-based intrusion detection systems in IoT and industrial IoT networks. Sensors, 21: 1528 (4) 1424-8220 (Online) (doi: https://doi.org/10.3390/s21041528).
3. Papaioannou, Maria, Karageorgou, Marine, Mantas, Georgios, Sucasas, Victor, **Essop, Ismael**, Rodriguez, Jonathan, Lymberopoulos, Dimitrios (2020), A Survey on security threats and countermeasures in Internet of Medical Things (IoMT). Transactions on Emerging Telecommunications Technologies: e4049 ISSN: 2161-3915 (Print), (doi: https://doi.org/10.1002/ett.4049).
4. Essop, Ismael A. , Evans, Richard D., Wan, Shan, Giddaluru, Muni P. , Gao, James X. , Baudry, David , Mahdikhah, Sara , Messaadia, Mourad (2016), Investigation into current industrial practices relating to product lifecycle management in a multi-national manufacturing company. Computer-Aided Design and Applications, 13 (5) pp. 647-661 1686-4360 (Online) (doi: http://dx.doi.org/10.1080/16864360.2016.1150711).

International Peer Reviewed Book Chapters (1)

1. Karageorgou, Marina, Mantas, Georgios, Essop, Ismael, Rodriguez, J , Lymberopoulos, D (2020), Cybersecurity Attacks on Medical IoT Devices for Smart City Healthcare Services. In: Fadi Al-Turjman, Muhammad Imran (eds.), IOT Technologies in Smart-Cities: From Sensors to Big Data, Security and Trus. Institution of Engineering & Technology, pp. 171-187. ISBN: 9781785618697 (doi: https://doi.org/10.1049/PBCE128E).

International Peer Reviewed Conference papers (3)

1. Zachos, Georgios, **Essop, Ismael**, Mantas, Georgios, Kyriakos, Porfyrakis , Jose, Ribeiro , Jonathan, Rodriguez (2021), Generating IoT Edge Network Datasets based on the TON_IoT telemetry dataset. (1st) . pp. 1-6 . ISBN: 9781665417792ISSN: 2378-4865 (Print), 2378-4873 (Online) (doi: https://doi.org/10.1109/CAMAD52502.2021.9617799).
2. Essop, Ismael A. , Evans, Richard D., Giddalaru, Muni P., Gao, James X. , Wan, Shan , Baudry, David , Mahdikhah, Sara , Messaadia, Mourad (2015), Investigation into current industrial practices relating to product lifecycle management in a multi-national manufacturing company. . pp. 437-442 (doi: http://dx.doi.org/10.14733/cadconfP.2015.437-442).

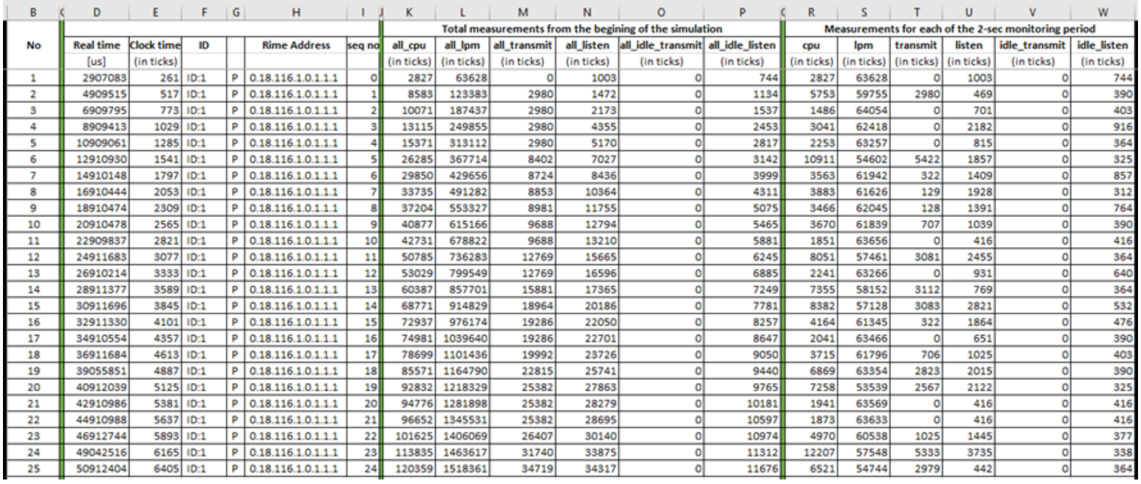
3. Essop, Ismael and Gao, Xiaoyu (2015) Opportunities and challenges in using Big Data for product maintenance in the power generation industry. In: Proceedings of the International Conference on Manufacturing Research. ICMR 2015, 13th International Conference on Manufacturing Research. ISBN 1857901878.

Appendix 1

A1.1 Datasets for Benign Motes

A1.1.1- Benign “mote1.csv”

The generated benign “mote1.csv” file, related to the UDP-server mote1, consists of 1,799 records and its first 25 records (i.e., 1–25) are depicted in Figure A1.

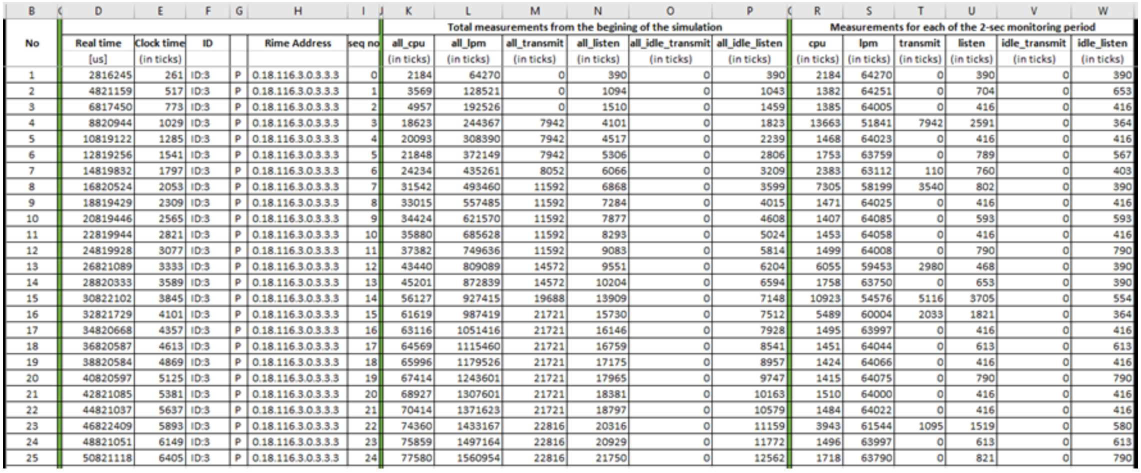


B	D	E	F	G	H	I	K	L	M	N	O	P	R	S	T	U	V	W
							Total measurements from the begining of the simulation						Measurements for each of the 2-sec monitoring period					
No	Real time	Clock time	ID		Rime Address	seq no	all_cpu	all_lpm	all_transmit	all_listen	all_idle_transmit	all_idle_listen	cpu	lpm	transmit	listen	idle_transmit	idle_listen
	[us]	(in ticks)					(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)
1	2907083	261	ID:1	P	0.18.116.1.0.1.1.1	0	2827	63628	0	1003	0	744	2827	63628	0	1003	0	744
2	4909515	517	ID:1	P	0.18.116.1.0.1.1.1	1	8583	123383	2980	1472	0	1134	5753	59755	2980	469	0	390
3	6909795	773	ID:1	P	0.18.116.1.0.1.1.1	2	10071	187437	2980	2173	0	1537	1486	64054	0	701	0	403
4	8909413	1029	ID:1	P	0.18.116.1.0.1.1.1	3	13115	249855	2980	4355	0	2453	3041	62418	0	2182	0	916
5	10909061	1285	ID:1	P	0.18.116.1.0.1.1.1	4	15371	313112	2980	5170	0	2817	2253	63257	0	815	0	364
6	12910930	1541	ID:1	P	0.18.116.1.0.1.1.1	5	26285	367714	8402	7027	0	3142	10911	54602	5422	1857	0	325
7	14910148	1797	ID:1	P	0.18.116.1.0.1.1.1	6	29850	429656	8724	8436	0	3999	3563	61942	322	1409	0	857
8	16910444	2053	ID:1	P	0.18.116.1.0.1.1.1	7	33735	491282	8853	10364	0	4311	3883	61626	129	1928	0	312
9	18910474	2309	ID:1	P	0.18.116.1.0.1.1.1	8	37204	553327	8981	11755	0	5075	3466	62045	128	1391	0	764
10	20910478	2565	ID:1	P	0.18.116.1.0.1.1.1	9	40877	615166	9688	12794	0	5465	3670	61839	707	1039	0	390
11	22909837	2821	ID:1	P	0.18.116.1.0.1.1.1	10	42731	678822	9688	13210	0	5881	1851	63656	0	416	0	416
12	24911683	3077	ID:1	P	0.18.116.1.0.1.1.1	11	50785	736283	12769	15665	0	6245	8051	57461	3081	2455	0	364
13	26910214	3333	ID:1	P	0.18.116.1.0.1.1.1	12	53029	799549	12769	16596	0	6885	2241	63266	0	931	0	640
14	28911377	3589	ID:1	P	0.18.116.1.0.1.1.1	13	60387	857701	15881	17365	0	7249	7355	58152	3112	769	0	364
15	30911696	3845	ID:1	P	0.18.116.1.0.1.1.1	14	68771	914829	18964	20186	0	7781	8382	57128	3083	2821	0	532
16	32911330	4101	ID:1	P	0.18.116.1.0.1.1.1	15	72937	976174	19286	22050	0	8257	4164	61345	322	1864	0	476
17	34910554	4357	ID:1	P	0.18.116.1.0.1.1.1	16	74981	1039640	19286	22701	0	8647	2041	63466	0	651	0	390
18	36911684	4613	ID:1	P	0.18.116.1.0.1.1.1	17	78699	1101436	19992	23726	0	9050	3715	61796	706	1025	0	403
19	39055851	4887	ID:1	P	0.18.116.1.0.1.1.1	18	85571	1164790	22815	25741	0	9440	6869	63354	2823	2015	0	390
20	40912039	5125	ID:1	P	0.18.116.1.0.1.1.1	19	92832	1218329	25382	27863	0	9765	7258	53539	2567	2122	0	325
21	42910986	5381	ID:1	P	0.18.116.1.0.1.1.1	20	94776	1281898	25382	28279	0	10181	1941	63569	0	416	0	416
22	44910988	5637	ID:1	P	0.18.116.1.0.1.1.1	21	96652	1345531	25382	28695	0	10597	1873	63633	0	416	0	416
23	46912744	5893	ID:1	P	0.18.116.1.0.1.1.1	22	101625	1406069	26407	30140	0	10974	4970	60538	1025	1445	0	377
24	49042516	6165	ID:1	P	0.18.116.1.0.1.1.1	23	113835	1463617	31740	33875	0	11312	12207	57548	5333	3735	0	338
25	50912404	6405	ID:1	P	0.18.116.1.0.1.1.1	24	120359	1518361	34719	34317	0	11676	6521	54744	2979	442	0	364

Figure A1 Benign “mote1.csv”—1 to 25 records.

A1.1.2 Benign “mote3.csv”

The generated benign “mote3.csv” file, related to the UDP-client mote3, consists of 1,799 records and its first 25 records (i.e., 1–25) are depicted in Figure .



B	D	E	F	G	H	I	K	L	M	N	O	P	R	S	T	U	V	W
							Total measurements from the begining of the simulation						Measurements for each of the 2-sec monitoring period					
No	Real time	Clock time	ID		Rime Address	seq no	all_cpu	all_lpm	all_transmit	all_listen	all_idle_transmit	all_idle_listen	cpu	lpm	transmit	listen	idle_transmit	idle_listen
	[us]	(in ticks)					(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)
1	2816245	261	ID:3	P	0.18.116.3.0.3.3.3	0	2184	64270	0	390	0	390	2184	64270	0	390	0	390
2	4821159	517	ID:3	P	0.18.116.3.0.3.3.3	1	3569	128521	0	1094	0	1043	1382	64251	0	704	0	653
3	6817450	773	ID:3	P	0.18.116.3.0.3.3.3	2	4957	192526	0	1510	0	1459	1385	64005	0	416	0	416
4	8820944	1029	ID:3	P	0.18.116.3.0.3.3.3	3	18623	244367	7942	4101	0	1823	13663	51841	7942	2591	0	364
5	10819122	1285	ID:3	P	0.18.116.3.0.3.3.3	4	20093	308390	7942	4517	0	2239	1468	64023	0	416	0	416
6	12819256	1541	ID:3	P	0.18.116.3.0.3.3.3	5	21848	372149	7942	5306	0	2806	1753	63759	0	789	0	567
7	14819832	1797	ID:3	P	0.18.116.3.0.3.3.3	6	24234	435261	8052	6066	0	3209	2383	63112	110	760	0	403
8	16820524	2053	ID:3	P	0.18.116.3.0.3.3.3	7	31542	493460	11592	6868	0	3599	7305	58199	3540	802	0	390
9	18819429	2309	ID:3	P	0.18.116.3.0.3.3.3	8	33015	557485	11592	7284	0	4015	1471	64025	0	416	0	416
10	20819446	2565	ID:3	P	0.18.116.3.0.3.3.3	9	34424	621570	11592	7877	0	4608	1407	64085	0	593	0	593
11	22819944	2821	ID:3	P	0.18.116.3.0.3.3.3	10	35880	685628	11592	8293	0	5024	1453	64058	0	416	0	416
12	24819928	3077	ID:3	P	0.18.116.3.0.3.3.3	11	37382	749636	11592	9083	0	5814	1499	64008	0	790	0	790
13	26821089	3333	ID:3	P	0.18.116.3.0.3.3.3	12	43440	809089	14572	9551	0	6204	6055	59453	2980	468	0	390
14	28820333	3589	ID:3	P	0.18.116.3.0.3.3.3	13	45201	872839	14572	10204	0	6594	1758	63750	0	653	0	390
15	30822102	3845	ID:3	P	0.18.116.3.0.3.3.3	14	56127	927415	19688	13909	0	7148	10923	54576	5116	3705	0	554
16	32821729	4101	ID:3	P	0.18.116.3.0.3.3.3	15	61619	987419	21721	15730	0	7512	5489	60004	2033	1821	0	364
17	34820668	4357	ID:3	P	0.18.116.3.0.3.3.3	16	63116	1051416	21721	16146	0	7928	1495	63997	0	416	0	416
18	36820587	4613	ID:3	P	0.18.116.3.0.3.3.3	17	64569	1115460	21721	16759	0	8541	1451	64044	0	613	0	613
19	38820584	4869	ID:3	P	0.18.116.3.0.3.3.3	18	65996	1179526	21721	17175	0	8957	1424	64066	0	416	0	416
20	40820597	5125	ID:3	P	0.18.116.3.0.3.3.3	19	67414	1243601	21721	17965	0	9747	1415	64075	0	790	0	790
21	42821085	5381	ID:3	P	0.18.116.3.0.3.3.3	20	68927	1307601	21721	18381	0	10163	1510	64000	0	416	0	416
22	44821037	5637	ID:3	P	0.18.116.3.0.3.3.3	21	70414	1371623	21721	18797	0	10579	1484	64022	0	416	0	416
23	46822409	5893	ID:3	P	0.18.116.3.0.3.3.3	22	74360	1433167	22816	20316	0	11159	3943	61544	1095	1519	0	580
24	48821051	6149	ID:3	P	0.18.116.3.0.3.3.3	23	75859	1497164	22816	20929	0	11772	1496	63997	0	613	0	613
25	50821118	6405	ID:3	P	0.18.116.3.0.3.3.3	24	77580	1560954	22816	21750	0	12562	1718	63790	0	821	0	790

Figure A2 Benign “mote3.csv”—1 to 25 records.

A1.2 - Datasets for UDP Flood Attacks

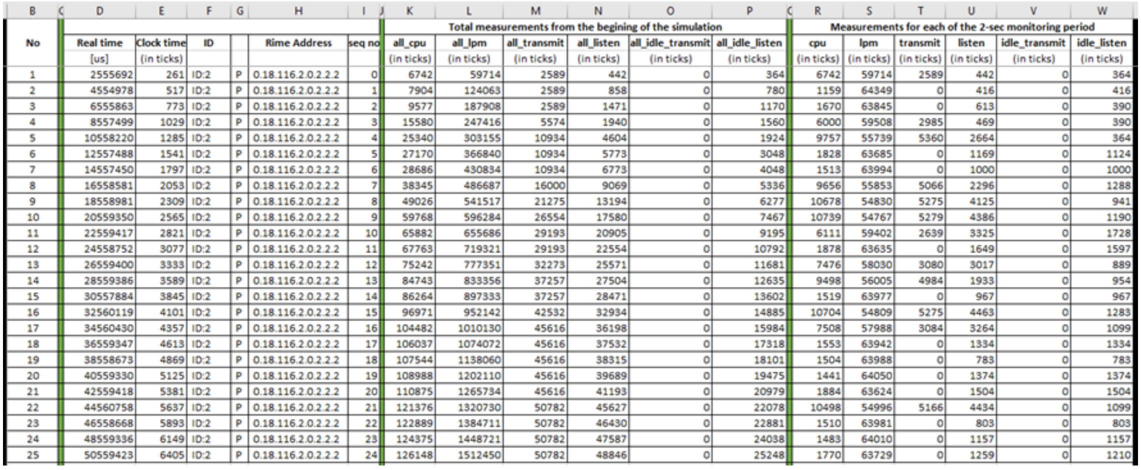
A1.2.1 - “udp-flood-mote1.csv”

The generated “udp-flood-mote1.csv” file, related to the benign UDP-server mote1, consists of 1,799 records and its first 25 records (i.e., 1–25) are depicted below in **Error! Reference source not found.**.

Figure A3 Malicious “udp-flood-mote1.csv” — 1 to 25 records.

A1.2.2 - “udp-flood-mote2.csv”

The generated “udp-flood-mote2.csv” file, related to the benign UDP-client mote2, consists of 1,799 records and its first 25 records (i.e., 1–25) are depicted below in Figure .

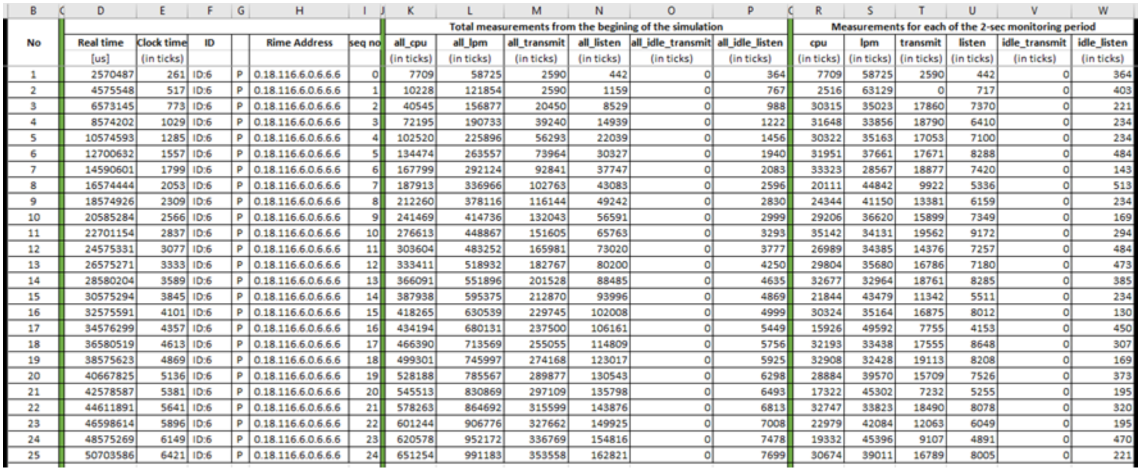


B	D	E	F	G	H	I	K	L	M	N	O	P	R	S	T	U	V	W
No							Total measurements from the begining of the simulation						Measurements for each of the 2-sec monitoring period					
	Real time	Clock time	ID		Rime Address	seq no	all_cpu	all_lpm	all_transmit	all_listen	all_idle_transmit	all_idle_listen	cpu	lpm	transmit	listen	idle_transmit	idle_listen
	[us]	(in ticks)					(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)
1	2555692	261	ID:2	P	0.18.116.2.0.2.2.2	0	6742	59714	2589	442	0	364	6742	59714	2589	442	0	364
2	4554978	517	ID:2	P	0.18.116.2.0.2.2.2	1	7904	124063	2589	858	0	780	1159	64349	0	416	0	416
3	6555863	773	ID:2	P	0.18.116.2.0.2.2.2	2	9577	187908	2589	1471	0	1170	1670	63845	0	613	0	390
4	8557499	1029	ID:2	P	0.18.116.2.0.2.2.2	3	15580	247416	5574	1940	0	1560	6000	59508	2985	469	0	390
5	10558220	1285	ID:2	P	0.18.116.2.0.2.2.2	4	25340	303155	10934	4604	0	1924	9757	55739	5360	2664	0	364
6	12557488	1541	ID:2	P	0.18.116.2.0.2.2.2	5	27170	366840	10934	5773	0	3048	1828	63685	0	1169	0	1124
7	14557450	1797	ID:2	P	0.18.116.2.0.2.2.2	6	28686	430834	10934	6773	0	4048	1513	63994	0	1000	0	1000
8	16558581	2053	ID:2	P	0.18.116.2.0.2.2.2	7	38345	486687	16000	9069	0	5336	9656	55853	5066	2296	0	1288
9	18558981	2309	ID:2	P	0.18.116.2.0.2.2.2	8	49026	541517	21275	13194	0	6277	10678	54830	5275	4125	0	941
10	20559350	2565	ID:2	P	0.18.116.2.0.2.2.2	9	59768	596284	26554	17580	0	7467	10739	54767	5279	4386	0	1190
11	22559417	2821	ID:2	P	0.18.116.2.0.2.2.2	10	65882	655686	29193	20905	0	9195	6111	59402	2639	3325	0	1728
12	24558752	3077	ID:2	P	0.18.116.2.0.2.2.2	11	67763	719321	29193	22554	0	10792	1878	63635	0	1649	0	1597
13	26559400	3333	ID:2	P	0.18.116.2.0.2.2.2	12	75242	777351	32273	25571	0	11681	7476	58030	3080	3017	0	889
14	28559386	3589	ID:2	P	0.18.116.2.0.2.2.2	13	84743	833356	37257	27504	0	12635	9498	56005	4984	1933	0	954
15	30557884	3845	ID:2	P	0.18.116.2.0.2.2.2	14	86264	897333	37257	28471	0	13602	1519	63977	0	967	0	967
16	32560119	4101	ID:2	P	0.18.116.2.0.2.2.2	15	96971	952142	42532	32934	0	14885	10704	54809	5275	4463	0	1283
17	34560430	4357	ID:2	P	0.18.116.2.0.2.2.2	16	104482	1010130	45616	36198	0	15984	7508	57988	3084	3264	0	1099
18	36559347	4613	ID:2	P	0.18.116.2.0.2.2.2	17	106037	1074072	45616	37532	0	17318	1553	63942	0	1334	0	1334
19	38558673	4869	ID:2	P	0.18.116.2.0.2.2.2	18	107544	1138060	45616	38315	0	18101	1504	63988	0	783	0	783
20	40559330	5125	ID:2	P	0.18.116.2.0.2.2.2	19	108988	1202110	45616	39689	0	19475	1441	64050	0	1374	0	1374
21	42559418	5381	ID:2	P	0.18.116.2.0.2.2.2	20	110875	1265734	45616	41193	0	20979	1884	63624	0	1504	0	1504
22	44560758	5637	ID:2	P	0.18.116.2.0.2.2.2	21	121376	1320730	50782	45627	0	22078	10498	54996	5166	4434	0	1099
23	46558668	5893	ID:2	P	0.18.116.2.0.2.2.2	22	122889	1384711	50782	46430	0	22881	1510	63981	0	803	0	803
24	48559336	6149	ID:2	P	0.18.116.2.0.2.2.2	23	124375	1448721	50782	47587	0	24038	1483	64010	0	1157	0	1157
25	50559423	6405	ID:2	P	0.18.116.2.0.2.2.2	24	126148	1512450	50782	48846	0	25248	1770	63729	0	1259	0	1210

Figure A4 Malicious “udp-flood-mote2.csv” — 1 to 25 records.

A1.2.3 - “udp-flood-mote6.csv”

The generated “udp-flood-mote6.csv” file, related to the malicious UDP-client mote6, consists of 1,799 records and its first 25 records (i.e., 1–25) are depicted below in Figure .



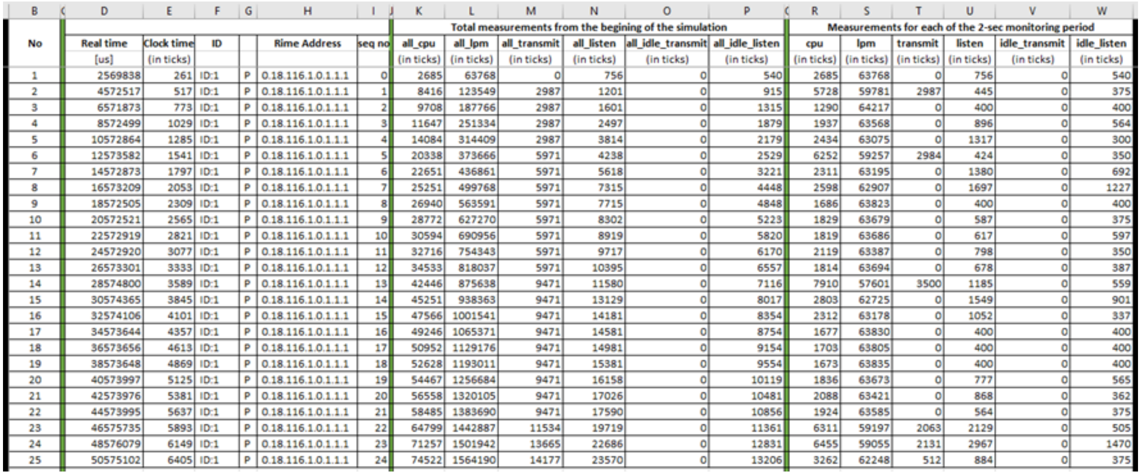
B	D	E	F	G	H	I	K	L	M	N	O	P	R	S	T	U	V	W
No							Total measurements from the begining of the simulation						Measurements for each of the 2-sec monitoring period					
	Real time	Clock time	ID		Rime Address	seq no	all_cpu	all_lpm	all_transmit	all_listen	all_idle_transmit	all_idle_listen	cpu	lpm	transmit	listen	idle_transmit	idle_listen
	[us]	(in ticks)					(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)
1	2570487	261	ID:6	P	0.18.116.6.0.6.6.6	0	7709	58725	2590	442	0	364	7709	58725	2590	442	0	364
2	4575548	517	ID:6	P	0.18.116.6.0.6.6.6	1	10228	121854	2590	1159	0	767	2516	63129	0	717	0	403
3	6573145	773	ID:6	P	0.18.116.6.0.6.6.6	2	40545	156877	20450	8529	0	988	30315	35023	17860	7370	0	221
4	8574202	1029	ID:6	P	0.18.116.6.0.6.6.6	3	72195	190733	39240	14939	0	1222	31648	33856	18790	6410	0	234
5	10574593	1285	ID:6	P	0.18.116.6.0.6.6.6	4	102520	225896	56293	22039	0	1456	30322	35163	17053	7100	0	234
6	12700632	1557	ID:6	P	0.18.116.6.0.6.6.6	5	134474	263557	73964	30327	0	1940	31951	37661	17671	8288	0	484
7	14590601	1799	ID:6	P	0.18.116.6.0.6.6.6	6	167799	292124	92841	37747	0	2083	33323	28567	18877	7420	0	143
8	16574444	2053	ID:6	P	0.18.116.6.0.6.6.6	7	187913	336966	102763	43083	0	2596	20111	44842	9922	5336	0	513
9	18574926	2309	ID:6	P	0.18.116.6.0.6.6.6	8	212260	378116	116144	49242	0	2830	24344	41150	13381	6159	0	234
10	20585284	2566	ID:6	P	0.18.116.6.0.6.6.6	9	241469	414736	132043	56591	0	2999	29206	36620	15899	7349	0	169
11	22701154	2837	ID:6	P	0.18.116.6.0.6.6.6	10	276613	448867	151605	65763	0	3293	35142	34131	19562	9172	0	294
12	24575331	3077	ID:6	P	0.18.116.6.0.6.6.6	11	303604	483252	165981	73020	0	3777	26989	34385	14376	7257	0	484
13	26575271	3333	ID:6	P	0.18.116.6.0.6.6.6	12	333411	518932	182767	80200	0	4250	29804	35680	16786	7180	0	473
14	28580204	3589	ID:6	P	0.18.116.6.0.6.6.6	13	366091	551896	201528	88485	0	4635	32677	32964	18761	8285	0	385
15	30575294	3845	ID:6	P	0.18.116.6.0.6.6.6	14	387938	595375	212870	93996	0	4869	21844	43479	11342	5511	0	234
16	32575591	4101	ID:6	P	0.18.116.6.0.6.6.6	15	418265	630539	229745	102008	0	4999	30324	35164	16875	8012	0	130
17	34576299	4357	ID:6	P	0.18.116.6.0.6.6.6	16	434194	680131	237500	106161	0	5449	15926	49592	7755	4153	0	450
18	36580519	4613	ID:6	P	0.18.116.6.0.6.6.6	17	466390	713569	255055	114809	0	5756	32193	33438	17555	8648	0	307
19	38575623	4869	ID:6	P	0.18.116.6.0.6.6.6	18	499301	745997	274168	123017	0	5925	32908	32428	19113	8208	0	169
20	40667825	5136	ID:6	P	0.18.116.6.0.6.6.6	19	528188	785567	289877	130543	0	6298	28884	39570	15709	7526	0	373
21	42578587	5381	ID:6	P	0.18.116.6.0.6.6.6	20	545513	830869	297109	135798	0	6493	17322	45302	7232	5255	0	195
22	44611891	5641	ID:6	P	0.18.116.6.0.6.6.6	21	578263	864692	315599	143876	0	6813	32747	33823	18490	8078	0	320
23	46598614	5896	ID:6	P	0.18.116.6.0.6.6.6	22	601244	906776	327662	149925	0	7008	22979	42084	12063	6049	0	195
24	48575269	6149	ID:6	P	0.18.116.6.0.6.6.6	23	620578	952172	336769	154816	0	7478	19332	45396	9107	4891	0	470
25	50703586	6421	ID:6	P	0.18.116.6.0.6.6.6	24	651254	991183	353558	162821	0	7699	30674	39011	16789	8005	0	221

Figure A5 Malicious “udp-flood-mote6.csv” — 1 to 25 records.

A1.3 - Datasets for Blackhole Attacks

A1.3.1 - “blackhole-mote1.csv”

The generated malicious “blackhole-mote1.csv” file, related to the benign UDP-server mote1, consists of 1,799 records and its first 25 records (i.e., 1–25) are depicted below in Figure .

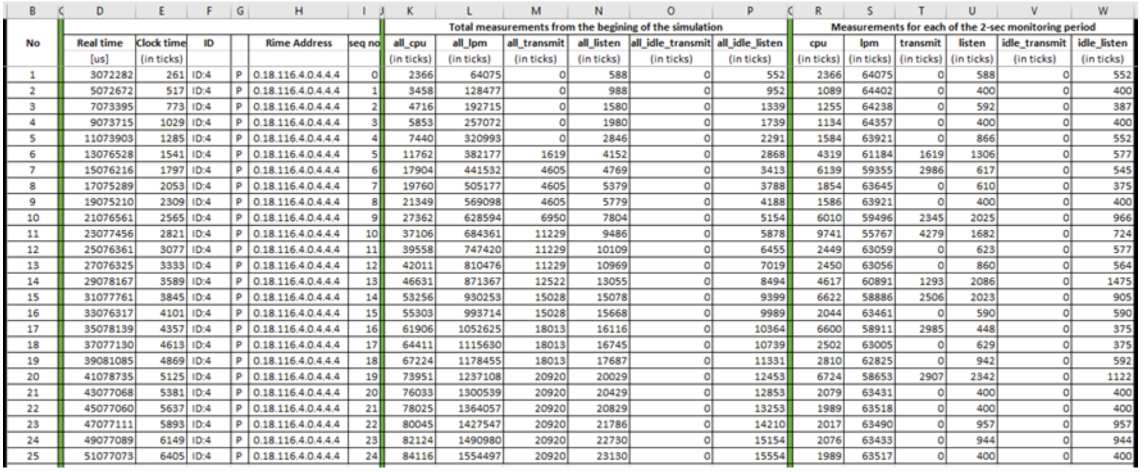


B	D	E	F	G	H	I	K	L	M	N	O	P	R	S	T	U	V	W
No							Total measurements from the begining of the simulation						Measurements for each of the 2-sec monitoring period					
	Real time	Clock time	ID		Rime Address	seq no	all_cpu	all_lpm	all_transmit	all_listen	all_idle_transmit	all_idle_listen	cpu	lpm	transmit	listen	idle_transmit	idle_listen
	[us]	(in ticks)					(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)
1	2569838	261	ID:1	P	0.18.116.1.0.1.1.1	0	2685	63768	0	756	0	540	2685	63768	0	756	0	540
2	4572517	517	ID:1	P	0.18.116.1.0.1.1.1	1	8416	123549	2987	1201	0	915	5728	59781	2987	445	0	375
3	6571873	773	ID:1	P	0.18.116.1.0.1.1.1	2	9708	187766	2987	1601	0	1315	1290	64217	0	400	0	400
4	8572499	1029	ID:1	P	0.18.116.1.0.1.1.1	3	11647	251334	2987	2497	0	1879	1937	63568	0	896	0	564
5	10572864	1285	ID:1	P	0.18.116.1.0.1.1.1	4	14084	314409	2987	3814	0	2179	2434	63075	0	1317	0	300
6	12573582	1541	ID:1	P	0.18.116.1.0.1.1.1	5	20338	373666	5971	4238	0	2529	6252	59257	2984	424	0	350
7	14572873	1797	ID:1	P	0.18.116.1.0.1.1.1	6	22651	436861	5971	5618	0	3221	2311	63195	0	1380	0	692
8	16573209	2053	ID:1	P	0.18.116.1.0.1.1.1	7	25251	499768	5971	7315	0	4448	2598	62907	0	1697	0	1227
9	18572505	2309	ID:1	P	0.18.116.1.0.1.1.1	8	26940	563591	5971	7715	0	4848	1686	63823	0	400	0	400
10	20572521	2565	ID:1	P	0.18.116.1.0.1.1.1	9	28772	627270	5971	8302	0	5223	1829	63679	0	587	0	375
11	22572919	2821	ID:1	P	0.18.116.1.0.1.1.1	10	30594	690956	5971	8919	0	5820	1819	63686	0	617	0	597
12	24572920	3077	ID:1	P	0.18.116.1.0.1.1.1	11	32716	754343	5971	9717	0	6170	2119	63387	0	798	0	350
13	26573301	3333	ID:1	P	0.18.116.1.0.1.1.1	12	34533	818037	5971	10395	0	6557	1814	63694	0	678	0	387
14	28574800	3589	ID:1	P	0.18.116.1.0.1.1.1	13	42446	875638	9471	11580	0	7116	7910	57601	3500	1185	0	559
15	30574365	3845	ID:1	P	0.18.116.1.0.1.1.1	14	45251	938363	9471	13129	0	8017	2803	62725	0	1549	0	901
16	32574106	4101	ID:1	P	0.18.116.1.0.1.1.1	15	47566	1001541	9471	14181	0	8354	2312	63178	0	1052	0	337
17	34573644	4357	ID:1	P	0.18.116.1.0.1.1.1	16	49246	1065371	9471	14581	0	8754	1677	63830	0	400	0	400
18	36573656	4613	ID:1	P	0.18.116.1.0.1.1.1	17	50952	1129176	9471	14981	0	9154	1703	63805	0	400	0	400
19	38573648	4869	ID:1	P	0.18.116.1.0.1.1.1	18	52628	1193011	9471	15381	0	9554	1673	63835	0	400	0	400
20	40573997	5125	ID:1	P	0.18.116.1.0.1.1.1	19	54467	1256684	9471	16158	0	10119	1836	63673	0	777	0	565
21	42573976	5381	ID:1	P	0.18.116.1.0.1.1.1	20	56558	1320105	9471	17026	0	10481	2088	63421	0	868	0	362
22	44573995	5637	ID:1	P	0.18.116.1.0.1.1.1	21	58485	1383690	9471	17590	0	10856	1924	63585	0	564	0	375
23	46575735	5893	ID:1	P	0.18.116.1.0.1.1.1	22	64799	1442887	11534	19719	0	11361	6311	59197	2063	2129	0	505
24	48576079	6149	ID:1	P	0.18.116.1.0.1.1.1	23	71257	1501942	13665	22686	0	12831	6455	59055	2131	2967	0	1470
25	50575102	6405	ID:1	P	0.18.116.1.0.1.1.1	24	74522	1564190	14177	23570	0	13206	3262	62248	512	884	0	375

Figure A6 Malicious “blackhole-mote1.csv”—1 to 25 records.

A1.3.2 - “blackhole-mote4.csv”

The generated malicious “blackhole-mote4.csv” file, related to the benign UDP-client mote4, consists of 1,799 records and its first 25 records (i.e., 1–25) are depicted below in Figure .

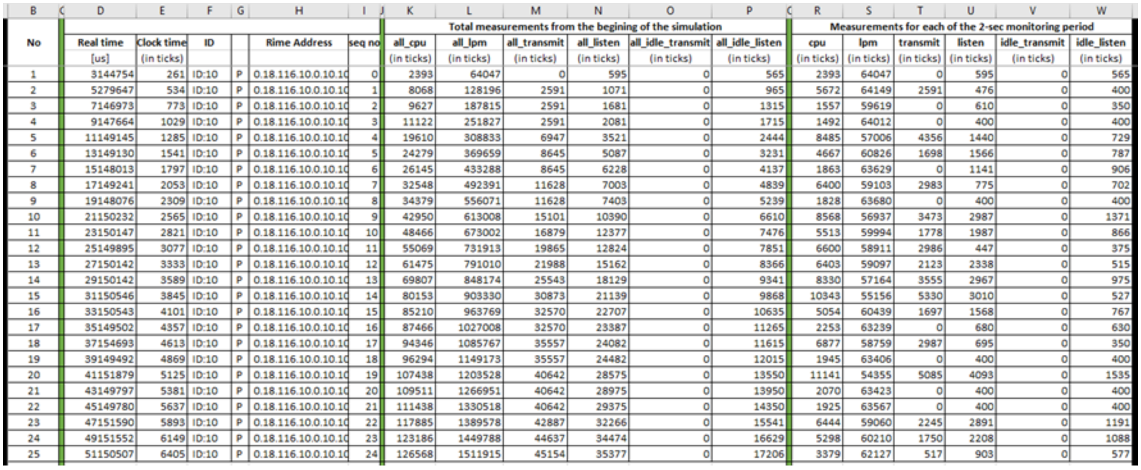


B	D	E	F	G	H	I	K	L	M	N	O	P	R	S	T	U	V	W
							Total measurements from the begining of the simulation						Measurements for each of the 2-sec monitoring period					
No	Real time	Clock time	ID		Rime Address	seq no	all_cpu	all_lpm	all_transmit	all_listen	all_idle_transmit	all_idle_listen	cpu	lpm	transmit	listen	idle_transmit	idle_listen
	[us]	(in ticks)					(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)
1	3072282	261	ID:4	P	0.18.116.4.0.4.4.4	0	2366	64075	0	588	0	552	2366	64075	0	588	0	552
2	5072672	517	ID:4	P	0.18.116.4.0.4.4.4	1	3458	128477	0	988	0	952	1089	64402	0	400	0	400
3	7073395	773	ID:4	P	0.18.116.4.0.4.4.4	2	4716	192715	0	1580	0	1339	1255	64238	0	592	0	387
4	9073715	1029	ID:4	P	0.18.116.4.0.4.4.4	3	5853	257072	0	1980	0	1739	1134	64357	0	400	0	400
5	11073903	1285	ID:4	P	0.18.116.4.0.4.4.4	4	7440	320993	0	2846	0	2291	1584	63921	0	866	0	552
6	13076528	1541	ID:4	P	0.18.116.4.0.4.4.4	5	11762	382177	1619	4152	0	2868	4319	61184	1619	1306	0	577
7	15076216	1797	ID:4	P	0.18.116.4.0.4.4.4	6	17904	441532	4605	4769	0	3413	6139	59355	2986	617	0	545
8	17075289	2053	ID:4	P	0.18.116.4.0.4.4.4	7	19760	505177	4605	5379	0	3788	1854	63645	0	610	0	375
9	19075210	2309	ID:4	P	0.18.116.4.0.4.4.4	8	21349	569098	4605	5779	0	4188	1586	63921	0	400	0	400
10	21076561	2565	ID:4	P	0.18.116.4.0.4.4.4	9	27362	628594	6950	7804	0	5154	6010	59496	2345	2025	0	966
11	23077456	2821	ID:4	P	0.18.116.4.0.4.4.4	10	37106	684361	11229	9486	0	5878	9741	55767	4279	1682	0	724
12	25076361	3077	ID:4	P	0.18.116.4.0.4.4.4	11	39558	747420	11229	10109	0	6455	2449	63059	0	623	0	577
13	27076325	3333	ID:4	P	0.18.116.4.0.4.4.4	12	42011	810476	11229	10969	0	7019	2450	63056	0	860	0	564
14	29078167	3589	ID:4	P	0.18.116.4.0.4.4.4	13	46631	871367	12522	13055	0	8494	4617	60891	1293	2086	0	1475
15	31077761	3845	ID:4	P	0.18.116.4.0.4.4.4	14	53256	930253	15028	15078	0	9399	6622	58886	2506	2023	0	905
16	33076317	4101	ID:4	P	0.18.116.4.0.4.4.4	15	55303	993714	15028	15668	0	9989	2044	63461	0	590	0	590
17	35078139	4357	ID:4	P	0.18.116.4.0.4.4.4	16	61906	1052625	18013	16116	0	10364	6600	58911	2985	448	0	375
18	37077130	4613	ID:4	P	0.18.116.4.0.4.4.4	17	64411	1115630	18013	16745	0	10739	2502	63005	0	629	0	375
19	39081085	4869	ID:4	P	0.18.116.4.0.4.4.4	18	67224	1178455	18013	17687	0	11331	2810	62825	0	942	0	592
20	41078735	5125	ID:4	P	0.18.116.4.0.4.4.4	19	73951	1237108	20920	20029	0	12453	6724	58653	2907	2342	0	1122
21	43077068	5381	ID:4	P	0.18.116.4.0.4.4.4	20	76033	1300539	20920	20429	0	12853	2079	63431	0	400	0	400
22	45077060	5637	ID:4	P	0.18.116.4.0.4.4.4	21	78025	1364057	20920	20829	0	13253	1989	63518	0	400	0	400
23	47077111	5893	ID:4	P	0.18.116.4.0.4.4.4	22	80045	1427547	20920	21786	0	14210	2017	63490	0	957	0	957
24	49077089	6149	ID:4	P	0.18.116.4.0.4.4.4	23	82124	1490980	20920	22730	0	15154	2076	63433	0	944	0	944
25	51077073	6405	ID:4	P	0.18.116.4.0.4.4.4	24	84116	1554497	20920	23130	0	15554	1989	63517	0	400	0	400

Figure A7 Malicious “blackhole-mote4.csv”—1 to 25 records.

A1.3.3 - “blackhole-mote10.csv”

The generated “blackhole-mote10.csv” file, related to the malicious UDP-client mote10, consists of 1,799 records and its first 25 records (i.e., 1–25) are depicted below in Figure .



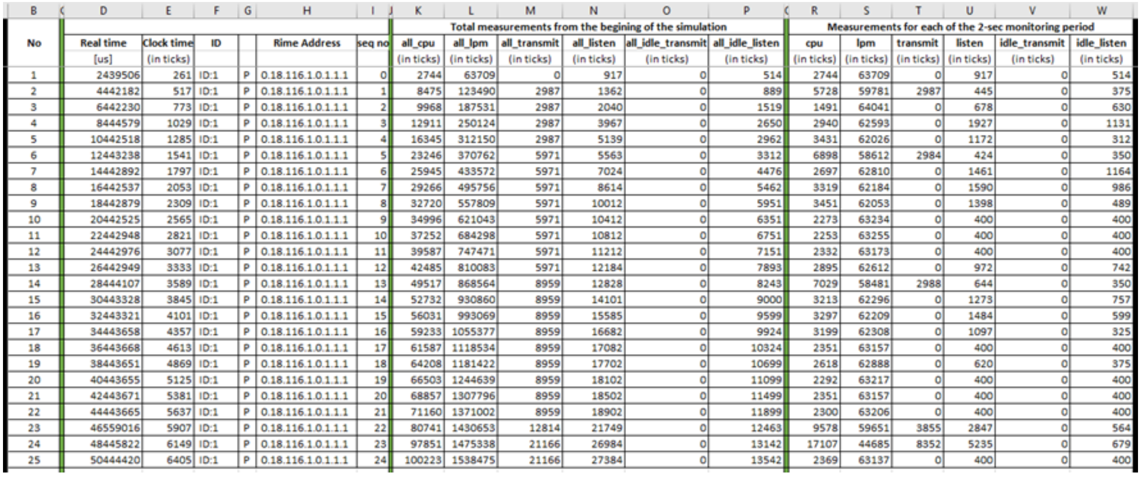
B	D	E	F	G	H	I	K	L	M	N	O	P	R	S	T	U	V	W
							Total measurements from the begining of the simulation						Measurements for each of the 2-sec monitoring period					
No	Real time	Clock time	ID		Rime Address	seq no	all_cpu	all_lpm	all_transmit	all_listen	all_idle_transmit	all_idle_listen	cpu	lpm	transmit	listen	idle_transmit	idle_listen
	[us]	(in ticks)					(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)
1	3144754	261	ID:10	P	0.18.116.10.0.10.10	0	2393	64047	0	595	0	565	2393	64047	0	595	0	565
2	5279647	534	ID:10	P	0.18.116.10.0.10.10	1	8068	128196	2591	1071	0	965	5672	64149	2591	476	0	400
3	7146973	773	ID:10	P	0.18.116.10.0.10.10	2	9627	187815	2591	1681	0	1315	1557	59619	0	610	0	350
4	9147664	1029	ID:10	P	0.18.116.10.0.10.10	3	11122	251827	2591	2081	0	1715	1492	64012	0	400	0	400
5	11149145	1285	ID:10	P	0.18.116.10.0.10.10	4	19610	308833	6947	3521	0	2444	8485	57006	4356	1440	0	729
6	13149130	1541	ID:10	P	0.18.116.10.0.10.10	5	24279	369659	8645	5087	0	3231	4667	60826	1698	1566	0	787
7	15148013	1797	ID:10	P	0.18.116.10.0.10.10	6	26145	433288	8645	6228	0	4137	1863	63629	0	1141	0	906
8	17149241	2053	ID:10	P	0.18.116.10.0.10.10	7	32548	492391	11628	7003	0	4839	6400	59103	2983	775	0	702
9	19148076	2309	ID:10	P	0.18.116.10.0.10.10	8	34379	556071	11628	7403	0	5239	1828	63680	0	400	0	400
10	21150232	2565	ID:10	P	0.18.116.10.0.10.10	9	42950	613008	15101	10390	0	6610	8568	56937	3473	2987	0	1371
11	23150147	2821	ID:10	P	0.18.116.10.0.10.10	10	48466	673002	16879	12377	0	7476	5513	59994	1778	1987	0	866
12	25149895	3077	ID:10	P	0.18.116.10.0.10.10	11	55069	731913	19865	12824	0	7851	6600	58911	2986	447	0	375
13	27150142	3333	ID:10	P	0.18.116.10.0.10.10	12	61475	791010	21988	15162	0	8366	6403	59097	2123	2338	0	515
14	29150142	3589	ID:10	P	0.18.116.10.0.10.10	13	69807	848174	25543	18129	0	9341	8330	57164	3555	2967	0	975
15	31150546	3845	ID:10	P	0.18.116.10.0.10.10	14	80153	903330	30873	21139	0	9868	10343	55156	5330	3010	0	527
16	33150543	4101	ID:10	P	0.18.116.10.0.10.10	15	85210	963769	32570	22707	0	10635	5054	60439	1697	1568	0	767
17	35149502	4357	ID:10	P	0.18.116.10.0.10.10	16	87466	1027008	32570	23387	0	11265	2253	63239	0	680	0	630
18	37154693	4613	ID:10	P	0.18.116.10.0.10.10	17	94346	1085767	35557	24082	0	11615	6877	58759	2987	695	0	350
19	39149492	4869	ID:10	P	0.18.116.10.0.10.10	18	96294	1149173	35557	24482	0	12015	1945	63406	0	400	0	400
20	41151879	5125	ID:10	P	0.18.116.10.0.10.10	19	107438	1203528	40642	28575	0	13550	11141	54355	5085	4093	0	1535
21	43149797	5381	ID:10	P	0.18.116.10.0.10.10	20	109511	1266951	40642	28975	0	13950	2070	63423	0	400	0	400
22	45149780	5637	ID:10	P	0.18.116.10.0.10.10	21	111438	1330518	40642	29375	0	14350	1925	63567	0	400	0	400
23	47151590	5893	ID:10	P	0.18.116.10.0.10.10	22	117885	1389578	42887	32266	0	15541	6444	59060	2245	2891	0	1191
24	49151552	6149	ID:10	P	0.18.116.10.0.10.10	23	123186	1449788	44637	34474	0	16629	5298	60210	1750	2208	0	1088
25	51150507	6405	ID:10	P	0.18.116.10.0.10.10	24	126568	1511915	45154	35377	0	17206	3379	62127	517	903	0	577

Figure A8 Malicious “blackhole-mote10.csv”—1 to 25 records.

A1.4 - Datasets for Sinkhole Attacks

A1.4.1 - “sinkhole-mote1.csv”

The generated “sinkhole-mote1.csv” file, related to the benign UDP-server mote1, consists of 1,799 records and its first 25 records (i.e., 1–25) are depicted in Figure .

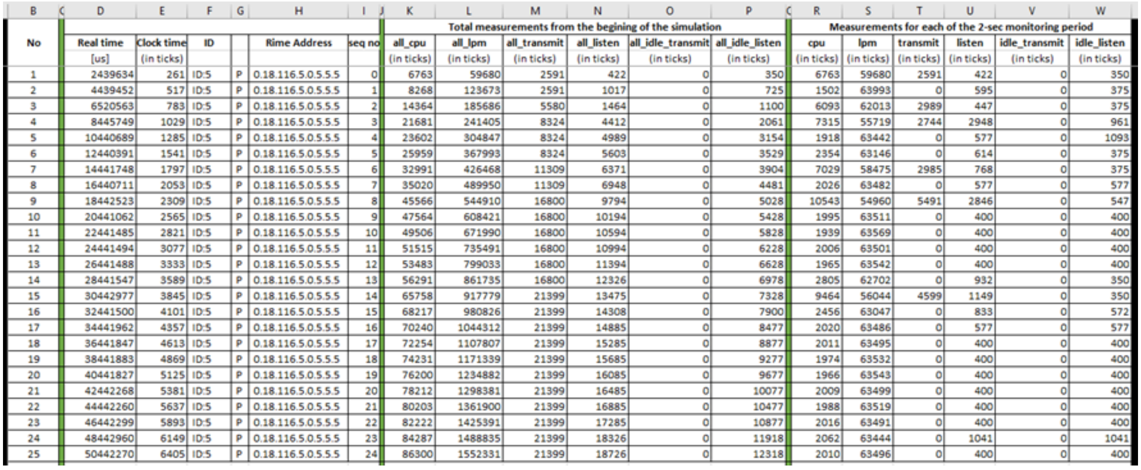


B	D	E	F	G	H	I	K	L	M	N	O	P	R	S	T	U	V	W
							Total measurements from the begining of the simulation						Measurements for each of the 2-sec monitoring period					
No	Real time	Clock time	ID		Rime Address	seq no	all_cpu	all_lpm	all_transmit	all_listen	all_idle_transmit	all_idle_listen	cpu	lpm	transmit	listen	idle_transmit	idle_listen
	[us]	(in ticks)					(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)
1	2439506	261	ID:1	P	0.18.116.1.0.1.1.1	0	2744	63709	0	917	0	514	2744	63709	0	917	0	514
2	4442182	517	ID:1	P	0.18.116.1.0.1.1.1	1	8475	123490	2987	1362	0	889	5728	59781	2987	445	0	375
3	6442230	773	ID:1	P	0.18.116.1.0.1.1.1	2	9968	187531	2987	2040	0	1519	1491	64041	0	678	0	630
4	8444579	1029	ID:1	P	0.18.116.1.0.1.1.1	3	12911	250124	2987	3967	0	2650	2940	62593	0	1927	0	1131
5	10442518	1285	ID:1	P	0.18.116.1.0.1.1.1	4	16345	312150	2987	5139	0	2962	3431	62026	0	1172	0	312
6	12443238	1541	ID:1	P	0.18.116.1.0.1.1.1	5	23246	370762	5971	5563	0	3312	6898	58612	2984	424	0	350
7	14442892	1797	ID:1	P	0.18.116.1.0.1.1.1	6	25945	433572	5971	7024	0	4476	2697	62810	0	1461	0	1164
8	16442537	2053	ID:1	P	0.18.116.1.0.1.1.1	7	29266	495756	5971	8614	0	5462	3319	62184	0	1590	0	986
9	18442879	2309	ID:1	P	0.18.116.1.0.1.1.1	8	32720	557809	5971	10012	0	5951	3451	62053	0	1398	0	489
10	20442525	2565	ID:1	P	0.18.116.1.0.1.1.1	9	34996	621043	5971	10412	0	6351	2273	63234	0	400	0	400
11	22442948	2821	ID:1	P	0.18.116.1.0.1.1.1	10	37252	684298	5971	10812	0	6751	2253	63255	0	400	0	400
12	24442976	3077	ID:1	P	0.18.116.1.0.1.1.1	11	39587	747471	5971	11212	0	7151	2332	63173	0	400	0	400
13	26442949	3333	ID:1	P	0.18.116.1.0.1.1.1	12	42485	810083	5971	12184	0	7893	2895	62612	0	972	0	742
14	28444107	3589	ID:1	P	0.18.116.1.0.1.1.1	13	49517	868564	8959	12828	0	8243	7029	58481	2988	644	0	350
15	30443328	3845	ID:1	P	0.18.116.1.0.1.1.1	14	52732	930860	8959	14101	0	9000	3213	62296	0	1273	0	757
16	32443321	4101	ID:1	P	0.18.116.1.0.1.1.1	15	56031	993069	8959	15585	0	9599	3297	62209	0	1484	0	599
17	34443658	4357	ID:1	P	0.18.116.1.0.1.1.1	16	59233	1055377	8959	16682	0	9924	3199	62308	0	1097	0	325
18	36443668	4613	ID:1	P	0.18.116.1.0.1.1.1	17	61587	1118534	8959	17082	0	10324	2351	63157	0	400	0	400
19	38443651	4869	ID:1	P	0.18.116.1.0.1.1.1	18	64208	1181422	8959	17702	0	10699	2618	62888	0	620	0	375
20	40443655	5125	ID:1	P	0.18.116.1.0.1.1.1	19	66503	1244639	8959	18102	0	11099	2292	63217	0	400	0	400
21	42443671	5381	ID:1	P	0.18.116.1.0.1.1.1	20	68857	1307796	8959	18502	0	11499	2351	63157	0	400	0	400
22	44443665	5637	ID:1	P	0.18.116.1.0.1.1.1	21	71160	1371002	8959	18902	0	11899	2300	63206	0	400	0	400
23	46559016	5907	ID:1	P	0.18.116.1.0.1.1.1	22	80741	1430653	12814	21749	0	12463	9578	59651	3855	2847	0	564
24	48445822	6149	ID:1	P	0.18.116.1.0.1.1.1	23	97851	1475338	21166	26984	0	13142	17107	44685	8352	5235	0	679
25	50444420	6405	ID:1	P	0.18.116.1.0.1.1.1	24	100223	1538475	21166	27384	0	13542	2369	63137	0	400	0	400

Figure A9 Malicious “sinkhole-mote1.csv”—1 to 25 records.

A1.4.2 - “sinkhole-mote5.csv”

The generated “sinkhole-mote5.csv” file, related to the benign UDP-client mote5, consists of 1,799 records and its first 25 records (i.e., 1–25) are depicted in Figure .

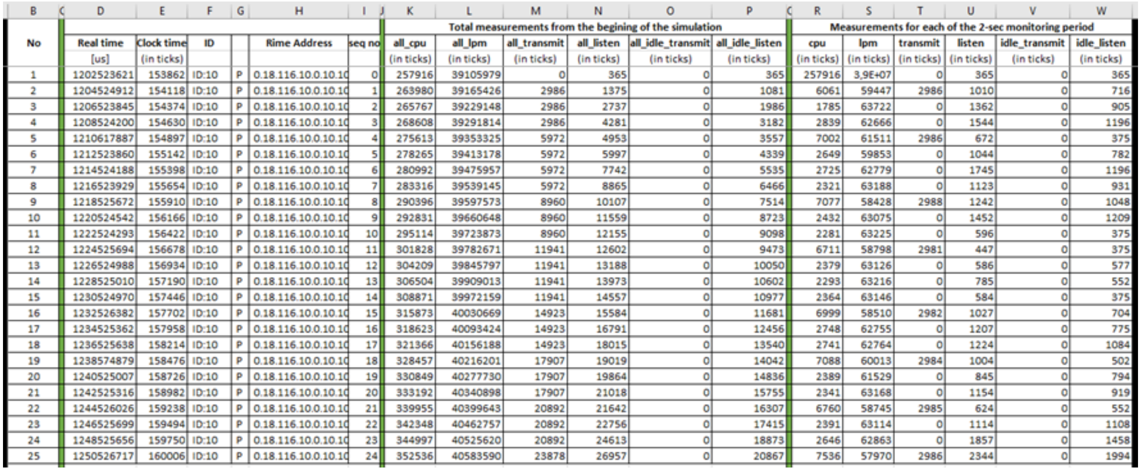


B	D	E	F	G	H	I	K	L	M	N	O	P	R	S	T	U	V	W
							Total measurements from the begining of the simulation						Measurements for each of the 2-sec monitoring period					
No	Real time	Clock time	ID		Rime Address	seq no	all_cpu	all_lpm	all_transmit	all_listen	all_idle_transmit	all_idle_listen	cpu	lpm	transmit	listen	idle_transmit	idle_listen
	[us]	(in ticks)					(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)
1	2439634	261	ID:5	P	0.18.116.5.0.5.5.5	0	6763	59680	2591	422	0	350	6763	59680	2591	422	0	350
2	4439452	517	ID:5	P	0.18.116.5.0.5.5.5	1	8268	123673	2591	1017	0	725	1502	63993	0	595	0	375
3	6520563	783	ID:5	P	0.18.116.5.0.5.5.5	2	14364	185686	5580	1464	0	1100	6093	62013	2989	447	0	375
4	8445749	1029	ID:5	P	0.18.116.5.0.5.5.5	3	21681	241405	8324	4412	0	2061	7315	55719	2744	2948	0	961
5	10440689	1285	ID:5	P	0.18.116.5.0.5.5.5	4	23602	304847	8324	4989	0	3154	1918	63442	0	577	0	1093
6	12440391	1541	ID:5	P	0.18.116.5.0.5.5.5	5	25959	367993	8324	5603	0	3529	2354	63146	0	614	0	375
7	14441748	1797	ID:5	P	0.18.116.5.0.5.5.5	6	32991	426468	11309	6371	0	3904	7029	58475	2985	768	0	375
8	16440711	2053	ID:5	P	0.18.116.5.0.5.5.5	7	35020	489950	11309	6948	0	4481	2026	63482	0	577	0	577
9	18442523	2309	ID:5	P	0.18.116.5.0.5.5.5	8	45566	544910	16800	9794	0	5028	10543	54960	5491	2846	0	547
10	20441062	2565	ID:5	P	0.18.116.5.0.5.5.5	9	47564	608421	16800	10194	0	5428	1995	63511	0	400	0	400
11	22441485	2821	ID:5	P	0.18.116.5.0.5.5.5	10	49506	671990	16800	10594	0	5828	1939	63569	0	400	0	400
12	24441494	3077	ID:5	P	0.18.116.5.0.5.5.5	11	51515	735491	16800	10994	0	6228	2006	63501	0	400	0	400
13	26441488	3333	ID:5	P	0.18.116.5.0.5.5.5	12	53483	799033	16800	11394	0	6628	1965	63542	0	400	0	400
14	28441547	3589	ID:5	P	0.18.116.5.0.5.5.5	13	56291	861735	16800	12326	0	6978	2805	62702	0	932	0	350
15	30442977	3845	ID:5	P	0.18.116.5.0.5.5.5	14	65758	917779	21399	13475	0	7328	9464	56044	4599	1149	0	350
16	32441500	4101	ID:5	P	0.18.116.5.0.5.5.5	15	68217	980826	21399	14308	0	7900	2456	63047	0	833	0	572
17	34441962	4357	ID:5	P	0.18.116.5.0.5.5.5	16	70240	1044312	21399	14885	0	8477	2020	63486	0	577	0	577
18	36441847	4613	ID:5	P	0.18.116.5.0.5.5.5	17	72254	1107807	21399	15285	0	8877	2011	63495	0	400	0	400
19	38441883	4869	ID:5	P	0.18.116.5.0.5.5.5	18	74231	1171339	21399	15685	0	9277	1974	63532	0	400	0	400
20	40441827	5125	ID:5	P	0.18.116.5.0.5.5.5	19	76200	1234882	21399	16085	0	9677	1966	63543	0	400	0	400
21	42442268	5381	ID:5	P	0.18.116.5.0.5.5.5	20	78212	1298381	21399	16485	0	10077	2009	63499	0	400	0	400
22	44442260	5637	ID:5	P	0.18.116.5.0.5.5.5	21	80203	1361900	21399	16885	0	10477	1988	63519	0	400	0	400
23	46442299	5893	ID:5	P	0.18.116.5.0.5.5.5	22	82222	1425391	21399	17285	0	10877	2016	63491	0	400	0	400
24	48442960	6149	ID:5	P	0.18.116.5.0.5.5.5	23	84287	1488835	21399	18326	0	11918	2062	63444	0	1041	0	1041
25	50442270	6405	ID:5	P	0.18.116.5.0.5.5.5	24	86300	1552331	21399	18726	0	12318	2010	63496	0	400	0	400

Figure A10 Malicious “sinkhole-mote5.csv”—1 to 25 records.

A1.4.3 - “sinkhole-mote10.csv”

The generated “sinkkhole-mote10.csv” file, related to the malicious UDP-server mote10, consists of 1,199 records and its first 25 records (i.e., 1–25) are depicted in Figure .



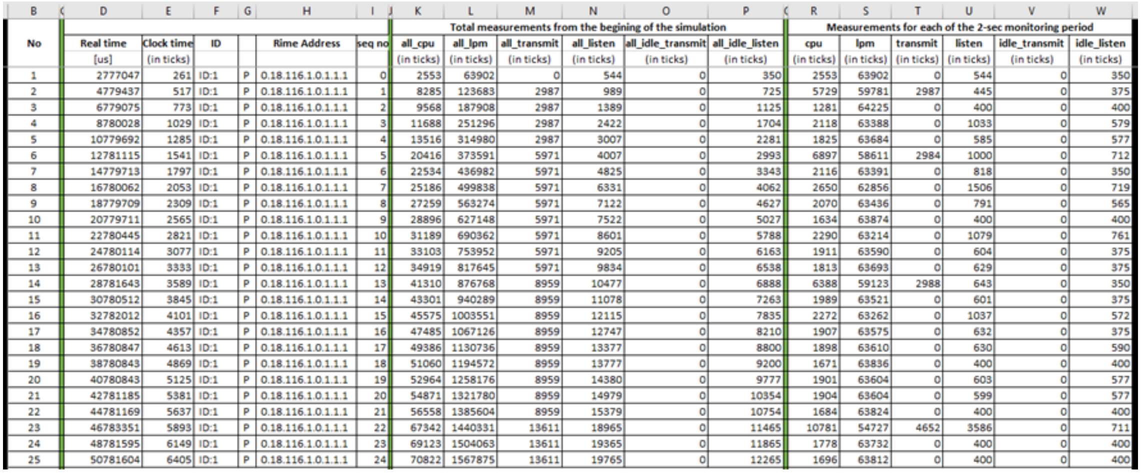
B	D	E	F	G	H	I	K	L	M	N	O	P	R	S	T	U	V	W
No							Total measurements from the begining of the simulation						Measurements for each of the 2-sec monitoring period					
	Real time	Clock time	ID		Rime Address	seq no	all_cpu	all_lpm	all_transmit	all_listen	all_idle_transmit	all_idle_listen	cpu	lpm	transmit	listen	idle_transmit	idle_listen
	[us]	(in ticks)					(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)
1	1202523621	153862	ID:10	P	0.18.116.10.0.10.10	0	257916	39105979	0	365	0	365	257916	3,9E+07	0	365	0	365
2	1204524912	154118	ID:10	P	0.18.116.10.0.10.10	1	263980	39165426	2986	1375	0	1081	6061	59447	2986	1010	0	716
3	1206523845	154374	ID:10	P	0.18.116.10.0.10.10	2	265767	39229148	2986	2737	0	1986	1785	63722	0	1362	0	905
4	1208524200	154630	ID:10	P	0.18.116.10.0.10.10	3	268608	39291814	2986	4281	0	3182	2839	62666	0	1544	0	1196
5	1210617887	154897	ID:10	P	0.18.116.10.0.10.10	4	275613	39353325	5972	4953	0	3557	7002	61511	2986	672	0	375
6	1212523860	155142	ID:10	P	0.18.116.10.0.10.10	5	278265	39413178	5972	5997	0	4339	2649	59853	0	1044	0	782
7	1214524188	155398	ID:10	P	0.18.116.10.0.10.10	6	280992	39475957	5972	7742	0	5535	2725	62779	0	1745	0	1196
8	1216523929	155654	ID:10	P	0.18.116.10.0.10.10	7	283316	39539145	5972	8865	0	6466	2321	63188	0	1123	0	931
9	1218525672	155910	ID:10	P	0.18.116.10.0.10.10	8	290396	39597573	8960	10107	0	7514	7077	58428	2988	1242	0	1048
10	1220524542	156166	ID:10	P	0.18.116.10.0.10.10	9	292831	39660648	8960	11559	0	8723	2432	63075	0	1452	0	1209
11	1222524293	156422	ID:10	P	0.18.116.10.0.10.10	10	295114	39723873	8960	12155	0	9098	2281	63225	0	596	0	375
12	1224525694	156678	ID:10	P	0.18.116.10.0.10.10	11	301828	39782671	11941	12602	0	9473	6711	58798	2981	447	0	375
13	1226524988	156934	ID:10	P	0.18.116.10.0.10.10	12	304209	39845797	11941	13188	0	10050	2379	63126	0	586	0	577
14	1228525010	157190	ID:10	P	0.18.116.10.0.10.10	13	306504	39909013	11941	13973	0	10602	2293	63216	0	785	0	552
15	1230524970	157446	ID:10	P	0.18.116.10.0.10.10	14	308871	39972159	11941	14557	0	10977	2364	63146	0	584	0	375
16	1232526382	157702	ID:10	P	0.18.116.10.0.10.10	15	315873	40030669	14923	15584	0	11681	6999	58510	2982	1027	0	704
17	1234525362	157958	ID:10	P	0.18.116.10.0.10.10	16	318623	40093424	14923	16791	0	12456	2748	62755	0	1207	0	775
18	1236525638	158214	ID:10	P	0.18.116.10.0.10.10	17	321366	40156188	14923	18015	0	13540	2741	62764	0	1224	0	1084
19	1238574879	158476	ID:10	P	0.18.116.10.0.10.10	18	328457	40216201	17907	19019	0	14042	7088	60013	2984	1004	0	502
20	1240525007	158726	ID:10	P	0.18.116.10.0.10.10	19	330849	40277730	17907	19864	0	14836	2389	61529	0	845	0	794
21	1242525316	158982	ID:10	P	0.18.116.10.0.10.10	20	333192	40340898	17907	21018	0	15755	2341	63168	0	1154	0	919
22	1244526026	159238	ID:10	P	0.18.116.10.0.10.10	21	339955	40399643	20892	21642	0	16307	6760	58745	2985	624	0	552
23	1246525699	159494	ID:10	P	0.18.116.10.0.10.10	22	342348	40462757	20892	22756	0	17415	2391	63114	0	1114	0	1108
24	1248525656	159750	ID:10	P	0.18.116.10.0.10.10	23	344997	40525620	20892	24613	0	18873	2646	62863	0	1857	0	1458
25	1250526717	160006	ID:10	P	0.18.116.10.0.10.10	24	352536	40583590	23878	26957	0	20867	7536	57970	2986	2344	0	1994

Figure A11 Malicious “sinkhole-mote10.csv”—1 to 25 records.

A1.5 - Datasets for Sleep Deprivation Attacks

A1.5.1 - “sleep_depr-mote1.csv”

The generated “sleep_depr-mote1.csv” file, related to the benign UDP-server mote1, consists of 1,799 records and its first 25 records (i.e., 1–25) are depicted below in Figure .

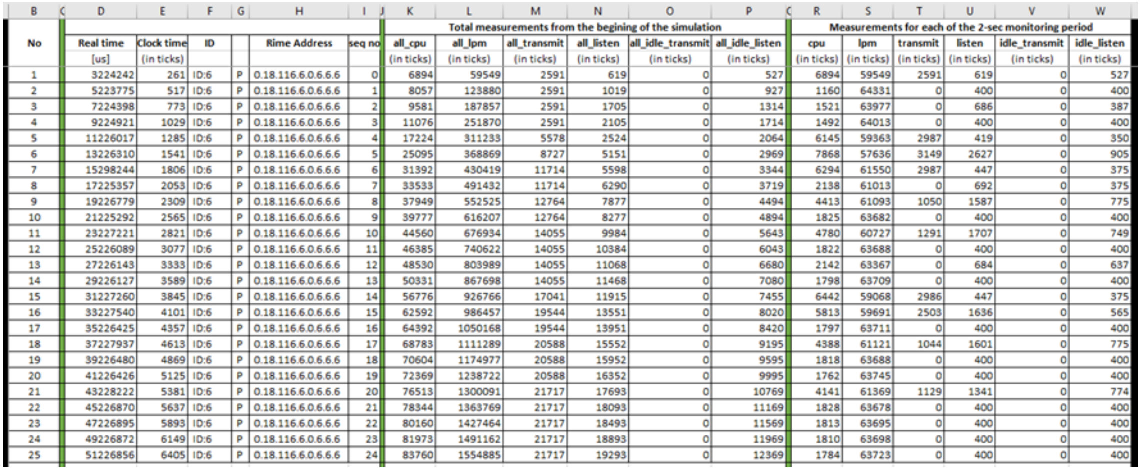


B	D	E	F	G	H	I	K	L	M	N	O	P	R	S	T	U	V	W
No							Total measurements from the begining of the simulation						Measurements for each of the 2-sec monitoring period					
	Real time	Clock time	ID		Rime Address	seq no	all_cpu	all_lpm	all_transmit	all_listen	all_idle_transmit	all_idle_listen	cpu	lpm	transmit	listen	idle_transmit	idle_listen
	[us]	(in ticks)					(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)
1	2777047	261	ID:1	P	0.18.116.1.0.1.1.1	0	2553	63902	0	544	0	350	2553	63902	0	544	0	350
2	4779437	517	ID:1	P	0.18.116.1.0.1.1.1	1	8285	123683	2987	989	0	725	5729	59781	2987	445	0	375
3	6779075	773	ID:1	P	0.18.116.1.0.1.1.1	2	9568	187908	2987	1389	0	1125	1281	64225	0	400	0	400
4	8780028	1029	ID:1	P	0.18.116.1.0.1.1.1	3	11688	251296	2987	2422	0	1704	2118	63388	0	1033	0	579
5	10779692	1285	ID:1	P	0.18.116.1.0.1.1.1	4	13516	314980	2987	3007	0	2281	1825	63684	0	585	0	577
6	12781115	1541	ID:1	P	0.18.116.1.0.1.1.1	5	20416	373591	5971	4007	0	2993	6897	58611	2984	1000	0	712
7	14779713	1797	ID:1	P	0.18.116.1.0.1.1.1	6	22534	436982	5971	4825	0	3343	2116	63391	0	818	0	350
8	16780062	2053	ID:1	P	0.18.116.1.0.1.1.1	7	25186	499838	5971	6331	0	4062	2650	62856	0	1506	0	719
9	18779709	2309	ID:1	P	0.18.116.1.0.1.1.1	8	27259	563274	5971	7122	0	4627	2070	63436	0	791	0	565
10	20779711	2565	ID:1	P	0.18.116.1.0.1.1.1	9	28896	627148	5971	7522	0	5027	1634	63874	0	400	0	400
11	22780445	2821	ID:1	P	0.18.116.1.0.1.1.1	10	31189	690362	5971	8601	0	5788	2290	63214	0	1079	0	761
12	24780114	3077	ID:1	P	0.18.116.1.0.1.1.1	11	33103	753952	5971	9205	0	6163	1911	63590	0	604	0	375
13	26780101	3333	ID:1	P	0.18.116.1.0.1.1.1	12	34919	817645	5971	9834	0	6538	1813	63693	0	629	0	375
14	28781643	3589	ID:1	P	0.18.116.1.0.1.1.1	13	41310	876768	8959	10477	0	6888	6388	59123	2988	643	0	350
15	30780512	3845	ID:1	P	0.18.116.1.0.1.1.1	14	43301	940289	8959	11078	0	7263	1989	63521	0	601	0	375
16	32782012	4101	ID:1	P	0.18.116.1.0.1.1.1	15	45575	1003551	8959	12115	0	7835	2272	63262	0	1037	0	572
17	34780852	4357	ID:1	P	0.18.116.1.0.1.1.1	16	47485	1067126	8959	12747	0	8210	1907	63575	0	632	0	375
18	36780847	4613	ID:1	P	0.18.116.1.0.1.1.1	17	49386	1130736	8959	13377	0	8800	1898	63610	0	630	0	590
19	38780843	4869	ID:1	P	0.18.116.1.0.1.1.1	18	51060	1194572	8959	13777	0	9200	1671	63836	0	400	0	400
20	40780843	5125	ID:1	P	0.18.116.1.0.1.1.1	19	52964	1258176	8959	14380	0	9777	1901	63604	0	603	0	577
21	42781185	5381	ID:1	P	0.18.116.1.0.1.1.1	20	54871	1321780	8959	14979	0	10354	1904	63604	0	599	0	577
22	44781169	5637	ID:1	P	0.18.116.1.0.1.1.1	21	56558	1385604	8959	15379	0	10754	1684	63824	0	400	0	400
23	46783351	5893	ID:1	P	0.18.116.1.0.1.1.1	22	67342	1440331	13611	18965	0	11465	10781	54727	4652	3586	0	711
24	48781595	6149	ID:1	P	0.18.116.1.0.1.1.1	23	69123	1504063	13611	19365	0	11865	1778	63732	0	400	0	400
25	50781604	6405	ID:1	P	0.18.116.1.0.1.1.1	24	70822	1567875	13611	19765	0	12265	1696	63812	0	400	0	400

Figure A12 Malicious “sleep_depr-mote1.csv”—1 to 25 records.

A1.5.2 - “sleep_depr-mote6.csv”

The generated “sleep_depr-mote6.csv” file, related to the benign UDP-client mote6, consists of 1,799 records and its first 25 records (i.e., 1–25) are depicted below in Figure .

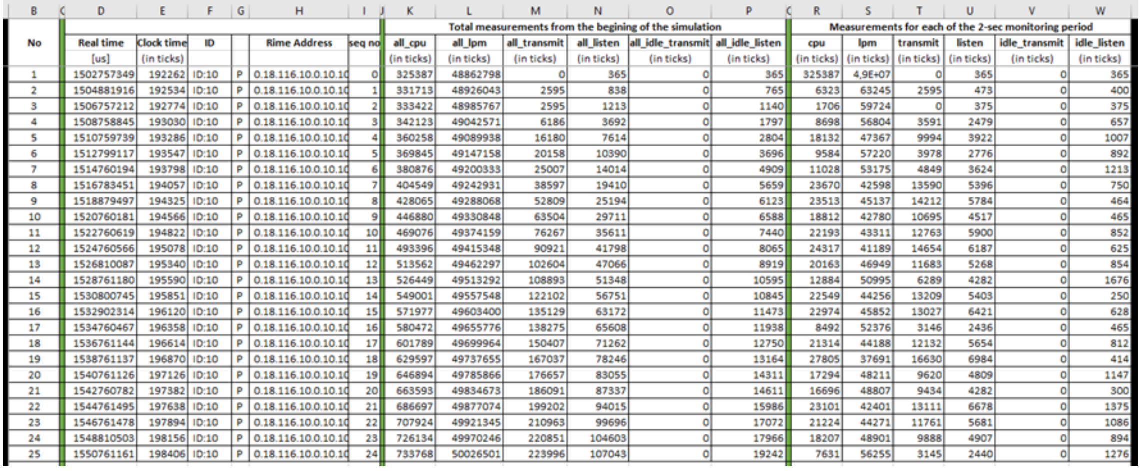


B	D	E	F	G	H	I	K	L	M	N	O	P	R	S	T	U	V	W
							Total measurements from the begining of the simulation						Measurements for each of the 2-sec monitoring period					
No	Real time	Clock time	ID		Rime Address	seq no	all_cpu	all_lpm	all_transmit	all_listen	all_idle_transmit	all_idle_listen	cpu	lpm	transmit	listen	idle_transmit	idle_listen
	[us]	(in ticks)					(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)
1	3224242	261	ID:6	P	0.18.116.6.0.6.6.6	0	6894	59549	2591	619	0	527	6894	59549	2591	619	0	527
2	5223775	517	ID:6	P	0.18.116.6.0.6.6.6	1	8057	123880	2591	1019	0	927	1160	64331	0	400	0	400
3	7224398	773	ID:6	P	0.18.116.6.0.6.6.6	2	9581	187857	2591	1705	0	1314	1521	63977	0	686	0	387
4	9224921	1029	ID:6	P	0.18.116.6.0.6.6.6	3	11076	251870	2591	2105	0	1714	1492	64013	0	400	0	400
5	11226017	1285	ID:6	P	0.18.116.6.0.6.6.6	4	17224	311233	5578	2524	0	2064	6145	59363	2987	419	0	350
6	13226310	1541	ID:6	P	0.18.116.6.0.6.6.6	5	25095	368869	8727	5151	0	2969	7868	57636	3149	2627	0	905
7	15298244	1806	ID:6	P	0.18.116.6.0.6.6.6	6	31392	430419	11714	5598	0	3344	6294	61550	2987	447	0	375
8	17225357	2053	ID:6	P	0.18.116.6.0.6.6.6	7	33533	491432	11714	6290	0	3719	2138	61013	0	692	0	375
9	19226779	2309	ID:6	P	0.18.116.6.0.6.6.6	8	37949	552525	12764	7877	0	4494	4413	61093	1050	1587	0	775
10	21225292	2565	ID:6	P	0.18.116.6.0.6.6.6	9	39777	616207	12764	8277	0	4894	1825	63682	0	400	0	400
11	23227221	2821	ID:6	P	0.18.116.6.0.6.6.6	10	44560	676934	14055	9984	0	5643	4780	60727	1291	1707	0	749
12	25226089	3077	ID:6	P	0.18.116.6.0.6.6.6	11	46385	740622	14055	10384	0	6043	1822	63688	0	400	0	400
13	27226143	3333	ID:6	P	0.18.116.6.0.6.6.6	12	48530	803989	14055	11068	0	6680	2142	63367	0	684	0	637
14	29226127	3589	ID:6	P	0.18.116.6.0.6.6.6	13	50331	867698	14055	11468	0	7080	1798	63709	0	400	0	400
15	31227260	3845	ID:6	P	0.18.116.6.0.6.6.6	14	56776	926766	17041	11915	0	7455	6442	59068	2986	447	0	375
16	33227540	4101	ID:6	P	0.18.116.6.0.6.6.6	15	62592	986457	19544	13551	0	8020	5813	59691	2503	1636	0	565
17	35226425	4357	ID:6	P	0.18.116.6.0.6.6.6	16	64392	1050168	19544	13951	0	8420	1797	63711	0	400	0	400
18	37227937	4613	ID:6	P	0.18.116.6.0.6.6.6	17	68783	1111289	20588	15552	0	9195	4388	61121	1044	1601	0	775
19	39226480	4869	ID:6	P	0.18.116.6.0.6.6.6	18	70604	1174977	20588	15952	0	9595	1818	63688	0	400	0	400
20	41226426	5125	ID:6	P	0.18.116.6.0.6.6.6	19	72369	1238722	20588	16352	0	9995	1762	63745	0	400	0	400
21	43228222	5381	ID:6	P	0.18.116.6.0.6.6.6	20	76513	1300091	21717	17693	0	10769	4141	61369	1129	1341	0	774
22	45226870	5637	ID:6	P	0.18.116.6.0.6.6.6	21	78344	1363769	21717	18093	0	11169	1828	63678	0	400	0	400
23	47226895	5893	ID:6	P	0.18.116.6.0.6.6.6	22	80160	1427464	21717	18493	0	11569	1813	63695	0	400	0	400
24	49226872	6149	ID:6	P	0.18.116.6.0.6.6.6	23	81973	1491162	21717	18893	0	11969	1810	63698	0	400	0	400
25	51226856	6405	ID:6	P	0.18.116.6.0.6.6.6	24	83760	1554885	21717	19293	0	12369	1784	63723	0	400	0	400

Figure A13 Malicious “sleep_depr-mote6.csv”—1 to 25 records.

A1.5.3 - “sleep_depr-mote10.csv”

The generated “sleep_depr-mote10.csv” file, related to the malicious UDP-client mote10, consists of 1,049 records and its first 25 records (i.e., 1–25) are depicted below in Figure .



B	D	E	F	G	H	I	K	L	M	N	O	P	R	S	T	U	V	W
							Total measurements from the begining of the simulation						Measurements for each of the 2-sec monitoring period					
No	Real time	Clock time	ID		Rime Address	seq no	all_cpu	all_lpm	all_transmit	all_listen	all_idle_transmit	all_idle_listen	cpu	lpm	transmit	listen	idle_transmit	idle_listen
	[us]	(in ticks)					(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)	(in ticks)
1	1502757349	192262	ID:10	P	0.18.116.10.0.10.1	0	325387	48862798	0	365	0	365	325387	4,9E+07	0	365	0	365
2	1504881916	192534	ID:10	P	0.18.116.10.0.10.1	1	331713	48926043	2595	838	0	765	6323	63245	2595	473	0	400
3	1506757212	192774	ID:10	P	0.18.116.10.0.10.1	2	333422	48985767	2595	1213	0	1140	1706	59724	0	375	0	375
4	1508758845	193030	ID:10	P	0.18.116.10.0.10.1	3	342123	49042571	6186	3692	0	1797	8698	56804	3591	2479	0	657
5	1510759739	193286	ID:10	P	0.18.116.10.0.10.1	4	360258	49089938	16180	7614	0	2804	18132	47367	9994	3922	0	1007
6	1512799117	193547	ID:10	P	0.18.116.10.0.10.1	5	369845	49147158	20158	10390	0	3696	9584	57220	3978	2776	0	892
7	1514760194	193798	ID:10	P	0.18.116.10.0.10.1	6	380876	49200333	25007	14014	0	4909	11028	53175	4849	3624	0	1213
8	1516783451	194057	ID:10	P	0.18.116.10.0.10.1	7	404549	49242931	38597	19410	0	5659	23670	42598	13590	5396	0	750
9	1518879497	194325	ID:10	P	0.18.116.10.0.10.1	8	428065	49288068	52809	25194	0	6123	23513	45137	14212	5784	0	464
10	1520760181	194566	ID:10	P	0.18.116.10.0.10.1	9	446880	49330848	63504	29711	0	6588	18812	42780	10695	4517	0	465
11	1522760619	194822	ID:10	P	0.18.116.10.0.10.1	10	469076	49374159	76267	35611	0	7440	22193	43311	12763	5900	0	852
12	1524760566	195078	ID:10	P	0.18.116.10.0.10.1	11	493396	49415348	90921	41798	0	8065	24317	41189	14654	6187	0	625
13	1526810087	195340	ID:10	P	0.18.116.10.0.10.1	12	513562	49462297	102604	47066	0	8919	20163	46949	11683	5268	0	854
14	1528761180	195590	ID:10	P	0.18.116.10.0.10.1	13	526449	49513292	108893	51348	0	10595	12884	50995	6289	4282	0	1676
15	1530800745	195851	ID:10	P	0.18.116.10.0.10.1	14	549001	49557548	122102	56751	0	10845	22549	44256	13209	5403	0	250
16	1532902314	196120	ID:10	P	0.18.116.10.0.10.1	15	571977	49603400	135129	63172	0	11473	22974	45852	13027	6421	0	628
17	1534760467	196358	ID:10	P	0.18.116.10.0.10.1	16	580472	49655776	138275	65608	0	11938	8492	52376	3146	2436	0	465
18	1536761144	196614	ID:10	P	0.18.116.10.0.10.1	17	601789	49699964	150407	71262	0	12750	21314	44188	12132	5654	0	812
19	1538761137	196870	ID:10	P	0.18.116.10.0.10.1	18	629597	49737655	167037	78246	0	13164	27805	37691	16630	6984	0	414
20	1540761126	197126	ID:10	P	0.18.116.10.0.10.1	19	646894	49785866	176657	83055	0	14311	17294	48211	9620	4809	0	1147
21	1542760782	197382	ID:10	P	0.18.116.10.0.10.1	20	663593	49834673	186091	87337	0	14611	16696	48807	9434	4282	0	300
22	1544761495	197638	ID:10	P	0.18.116.10.0.10.1	21	686697	49877074	199202	94015	0	15986	23101	42401	13111	6678	0	1375
23	1546761478	197894	ID:10	P	0.18.116.10.0.10.1	22	707924	49921345	210963	99696	0	17072	21224	44271	11761	5681	0	1086
24	1548810503	198156	ID:10	P	0.18.116.10.0.10.1	23	726134	49970246	220851	104603	0	17966	18207	48901	9888	4907	0	894
25	1550761161	198406	ID:10	P	0.18.116.10.0.10.1	24	733768	50026501	223996	107043	0	19242	7631	56255	3145	2440	0	1276

Figure A14 Malicious “sleep_depr-mote10.csv”—1 to 25 records.