Insights into peer-to-peer botnet dynamics: reviewing emulation testbeds and proposing a conceptual model

Mithiiran Parthipan[1], Shams Ul Arfeen Laghari[2], Ashish Jaisan[1], Amber Baig[3], Muhammad Asim Ali[4], Shankar Karuppayah[1,5]

[1]National Advanced IPv6 Centre, Universiti Sains Malaysia, Penang, Malaysia
[2]School of Information and Communication Technology, Faculty of Engineering Design Information and Communications Technology (EDICT), Bahrain Polytechnic, Isa Town, Bahrain
[3]Department of Computer Science, Faculty of Engineering, Science and Technology, Isra University, Hyderabad, Pakistan
[4]Department of Electrical Engineering, Sukkur IBA University, Sukkur, Pakistan
[5]BitRanger Sdn Bhd, Penang, Malaysia

Article Info

Article history:

Received May 1, 2024
Revised Nov 21, 2024
Accepted Nov 28, 2024

Keywords:

Botnet
Cybersecurity
Malware
Peer-to-peer
Testbed

ABSTRACT

Peer-to-peer (P2P) botnets have emerged as a resilient cybercrime tool, utilizing decentralized architectures to evade detection and complicate takedown efforts. Existing botnet emulation testbeds often fall short in replicating the dynamic and large-scale environments that these botnets operate in, limiting their effectiveness in research and defense strategy development. This paper addresses these gaps by proposing a scalable, flexible emulation testbed for P2P botnets that integrates advanced virtualization and automation technologies. Our framework enables the accurate emulation of real-world botnet behaviors without relying on reverse engineering, offering researchers a secure and adaptable environment to test and validate botnet detection and mitigation strategies. The testbed's dynamic scalability and robust configuration management streamline experimentation across diverse network topologies and botnet types. Our results show that this approach significantly enhances the ability to study P2P botnets in a controlled, reproducible setting, providing valuable insights for advancing cybersecurity defenses.

Corresponding Author:

Shankar Karuppayah
National Advanced IPv6 Centre, Universiti Sains Malaysia
Gelugor, 11800 Penang, Malaysia
Email: kshankar@usm.my

1. INTRODUCTION

The internet has transformed how we live, work, and interact with the world. It provides a limitless source of information, facilitates instant connection across countries, and accelerates innovation at an unprecedented rate. However, this interconnected landscape harbors a dark side. Malicious actors exploit vulnerabilities with increasing sophistication, employing tools like botnets, ransomware, sniffers, and trojan horses to wreak havoc [1]. From the techniques mentioned, botnets are the most innovative and are becoming the main topic of discussion in distributed computing [2]. Early botnets operated under the client-server model, in which all bots are managed by a single and centralized command and control server (C2) [3]. Centralized servers, however, put the entire botnet at risk as they act as a single point of failure that is simple to shut down. One improvement over centralized botnet is the paradigm shift of peer-to-peer (P2P) botnet. Unlike their centralized counterparts, P2P botnets dismantle the single point of failure, creating a decentralized network of infected devices. Each compromised device functions as client and server, adapting its role within the ever-shifting network. This resilience presents a significant challenge, highlighting the need for innovative defense mechanisms to counter this sophisticated threat [4].

In contemporary times, botmasters [5] employ swift registration click buttons and deploy a multitude of IP addresses distributed globally. While click fraud operations yield substantial profits for attackers and their accomplices, they pose a significant threat to content providers and advertisers, emerging as a growing menace to the e-commerce landscape [6]. Botnets also typically target smart home apps because of their widespread use and lax security, which makes devices easily compromised [7]. In this case, the attacker may attack a crucial service by using hacked devices (bots) as part of a botnet [8]. In addition, botnets pose an increasing danger to power system stability, where attackers can weaponize many bots to cause cascading effects inside the system, avoiding direct attacks on vital infrastructure [9].

Several infamous botnet attacks have carved deep scars across the landscape of cyberspace. These meticulously orchestrated campaigns, perpetrated by malevolent actors, have significantly disrupted essential online services. One example is a significant distributed denial of service (DDoS) attack directed against Krebs on security [10] and OVH in mid-September 2016. The botnet behind this attack is Mirai, which specifically aims at internet of things (IoT) devices. Mirai was first reported on August 31, 2016, and there were multiple notable attacks in addition to those previously listed [11]. Another important event in this chronology is the public dissemination of Mirai's source code. The release of this source code resulted in the spread of various Mirai variants controlled by different operators. A significant variation included a feature that allowed for a router vulnerability to be exploited using the customer premises equipment wide area network (WAN) control protocol (CWMP) [12], [13]. In November 2016, an exploit by the Mirai variant caused an outage at Deutsche Telekom, and the alleged attacker was only caught in February 2017 [14]. Another regular malicious activity bots conduct is the theft of sensitive information from compromised machines. Botnets steal important information such as website login credentials, cookies, credit cards, banking accounts, and passwords. Botnets employ many techniques to steal the information described above, i.e., Torpig [15] conducts man-in-the-browser phishing attempts to obtain bank account and credit card information. The Torpig configuration file includes approximately 300 domains of banks and financial institutions. It also pilfers a range of additional personal data. Next, Zeus, the largest bank theft botnet, infected at least 200,000 devices worldwide and was responsible for over 100 million US dollars in 3 years [16], [17].

Botnets have also impacted the healthcare industry's infrastructure. There is cause for alarm due to the pro-Russian hacktivist group Killnet's surge in DDoS attacks against US healthcare institutions that lasted 48 hours [18]. These kinds of attacks aim to flood a system or network with traffic, making it difficult or impossible for users to obtain critical healthcare services. Since doctors and other healthcare providers need access to patient data and systems to diagnose and treat patients properly, this might have a disastrous effect on the healthcare industry. The potential attack surface is further increased by the development of electric cars and the associated infrastructure for charging them, which has a significant energy consumption [19]. As a result, there is a strong reason to investigate these botnets to identify, evaluate, and create efficient defenses against them. The observed surge in global botnet activity from December 2023 to January 2024, which is shown in Figure 1, underscores a significant cybersecurity challenge. The escalation commenced with an alarming spike to 35,144 infected devices on December 8th, 2023, followed by a subsequent surge to 43,194 on December 20th, 2023, surpassing the typical daily median of 10,000. This upward trend peaked on December 29th, 2023, with a staggering 143,957 devices engaged in malicious activities—an alarming tenfold increase compared to normal volumes. Remarkably, this heightened activity persisted into January, with daily surges ranging between 50,000 to 100,000 infected devices. Notably, January 5th and 6th, 2024, witnessed an unprecedented escalation, with over a million infected devices detected per day (1,294,416 and 1,134,999, respectively), reaching unprecedented levels of concern. This surge in botnet activity underscores the evolving threat landscape, emphasizing the critical importance of robust cybersecurity measures to mitigate the risks posed by these increasingly sophisticated attacks [20].

The rationale for investigating botnets remains unambiguous. However, the research methodology demands careful consideration due to the inherent complexities of botnet architectures. These complexities arise from the vast spectrum of targeted devices and systems encompassing critical infrastructure. A particularly concerning scenario arises when bots infiltrate medical devices within hospitals [21]-[23]. In such cases, a takedown attempt or even monitoring the botnet could inadvertently disrupt essential medical care, endangering numerous innocent lives. In addition, monitoring attempts of P2P botnets have yielded limited success and caused the researchers to suffer from retaliatory DDoS from botmasters that detected monitoring activities within their botnet [24]. Considering the dangers involved in researching botnets in the wild, a secure, controlled, and realistic testbed will provide a crucial solution to studying botnets. These testbeds allow researchers to emulate botnet behavior and communication patterns without jeopardizing real-world systems, potentially harming innocent people and themselves. In essence, testbeds function as a haven for botnet research. They provide a platform for researchers to delve into the intricacies of these malicious networks, developing effective detection and mitigation strategies without the ethical and practical concerns associated with studying botnets in the uncontrolled environment of the live internet.

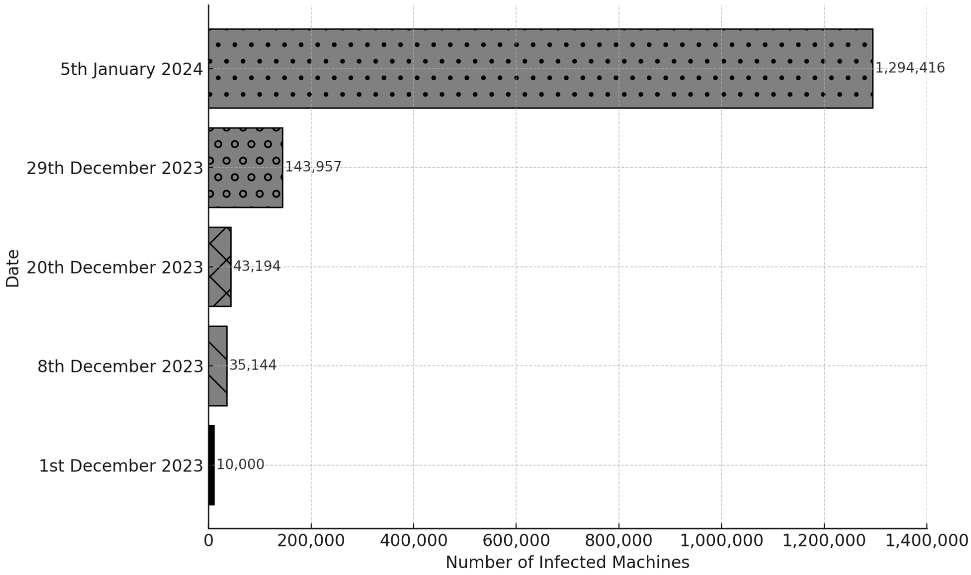


Figure 1. Statistics of botnet infected machines

Despite significant advancements in botnet research, there remain substantial challenges in effectively analyzing and combating P2P botnets. Current testbeds often suffer from limitations such as small-scale network emulations and a heavy reliance on reverse engineering, which restrict their ability to replicate the complex, large-scale environments where P2P botnets operate accurately. These constraints hinder the development of robust detection and mitigation strategies that adapt to botnet threats' evolving nature. While studies like BotsideP2P and the 3000-node testbed have made strides in botnet emulation, they have not fully addressed these critical gaps. This study proposes a framework that overcomes these issues by enabling dynamic scaling of the testbed according to the specific type of botnet under investigation, without the need for reverse engineering. The proposed approach allows for the emulation of realistic P2P botnet networks, offering a more effective platform for testing and refining cybersecurity measures. The implications of proposed framework are significant, as it provides a flexible, scalable solution that supports the ongoing development of advanced defense mechanisms, ensuring that research can keep pace with the increasingly sophisticated tactics employed by botnets.

This paper focuses solely on discussing numerous P2P botnet topics, including comparing centralized botnet (traditional) designs and P2P botnet designs, P2P detection methods, and related work in P2P botnet emulation testbeds. The fundamental problem in the existing botnet research, particularly in P2P botnets, is the lack of real network traffic datasets that could be used to propose or even improve advanced monitoring systems. One possible solution to this problem is to use a P2P emulation Testbed for producing network traffic datasets, validation, and verification with the ground truth, e.g., the total number of infected machines. Using such testbeds provides opportunities to test the effectiveness of advanced monitoring mechanisms and provide a complete understanding of the bot's behaviours. The main contribution of this paper is to provide a conceptual framework of an emulation testbed for P2P botnet analysis that could contribute an educational resource and some level of insights for students and researchers in this domain.

2. BACKGROUND

This section aims to provide a concise description of botnet design and attack models. First, we discuss the centralized botnet design, followed by the decentralized and P2P botnet design. We then elaborated on the existing mitigation schemes. Finally, we will provide the summary for this section.

2.1. Botnet design paradigms

Traditional botnets use a C2 server, and the botmaster issues commands through it, which resembles a centralized architecture as shown in Figure 2. The C2s are commonly deployed on either self-deployed IRC servers or compromised web servers [25]. Bots within a centralized botnet frequently check with the C2 server for the latest updates from botmasters [5] and promptly implement them. Even though centralized C2s are simple to implement, they are vulnerable to becoming a single point of failure, such as in botnet takedown operations [26]. Disabling the C2 server makes the botnet unable to receive commands or communicate with the botmaster. Defenders can efficiently list all affected workstations by analyzing the server's communication records. The centralized botnet design follows a client-server model where the bot primarily receives control commands from the C2 through polling, and the botmasters send these commands to the bots via these servers. Centralized botnets offer benefits, including straightforward deployment, optimal performance, and effective organization, but their management is vulnerable to a single point of failure [27].

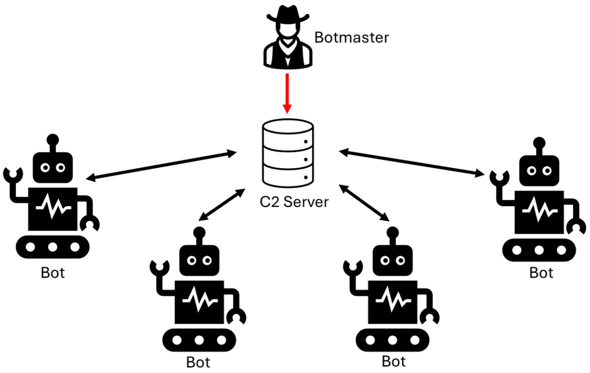


Figure 2. Centralised botnet architecture

In contrast, P2P botnets provide an overlay network for communication between the bots involved rather than depending on a central server. The botmaster can disseminate commands by exploiting any compromised device within the P2P network, as illustrated in Figure 3. It is more resistant to external attacks on its functioning compared to a centralized C2 botnet [24]. There are two methods of deploying a P2P network; one such method is deploying in a structured mode. Kademlia [28] is a structured P2P network where it utilizes a distributed hash table (DHT) [29]. Another mode of P2P network is unstructured P2P overlays that lack a specific organization yet sustain connectivity through a membership management (MM) mechanism [30]. The primary distinction between structured and unstructured P2P networks in botnets is in the challenge of surveillance where structured botnets like Storm [31] can be effectively monitored [32]. Unstructured P2P botnets like Sality [33], GameOver Zeus [17], and ZeroAccess [34] utilize unstructured P2P overlays. The absence of a structure in these networks hinders efficient methods commonly used in structured P2P networks, making them harder to monitor [35].

2.2. Botnet countermeasures

Botnet countermeasures can be divided into three categories: detection, monitoring, and mitigation. Identifying and locating a botnet within a network is referred to as detection [36]. Monitoring a botnet is observing the botnet's operations and communication pattern. It can aid in a better understanding of a botmaster's intent and the botnet's behaviour and architecture. Two monitoring forms exist, namely, passive and active [37]. In passive monitoring, traps are deployed as new bots for the botnets to communicate, such as in the case of dark address space monitoring, where unused portions of public IP addresses are used to create the traps [38]. In active monitoring, bots are actively contacted to investigate their behaviors. Mitigation is getting rid of a botnet found, either by disinfecting all or most of the bots within it or by detecting and monitoring the botmaster's capacity to command and control the botnet [39]. The ultimate goal of botnet defense is mitigation after botnet detection and monitoring. Thus, blocking a botnet's C2 channels to isolate bots is known as botnet mitigation. This concept is easily applicable to centralized botnets, as the C2 traffic in a centralized botnet passes via one or more central servers that are well-known.

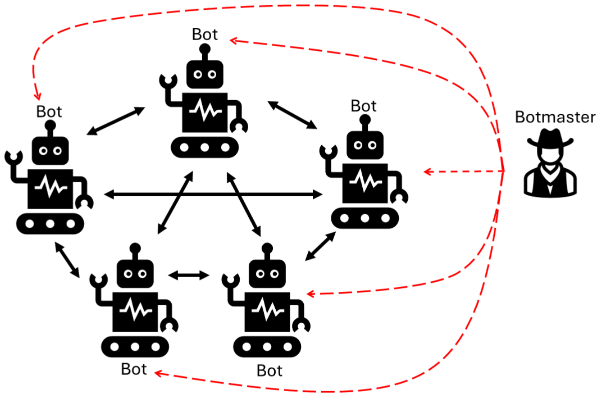


Figure 3. P2P botnet architecture

However, since a P2P botnet uses a P2P network to distribute critical messages, stopping the information from spreading is considerably more difficult [40]. A botmaster could select any bot within the P2P botnet to inject commands, which are then distributed to all the bots in the network. The ability to inject commands into any bot on the network gives the botmaster many access points. It hides the command's source to thwart any effort at a traceback, which makes them very resilient to monitoring and takedown attempts by researchers and law enforcement [24].

Due to the popularity of botnets within law enforcement and researchers, botmasters have equipped their botnets with various anti-monitoring mechanisms, such as blocklisting or restricted neighbor nodes return sizes. These anti-monitoring mechanisms are a great challenge to the successful monitoring attempts of the defenders [24]. Studying botnets in a real-world environment is crucial to understanding and mitigating their risks. One potential approach is conducting emulation tests in a lab setting with equipment and infrastructure [32], [41]. To facilitate botnet life cycle analysis, the experiments must be carried out in a testbed that mimics a portion of the internet. Additionally, it will aid in developing the new advanced monitoring mechanisms as the testbed enables realistic tests. Researchers will be able to develop precise, dependable, and performance-optimized countermeasure mechanisms as they will be able to validate the effectiveness of the mechanisms with the ground truth.

Furthermore, it's critical to have a testbed setup where new procedures and parameters may be tried and assessed to enhance the effectiveness of the testbed further, as a particular mitigation mechanism may only be effective to a particular botnet [42]. For example, network analysis is the main method of detecting botnets like Mirai and Zeus that were previously discussed [24]. Due to their large size, bots frequently send large amounts of traffic to nearby nodes through updates and queries. As a result, some botnets make no effort to conceal their existence. For example, the Mirai botnet exhibited a distinctive fingerprint due to the volume of compromised devices constantly communicating. This characteristic allowed system administrators to identify the attack quickly. Preventive measures like intrusion detection systems (IDS) or anti-virus software could be implemented to prevent new bots from infiltrating the network [32].

3. RELATED WORK

This section describes and discusses several proposed methodologies for botnet testbeds for network security. Due to the lack of P2P botnet analysis testbeds, we have included testbeds that analyze IoT botnets [43], HTTP botnets [44], and testbeds that aim to emulate attacks executed by botnets [45]. We have explained each testbed concerning the proposed taxonomy of the botnet emulation testbed, as shown in Figure 2.

3.1. Peer-to-peer botnet emulation testbed

The testbeds outlined in this section are designed to emulate P2P botnet topologies and behaviors. This section aims to imitate and mimic the typical characteristics of P2P botnets, enabling thorough research and testing in a controlled setting.

Beauchaine *et al.* [32] designed to simulate a botnet network named BotSideP2P using a Kademlia DHT [46] for routing. The network consists of three nodes: bootstrapper [24], commander, and worker. The bootstrapper node initializes new nodes and adds them to the DHT. Commander nodes send commands and monitor workers, while worker nodes start a listening server, join the network, and wait for commands or payloads. The bot programs operate asynchronously, allowing them to handle multiple tasks simultaneously. Payloads can be included in the program directory or downloaded remotely. The testbed was successfully refactored to use the Asyncio framework [47] and is now compatible with Python 3.5+. This refactoring has improved the testbed's long-term stability and ease of use. The security aspect for BotSideP2P is available as the developers isolated the testbed from internet connectivity. However, scalability and self-configuration criteria were absent, as virtualization and automation tools were not used to deploy nodes within the testbed. BotSideP2P was developed using Raspberry Pi [48]; thus, more physical Raspberry will be needed to scale the testbed. Next, diversity and fidelity are also not present as students and researchers developed the malware used for the experiments, thus it does not emulate the behaviors of a real botnet. Additionally, the researchers did not mention executing experiments using different P2P botnets. It is imperative to enhance the documentation of BotSideP2P, particularly focusing on improving the topology and node setup description. The existing documentation highlights the testbed's features but lacks sufficient detail to enable successful replication. Therefore, it is essential to refine the documentation to provide comprehensive guidance for users seeking to replicate the BotSideP2P testbed accurately.

In addition, Calvet *et al.* [41] developed a testbed that emulates a Waledac botnet [49] and the countermeasures against the latter. It uses virtual machine (VM) templates [50] for spammers and repeaters [51] and a C2 server implemented as a custom Python script. The botnet consists of the following components: repeaters, spammers, protectors, and the C2 server; however, the quantity for each component differs from those of the real Waledac botnet. Moreover, standalone Linux machines are running essential internet services such as domain name system (DNS) [52], simple mail transfer protocol (SMTP) [53], and dynamic host configuration protocol (DHCP) [54], reflecting an environment where a real botnet uses its bots. Hence, the testbed provides a means to conduct this integrated research, more precisely, it allows studying the currently operating Waledac botnet's functionality and possible ways to reduce its efficacy as countermeasures. From the security perspective, the testbed maintains isolation between the entire facility and the external world as another internet plane.

Meanwhile, the researchers deployed the virtualization infrastructure and pre-configured VM templates, which ensured the testbed's scalability. The testbed features scripts designed to administer commands to the VM, demonstrating self-configuration capabilities. Furthermore, utilising the real-life Waledac botnet, with minimal alterations to its binary structure [55], ensures a high level of fidelity within the testbed environment. Regarding topology flexibility, although the testbed offers the potential to emulate various types of botnets, the paper's focus is solely on different Waledac variants, which limits the diversity aspect to some extent. Reproducibility within the testbed is hindered due to its reliance on reverse engineering methodologies, which may not be universally applicable across all botnet scenarios. The literature presented in this section shares some similarities to our proposed testbed. However, both the testbeds presented earlier are deployed to analyze a specific type of P2P botnets. Thus, we are proposing a testbed that can be easily customized to analyze a wide variety of P2P botnets. Additionally, the researchers had a heavy dependence on the botnet's source code, which is not our research's focus. Our research focuses on methodologies that do not necessitate access to botnet source code. Botnet source code is typically not easily accessible because of the clandestine nature of these harmful programs.

3.2. Internet of things botnet emulation testbed

The following testbeds have analyzed IoT botnets such as Mirai by incorporating IoT communication protocols within the testbeds. Saez-de-Camara *et al.* [56] built on graphical network simulator-3 (GNS3) [57] for testing IoT network security. It leverages GNS3's functionalities for network emulation using VMs, containers [58], and real devices. The testbed, designed for assessing IoT network security, employs several key components. An IoT testbed orchestrator automates node creation, configuration, and link management to make the testbed setup process more efficient. A template creation engine produces reusable templates for network devices such as routers, IoT nodes, and switches. The templates are then used by the topology builder to automatically set up and arrange nodes, creating the appropriate network structure for the particular test scenario. The scenario generator controls the testing process by initiating nodes in a certain sequence, setting runtime parameters such as resource restrictions and network limits, and arranging specific attacks or behaviours in the test environment. This integrated technique enables effective and thorough testing of vulnerabilities in IoT network security [59]. The researchers have created a testbed that fulfils certain essential requirements such as security, scalability, reproducibility, self-configuration, and integrity. They ensured security by isolating the testbed from external internet access and internal network connections, creating a regulated and secure environment. Utilizing VM templates and virtualization technologies guaranteed scalability, enabling the testbed to handle different workloads and configurations easily. The testbed documentation was thorough, allowing for the replication of tests and configurations. Incorporating scripts to automate operations like node creation, configuration, and experiment execution enhanced the testbed's self-configuration skills. The fidelity was maintained by using real-world malware with minimum binary modifications, which increased the realism of experiments in the testbed. The testbed has a constraint in diversity as it mostly depends on IoT protocols [60], making it suitable mainly for IoT botnet simulations. This specialization may limit its relevance to a wider variety of botnet types, decreasing diversity in experimental circumstances.

Kumar and Lim [61] shift to a new IoT botnet emulation testbed that centers on IoT botnets. They use emulated Raspberry Pi devices that closely mimic real-world IoT devices in platform, operating system, applications, and networking capabilities. Network simulator (NS) [62] scripts enable the execution of instructions like commence, terminate, alter, and restart malware studies, removing the necessity for human setup. The testbed may be expanded to include additional devices and network components, and it is compatible with both physical and virtual devices. The testbed topology can also be altered to replicate other network conditions. It simplifies the testing of other types of Mirai variants, such as its advanced variant, by providing pre-configured additional infrastructure and the ability to run bot binaries and exploit codes. The testbed is secured with security measures by isolating it from external internet connections, creating a safe environment for experimentation. The scalability of the testbed is demonstrated by the ability to increase the number of QEMU VMs [63] per physical computer, allowing for incorporating more IoT devices as required. Users can easily manage and modify malware studies using NS scripts to simplify self-configuration. Comprehensive documentation of the testbed improves reproducibility, enabling researchers to reproduce tests correctly. Fidelity is preserved by using Mirai malware for testing, enhancing the authenticity of the test environment. The testbed's focus is restricted to evaluating variations of the Mirai malware, which limits its ability to handle a wider variety of malware kinds and behaviors.

Moreover, Gallopeni *et al.* [64] demonstrate a physical testbed for analyzing Mirai botnet behavior. The testbed utilizes a MikroTik router, six emulated Raspberry Pi devices, and two separate networks. Network traffic is captured and analyzed using Wireshark [65] for retrospective analysis and Pyshark (Python Library) for real-time inspection. The testbed allows researchers to trigger attack vectors on the emulated bots and analyze the traffic patterns for detection and mitigation purposes. The results demonstrate successful detection and interception of attack commands originating from specific IP addresses within the network [66]. The testbed's security is ensured through isolation from other lab resources, safeguarding the integrity of experiments. Utilizing the Mirai Botnet for testing enhances fidelity, providing a realistic representation of IoT botnet behavior. However, scalability is limited due to the reliance on Tinker Board devices for nodes, necessitating the addition of physical devices for expanding the testbed, thereby hindering scalability. Reproducibility is compromised by inadequate documentation, impeding the ability to replicate experiments accurately. Moreover, the testbed's focus solely on Mirai botnets restricts its diversity, limiting its applicability to broader malware research. Notably, the paper does not mention self-configuration aspects such as automation or scripted procedures, further detracting from the testbed's comprehensive functionality.

Beauchaine *et al.* [67] proposed a testbed consisting of two main network configurations: a home network and an enterprise network. Both configurations utilize various hardware components, including Raspberry Pi devices, routers, and a web server. The home network consists of Raspberry Pi devices, a gateway router, a web server, botnet server that is running on CentOS 7 with a bridged adapter for independent network device simulation. Vivid (Mirai variant) and UFONet (DDoS Service Toolkit) are the botnets deployed. The enterprise network consists of a wireless distribution system (WDS) [68] with four Linksys WRT54GL access points and a RADIUS server [69] with FreeRADIUS [70] on Raspberry Pi. The testbed demonstrates scalability by expanding across multiple network segments using virtualized devices. Furthermore, its comprehensive documentation enhances reproducibility, ensuring that experiments can be accurately replicated. The testbed's versatility is evident by its successful experimentation with various botnets, fulfilling the diversity criterion. Additionally, utilising a real-world botnet enhances fidelity, providing a realistic basis for research. However, the paper lacks discussion regarding the security measures or isolation of the testbed from the internet or the lab's internal network. Furthermore, there is no mention of scripts or automation to facilitate self-configuration, which may limit the testbed's efficiency and ease of use.

3.3. HTTP botnet emulation testbed

The following testbeds have analyzed HTTP botnets using machine learning and network manipulation methods. Dollah *et al.* [71] proposed a testbed that consists of five PCs infected with different bot types (Dorkbot [72], Zeus [17], Citadel [73], SpyEye [74], Cutwail [75]) and a sniffer server capturing network traffic. Normal traffic data is also collected for comparison. The collected data is used to train and validate a machine-learning model for botnet detection using 10-fold cross-validation [76]. The document lacks details about the specific classification algorithm used and the achieved performance metrics. Overall, the research presents a basic framework for HTTP botnet detection using traffic analysis, but further details and evaluation results are needed for a comprehensive assessment. The testbed exhibits fidelity and reproducibility criteria, attributed to using real-life botnets for experimentation and its documentation on the testbed construction procedures. However, several shortcomings are noted in other criteria. Security is compromised due to the presence of an internet-connected network router, posing a risk of malware escape. Scalability is limited as desktops were employed instead of virtualized resources, hindering the ease of scaling operations. Diversity is lacking as the experiments were confined to a specific type of bot, limiting the testbed's applicability to a broader range of scenarios. Moreover, the absence of scripts or automation impedes self-configuration, resulting in manual and potentially error-prone execution of experiments and node deployment processes.

Another study involving an HTTP botnet emulation testbed was conducted by Alomari *et al.* [77], where the researchers developed a testbed for network security research using a client-server model with remote access to a VM. This testbed allows researchers to create multiple scenarios, such as HTTP botnets launching DDoS attacks on target web servers. The testbed consists of 43 interconnected Windows workstations controlled by a C2 server and monitored by a dedicated workstation. Thirty workstations were installed with Windows XP service pack (SP) 2, 10 workstations with Windows 7, and one with SP3. The target web server is responsible for delivering the target website, and the monitoring workstation will monitor the traffic within the network. This setup allows researchers to generate malicious traffic to evaluate and validate their security systems. The testbed boasts security measures such as network isolation and the absence of internet connectivity, preventing malware from escaping. Scalability is ensured through the ability to spin up multiple VMs to accommodate additional nodes. Comprehensive documentation enables easy replication of the testbed. Real-world botnet experimentation enhances fidelity. However, the lack of automation and scripts hinders self-configuration for efficient node deployment. Additionally, focusing solely on HTTP botnets limits diversity in the testbed's applications.

3.4. Botnet attacks emulation testbed

The testbeds in the section conducted coordinated DDoS attacks on P2P grids and private clouds to simulate botnet attacks. Simon and Huraj [78] created middleware to assist in building P2P computational grids. The DDoS testbed architecture replicates a botnet structure by configuring worker nodes with OurGrid software, akin to installing bot software on compromised PCs. The worker nodes are installed by cloning VM templates and configured for connectivity. Perl scripts are generated to carry out commands on the workers and send job submissions. OurGrid job scripts oversee these tasks and administer the entire experiment. This method utilizes the commonalities between P2P grids and botnets to overcome the communication constraints of OurGrid. The testbed showcases security measures by functioning independently from the internet, minimizing the possibility of external threats and unauthorized entry. The system's scalability is demonstrated by the efficient process of adding new nodes through VM deployment, allowing for flexible expansion to meet different experimental requirements. Self-configuration capabilities are improved by having scripts to automate experiment running, increasing operational efficiency. The lack of real-world malware usage reduces the accuracy of the testbed as it may not faithfully reproduce the behaviours and traits of genuine threats. The limited variety of malware researched hinders the depth of insights that may be obtained from experiments, possibly neglecting crucial parts of cybersecurity research. Furthermore, the documentation of the testbed may be insufficient for replication, which can impede other researchers from reproducing tests and confirming results. Enhanced documentation will increase the transparency and dependability of research utilizing the testbed, aiding in the progress of cybersecurity knowledge and practices.

Al-Somaidai and Al-Hankawi [79] created a test environment to study the effects of SYN-Flood DDoS attacks on a private cloud setup. The testbed has five user PCs with the same setups, three data centre servers dedicated to various services (email, HTTP), and a network connected by UTP cables and a switch. The software design uses Oracle VM VirtualBox as the virtualization layer in the data centre, enabling adaptable resource allocation according to user-defined requirements. The experiment will simulate an SYN-Flood attack [80] on the email server using Opnet modeller [81] and compare the results with those from the physical testbed. The testbed has security mechanisms to prevent malware from spreading to real-world networks as it runs in an isolated environment. Thorough documentation simplifies replication, promoting transparency and allowing researchers to duplicate studies effectively. Scalability is difficult due to the constrained hardware specifications, which limit the number of nodes deployed in the testbed. Moreover, the experiments lack diversity and self-configuration, which mostly concentrate on specific DDOS attacks without automation or scripts for efficient experiment execution and node deployment. As a result, the testbed may not adequately address a broad range of cybersecurity scenarios or efficiently manage experiment workflows. Additionally, the fidelity of the testbed is uncertain, as there is no mention in the paper regarding the utilization of real-world malware, potentially limiting the relevance and applicability of research findings.

3.5. Required testbed features

In the ongoing fight against botnets, establishing a reliable testing environment is crucial. Over the past years, the research community has defined requirements for network testbeds specifically suited for emulating P2P botnets. These requirements, detailed in various research articles, ensure the accuracy and reliability of findings in botnet research. We have carefully reviewed these criteria and adapted them to our needs for P2P botnet emulation tests. These criteria serve as a critical benchmark to verify that the experiments are executed effectively and ensure that tests accurately replicate the behavior and communication patterns of real-world P2P botnets while maintaining safety and relevance within the controlled setting of the testbed environment.

a. Security: malware is created with malevolent intentions and could spread quickly. As a result, sufficient precautions should be taken before doing any malware-related research to guarantee that no malware is inadvertently released into the wild [82]. Isolating the experiment setting from the internet and other networks may be the only practical approach to effectively mitigate the danger posed by this hazard [83]. As a result, the emulation platform should be based on a secluded cluster housed in extremely secure premises. Robust physical barriers, monitoring systems, and a distinct access control system are all part of the labs' physical security measures [84]. The cluster is separated from other computer networks in terms of logical security.
b. Scalability: for the experiment's findings to be statistically significant, it is crucial to replicate a substantial number of elements like computers, subnets, processes, and other important entities [85]. The experiment seeks to closely simulate the scale and complexity of the real-world event being researched. The replication should include a big enough sample size to accurately represent the diversity and variability in the real environment [86]. A greater number of duplicated elements in an experiment leads to a more accurate reflection of real-world complexities and subtleties, improving the findings' reliability and generalizability.
c. Reproducibility: to allow a repeatable scenario, the following properties, defined at the node and topology level, should be present and documented [87]. Reproducibility regarding node setup is a description of each node's activity that includes information on all of the applications running there and how they are configured [83]. Reproducibility of the topology should include a thorough description of the network topology and each network link's features that connect each node [82].
d. Diversity: after the essential research has been done to clarify the structure and model of the botnet, the emulation platform should be able to replicate any botnet [88]. Flexibility is, therefore, a crucial prerequisite. The emulation platform should be simple to configure to accommodate different overlay network topologies [85]. For instance, mimicking the proportions seen in the real botnet should be possible by varying the ratio of private (unrouteable) IP addresses to public IP addresses [86].
e. Fidelity: the ability of a botnet emulation platform to replicate botnets that are functionally similar to those seen in the wild is a crucial prerequisite [87]. To do this, the bot binaries that are utilized to reproduce

the botnets must undergo very little if any, modification. Only those modifications that are required to get beyond the bot binaries' anti-virtualization and anti-debugging features should be made [83].

f. Self-configuration: self-configuration and automation play a crucial role in P2P botnet analysis testbeds as these functionalities streamline the deployment of testbed nodes, eliminating manual configuration and saving valuable time. Automation can also manage malware execution, initiating and terminating it based on testing requirements. Additionally, it allows for effortless test modifications during the analysis process. This efficiency is particularly important in P2P botnet research, where complex network interactions and numerous test scenarios are often involved. By reducing manual intervention and streamlining repetitive tasks, self-configuration and automation enable researchers to conduct comprehensive and efficient botnet analysis experiments. This requirement is introduced in this paper as we find that the time taken to configure the testbed to meet our research will be reduced drastically [89]. The overall efficiency of the testbed will be increased as minimal human intervention is needed to deploy various topologies.

3.6. Comparison of related work

To build on the discussion presented in the previous subsections, a comprehensive comparison of various testbeds is provided in Table 1. It outlines each testbed based on specific criteria, enabling a comparison to determine their suitability for achieving distinct research objectives within the cybersecurity domain. The testbeds offer diverse approaches to analyzing botnets, each with unique strengths and limitations. For example, recent testbeds like those from Saez-de-Camara *et al.* [56] and Kumar and Lim [61] are designed with scalability in mind, making them more effective for analyzing modern large-scale botnets such as P2P and IoT botnets. In contrast, older testbeds like those by Calvet *et al.* [41] and Basheer and Al-Hankawi [79] emphasize reproducibility, providing a solid foundation for conducting repeated experiments but may face challenges with scalability in today's complex threat landscape.

Table 1. Comparison with related work based on the required testbed

Ref.	Botnet analysed	Year	Security	Scalability	Reproducibility	Diversity	Self-configuration	Fidelity
[32]	P2P botnet	2021	●	○	◐	○	○	○
[41]		2010	●	●	○	◐	●	●
[56]	IoT botnet	2023	●	●	●	○	●	●
[61]		2019	●	●	●	○	●	●
[64]		2020	●	○	○	○	-	●
[67]		2021	-	●	●	●	○	●
[71]	HTTP botnet	2018	○	○	●	○	○	●
[77]		2016	●	●	●	○	○	●
[78]	Botnet attack	2016	●	●	○	○	●	○
[79]		2014	●	○	●	○	○	-

Security remains a priority across the majority of the testbeds, with efforts focusing on addressing the evolving security challenges posed by IoT botnets, as seen in Gallopeni *et al.* [64]. However, some earlier works, like Dollah *et al.* [71], exhibit limitations in scalability and security, indicating that while they provide valuable insights into HTTP botnets, they may not be suitable for addressing more diverse and large-scale botnet threats today. Additionally, certain testbeds emphasize fidelity and diversity, ensuring that the simulated environments closely mirror real-world scenarios, making them highly effective for practical botnet attack analysis.

Ultimately, no single testbed excels across all criteria, and the choice of testbed depends largely on the specific research focus. Testbeds like those by Saez-de-Camara *et al.* [56] offer robust scalability and fidelity, making them more adaptable for current botnet challenges, while older, well-established testbeds may still serve as valuable tools for reproducibility in specific types of botnet analysis. Researchers must balance these factors based on their experimental needs, whether prioritizing fidelity, security, or scalability.

4. OVERVIEW OF PROPOSED TESTBED SETUP AND DESIGN

Researching P2P botnets has always been tricky given their constant evolution. P2P botnets are notoriously difficult to track in real-world scenarios [90], and emulating them in a controlled environment presents additional hurdles, as some are equipped with anti-monitoring mechanisms. Their communication methods must stay active for defenders to capture any suspicious activity. To address these challenges, we propose a comprehensive methodology to secure the botnets within the testbed and deceive them into exhibiting malicious behaviors, thereby enabling successful monitoring by researchers. This section will be further subdivided into three categories, each delineating a distinct aspect of the proposed concept model. Subsequently, a comprehensive summary of the concept model and an elucidation of the requirements it addresses will be provided.

4.1. Malware profiling

Malware profiling is a crucial element of our conceptual model, aimed at identifying the specific ports used by malware for both server-side and client-side communication, as well as compiling a list of bootstrap IPs linked to the botnet. The results of this profiling process will be consolidated into a comprehensive configuration file. This finalized file will then act as the blueprint for bootstrapping the bots within an isolated network in the later stages of the project.

Port forwarding [91] serves as a critical mechanism for facilitating communication between different components within the testbed environment. For instance, if the malware opens a specific port for listening, this port is mapped to a designated forward port. Port forwarding enables the redirection of network traffic from one port to another, thereby establishing connectivity between specified endpoints. The process of port forwarding begins with the identification of open ports on the system where a list of all currently available ports is listed. Once a listening port associated with the malware binary is detected, port forwarding is initiated. A mapping between a specified forward port and the target port opened by the malware is carried out. Consequently, any incoming connection made to the forward port is redirected to the target port, enabling seamless communication with the malware-infected system.

4.2. Network configuration

The network architecture is designed with a distributed layout, where two servers are connected to each of the two routers, totaling four servers interconnected in pairs which is illustrated in Figure 4. The routers serve a dual purpose by facilitating the connection of servers and acting as gateways for botnets. Configured with OSPFv2 (IPv4), the routers enable effective communication between servers connected to different routers through the OSPF protocol, ensuring seamless interaction among nodes in various subnets. The router within the testbed play a vital role in deceiving the botnets into believing they are in a real internet network by allowing communication to any devices with any public IP address. A key aspect of the network configuration is the use of the OSPF interface as the exclusive link accessible to the infected machines, effectively deceiving the botnets into believing they are operating within a real internet network.

Additionally, each server is assigned a unique set of IP addresses, ensuring full coverage of IPv4 addresses within the testbed. This distribution ensures that the testbed includes a broad spectrum of IP addresses, allowing for comprehensive monitoring and analysis of future communication attempts by botnets. The proportion of public IPv4 and private IPv4 addresses can be adjusted to replicate real botnets, as various types of P2P botnets necessitate different proportions of these addresses. The sophisticated network design creates a secure and regulated setting for experimentation and study, providing both connectivity and isolation to fulfill the defined security needs.

Virtualization technology [92] is fundamental for managing the complex network of servers and VMs that make up the P2P botnet emulation testbed. We create a flexible and expandable platform that is suitable for efficiently setting up, configuring, and managing VMs that represent different aspects of the botnet network. Virtualization technology is crucial for coordinating the deployment and administration of VM within the P2P botnet emulation testbed. One important part of this orchestration process is assigning a distinct block of IP addresses to each server in the virtualized setup. Every server can accommodate a variety of VMs, each designed for a specific function or role in the botnet emulation architecture. This method of assigning IP addresses in modules makes the setup process more efficient and eases the control of network resources, enhancing the overall efficiency and effectiveness of the test environment. To further enhance security within the testbed, isolated bridges are established within the virtualized environment. By constructing many bridges, we can effectively isolate botnets and other dangerous elements in separate network segments to prevent them from causing harm to other parts of our system.

Snapshots are vital for controlling and maintaining the state of VMs within our testbed architecture. A snapshot captures the VM's state at a specific point in time, including disk contents, RAM, and configuration settings. By taking snapshots at various stages of setup and use, we can create checkpoints that allow for easy restoration to previous states if necessary. This capability enables us to experiment with different configurations, test software installations, and conduct experiments without risking permanent changes to the underlying architecture.

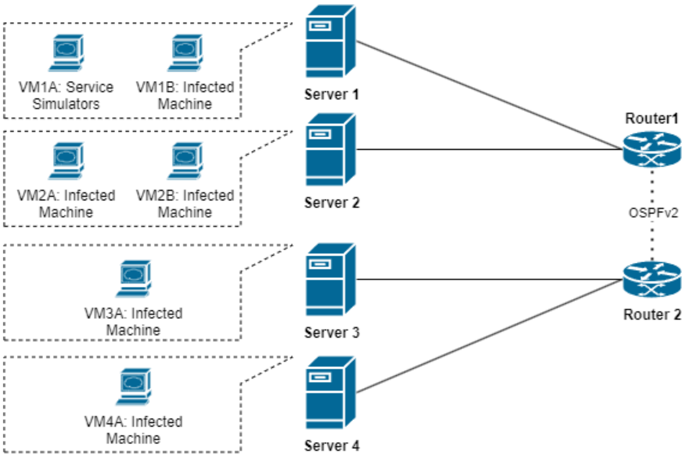


Figure 4. Schematic of hardware configuration in the testbed

It is crucial to duplicate the necessary internet-based services that malware depends on for its functioning [56]. These services, including DNS, HTTP, and file transfer protocol (FTP), are frequently used by malware for communication and data sharing [93]. Utilizing this architecture allows us to generate virtual environments that closely replicate the actions and reactions of actual services, offering an authentic setting for simulating botnet communication. The bots will be deployed in the testbed environment after emulating these services to construct their botnet communication overlay.

In anticipation of the testbed expansion, we plan to implement a scalable network architecture that can accommodate the full range of public IP addresses. This expanded setup will include a greater number of testbeds, each equipped with its own servers and routers, mirroring the current distributed configuration illustrated in Figure 5. By increasing the number of routers and testbeds, we aim to better simulate a wide variety of internet scenarios. The routers, configured with OSPFv2 (IPv4), will continue to play a critical role in connecting servers and serving as gateways for botnets. This scalability allows for more comprehensive emulation of internet-like interactions, providing a secure, controlled environment for researchers to explore and analyze diverse scenarios. The refined architecture ensures that the testbed remains adaptable and scalable, meeting the evolving needs of research and experimentation while maintaining a balance between connectivity and isolation for security.

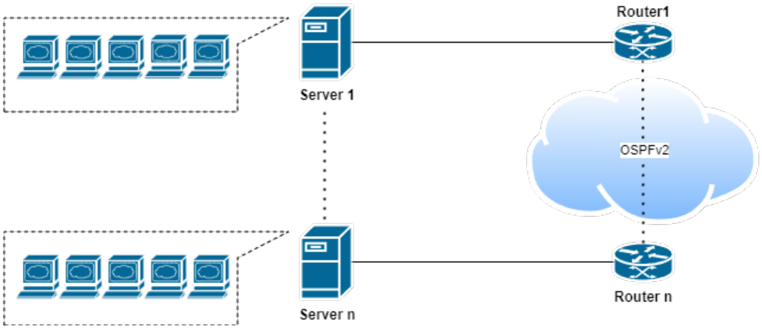


Figure 5. Expected schematic of hardware configuration in the testbed

4.3. Infrastructure and configuration management

Infrastructure and configuration management [94] are critical for maintaining the efficient functioning of the P2P botnet emulation testbed. Infrastructure management encompasses activities like provisioning, installing, and managing the VM that makes up the testbed environment. This involves choosing suitable operating systems, allocating hardware resources, and configuring networking settings to efficiently support the emulation operations. A structured approach to configuration management [95] is employed to establish and oversee the configurations, parameters, and interdependencies of the various components within the test environment. By employing configuration management, we may offer centralized control for all the computers (nodes) in our network, supervised by a single master node.

Our infrastructure management approach utilizes structured configuration files to define key parameters for VM creation, such as authentication credentials and resource requirements. By integrating these files into our development workflow, we streamline the process of creating nodes in the virtualized environment discussed earlier. This strategy prioritizes scalability, security, and efficiency, ensuring the seamless coordination of resources within the testbed environment. By simplifying complexities and automating repetitive tasks, we aim to increase productivity and enable rapid testing and iteration in research and development.

Configuration management plays a critical role in the deployment and maintenance of our testbed system. We adopt a systematic approach to automate the management of system configurations across multiple nodes, with a particular focus on infected workstations. A configuration management system is used to centrally control configuration settings, software installations, and botnet binary execution within the VMs. To simplify deployment, we create standardized templates for node setup, which include the necessary components and settings. These configuration files define tasks such as software distribution and execution, ensuring smooth and consistent propagation of configuration changes across the entire infrastructure.

5. EXPECTED OUTCOMES

This section provides a detailed overview of the anticipated results that would arise from executing the suggested P2P botnet emulation testbed. The outputs cover different aspects, such as evaluating functionality and anticipating the consequences of the proposed P2P botnet emulation testbed and its research findings on the broader botnet research community. Each component offers valuable information on how the testbed can help progress botnet research, improve comprehension of P2P botnets, and strengthen the creation of efficient countermeasures. The next sections will describe of these components.

5.1. Functionality evaluation

5.1.1. Rapid deployment of multiple nodes

The testbed's automated deployment features are anticipated to simplify and speed up the process of adding many nodes. Automation allows for the rapid creation of nodes with little need for manual involvement. This feature improves the effectiveness of experiments and simulations by enabling researchers to quickly expand the testbed and carry out experiments with various situations and setups. This scalability improves the testbed's flexibility and allows for the study of intricate P2P botnet scenarios with a large number of nodes.

5.1.2. Communication overlay of peer-to-peer botnets

To test the capabilities of Malbed, a real sample of the Sality botnet was retrieved from an online repository. The hash value of the botnet is d1471ad5eb84ea711f65f5f579aaf55aa5bec35d126e6158ea824e754fabb0a6. Nodes were deployed based on the report received from a sandbox, ensuring that the botnet would be bootstrapped within the testbed environment. Throughout the experiment, we gathered and analyzed packet capture (pcap) files from each node as shown in Figure 6. The analysis revealed that each node made multiple attempts to connect to the internet, confirming the botnet's efforts to establish external communication channels.

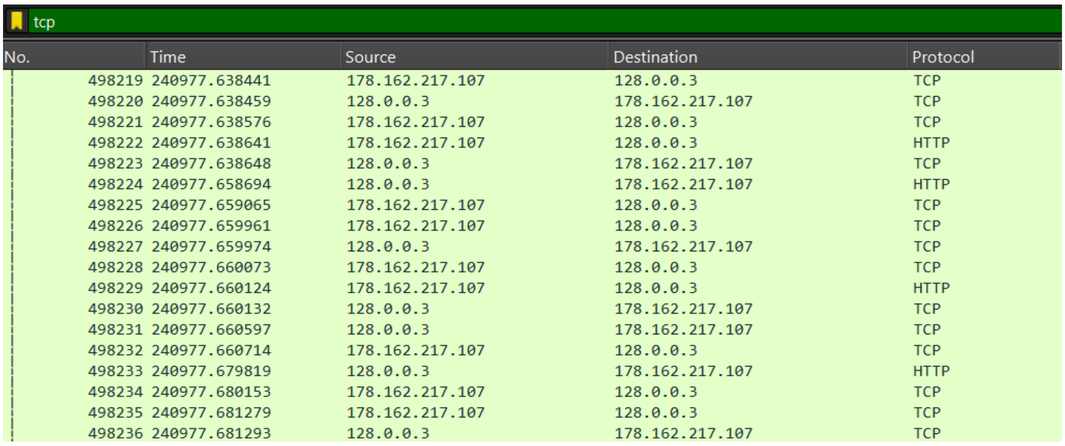
5.2. Impact evaluation

5.2.1. Insightful data for security research and innovation

The experimentation undertaken within the testbed is expected to yield a lot of valuable data about P2P botnet activity. This data, comprising network traffic patterns, attack vectors, and evasion strategies, will serve as a significant resource for security experts. The availability of such data stimulates innovation in developing sophisticated IDS, threat intelligence, and proactive security measures.

5.2.2. Enchanced understanding of peer-to-peer botnet dynamics

Through rigorous experimentation and analysis facilitated by the testbed, researchers intend to obtain insights into the subtleties of P2P botnet behaviors, communication patterns, and evasion tactics. This greater understanding is crucial for generating more strong countermeasures and staying ahead of evolving threats. By duplicating realistic scenarios within the emulation platform, researchers may analyze the effectiveness of existing countermeasures and investigate creative techniques for minimizing the impact of P2P botnet operations. Researchers working in the initiative intend to share their discoveries through scholarly papers, offering fresh ideas, approaches, and solutions to the broader academic and professional populations. This contribution to the corpus of knowledge will further advance the collective understanding of P2P botnets and bolster scholarly discourse on emergent cybersecurity concerns.



```
tcp
No.      Time                Source            Destination       Protocol
  498219 240977.638441 178.162.217.107   128.0.0.3         TCP
  498220 240977.638459 128.0.0.3         178.162.217.107   TCP
  498221 240977.638576 178.162.217.107   128.0.0.3         TCP
  498222 240977.638641 178.162.217.107   128.0.0.3         HTTP
  498223 240977.638648 128.0.0.3         178.162.217.107   TCP
  498224 240977.658694 128.0.0.3         178.162.217.107   HTTP
  498225 240977.659065 178.162.217.107   128.0.0.3         TCP
  498226 240977.659961 178.162.217.107   128.0.0.3         TCP
  498227 240977.659974 128.0.0.3         178.162.217.107   TCP
  498228 240977.660073 178.162.217.107   128.0.0.3         TCP
  498229 240977.660124 178.162.217.107   128.0.0.3         HTTP
  498230 240977.660132 128.0.0.3         178.162.217.107   TCP
  498231 240977.660597 128.0.0.3         178.162.217.107   TCP
  498232 240977.660714 178.162.217.107   128.0.0.3         TCP
  498233 240977.679819 128.0.0.3         178.162.217.107   HTTP
  498234 240977.680153 178.162.217.107   128.0.0.3         TCP
  498235 240977.681279 178.162.217.107   128.0.0.3         TCP
  498236 240977.681293 128.0.0.3         178.162.217.107   TCP
```

Figure 6. Example of botnet network traffic

6. DISCUSSION

In this section, we look into the critical significance of an emulation testbed, specifically in the context of P2P botnet emulation, emphasizing the paucity of dedicated P2P botnet analysis testbeds and the limitations of existing solutions. The lack of particular testbeds addressing P2P botnets spurred the inclusion of varied testbeds built for IoT and HTTP botnet analysis in the discussion. Existing testbeds for P2P botnet analysis, such as BotSideP2P [32] and the 3000-node botnet [41], offer valuable insights but fall short of fulfilling the unique requirements of P2P botnet emulation.

The proposed concept model optimizes existing solutions by utilizing automation and scripts to efficiently deploy nodes in the testbed. Notably, the BotSideP2P testbed [32], refactored to use the Asyncio framework, exemplifies the benefits of such enhancements in terms of stability and usability which is illustrated in Figure 7. The planned testbed incorporates virtualization, infrastructure management, and configuration management to demonstrate a methodology that guarantees scalability, security, and reproducibility. Section 3 discussed numerous testbeds that emphasized various sorts of botnets, such as IoT and HTTP botnets. Although general testbeds are valuable for studying botnet activities, the unique characteristics of P2P botnets require a specialized testbed for in-depth examination. The talk introduces the proposed concept model, emphasizing the necessity of a safe, scalable, and automated emulation platform designed specifically for the unique features of P2P botnets.

The concept model details the hardware components of the testbed, highlighting the distributed topology and the dual function of routers in enabling server connections and acting as gateways for botnets. The architecture guarantees both connectivity and isolation, addressing security issues while establishing a regulated environment for experimentation. Utilizing virtualization software, infrastructure management tools, and configuration management tools guarantees automation, fulfilling the need for self-configuration and efficiency in P2P botnet analysis testbeds.

Our proposed conceptual model provides a comprehensive solution tailored to meet the requirements of a successful P2P botnet emulation testbed. By making minimal modifications to bot binaries, fidelity is preserved, ensuring that bots behave similarly to those in real-world scenarios. This high fidelity allows the emulation platform to accurately replicate botnets found in the wild, facilitating detailed and realistic research. Furthermore, self-configuration and automation are integral components of the testbed. This approach significantly optimizes node deployment, minimizing manual intervention and enhancing efficiency. The automation feature not only accelerates the setup process but also allows researchers to control malware execution, enabling the initiation and termination of trials as required. Such flexibility is vital in P2P botnet research, given the complexity of managing network connections and varied test scenarios.

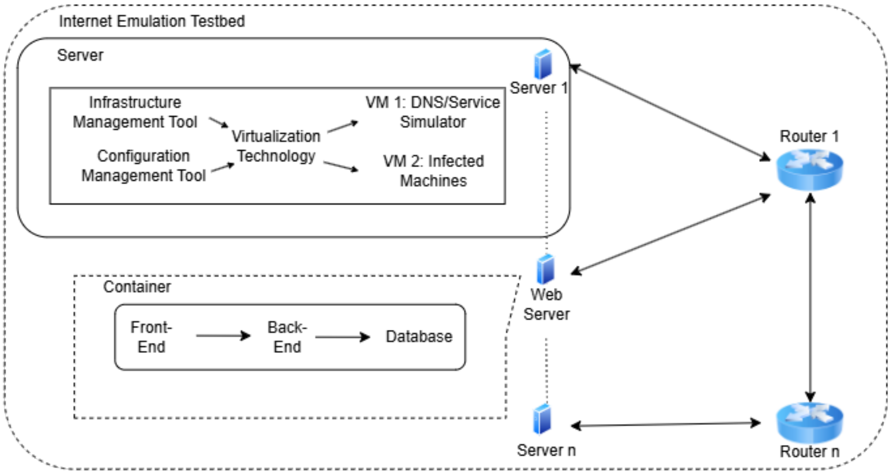

Internet Emulation Testbed
Server
Infrastructure Management Tool
Configuration Management Tool
Virtualization Technology
VM 1: DNS/Service Simulator
VM 2: Infected Machines
Server 1
Router 1
Container
Front-End
Back-End
Database
Web Server
Server n
Router n

Figure 7. Expected architecture diagram

The testbed's design also emphasizes diversity, enabling the emulation of a wide range of P2P botnets by accommodating different overlay network topologies. Its modular design allows researchers to mirror the structures observed in real botnets, offering flexibility to adapt to various conditions. Reproducibility is ensured through meticulous logging of node activities and network configurations at both the node and topology levels, establishing a framework for consistent and repeatable experiments. Scalability is another key feature of the testbed, designed to support an expanded network that can accommodate all public IP addresses. By adjusting the number of routers and servers, our model increases the capacity to simulate a broader range of internet scenarios, thus creating a diverse and robust research environment. Finally, the inclusion of enhanced security measures such as physical barriers, monitoring systems, and specific access controls underscores our commitment to maintaining a secure and controlled environment for experimentation, addressing the security requirements critical to P2P botnet analysis.

7. CONCLUSION

This research provides a significant contribution to the field of cybersecurity by addressing the challenges of accurately emulating P2P botnets within controlled environments. Through the development of a scalable and adaptable testbed, we have demonstrated how advanced profiling techniques, network architecture, and configuration management can be used to overcome limitations in previous botnet emulation approaches. The ability to dynamically scale the testbed without relying on reverse engineering offers a new level of flexibility, allowing researchers to replicate real-world botnet behaviors more effectively. The implications of this research extend beyond the academic sphere. By providing a robust platform for emulating botnets, our framework facilitates the development and testing of more effective botnet detection and mitigation strategies, which are crucial in addressing the evolving threats posed by cyberattacks. Furthermore, the methodologies outlined here offer a foundation for future studies to build upon, enabling ongoing advancements in P2P botnet research.

Looking forward, future research could focus on enhancing the scalability of the testbed for more complex botnet behaviors, as well as exploring additional techniques for botnet detection in decentralized networks. The flexibility of our testbed ensures it can be continuously adapted to meet the growing sophistication of botnets, further supporting efforts to safeguard against the persistent threat they pose to critical infrastructures and online services.

FUNDING INFORMATION

This work is funded partly by Universiti Sains Malaysia's Short-Term Grant [304/PNAV/6315576] and Community-Based Research 2 (CBR-2) project Grant [R504LRGAL00800000001410000/K164], supported by the Asian Internet Interconnection Initiatives and the School on Internet (AI3/SOI).

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Mithiiran Parthipan	✓	✓	✓	✓	✓	✓		✓	✓	✓			✓	
Shams Ul Arfeen Laghari		✓				✓		✓	✓	✓	✓	✓		
Ashish Jaisan	✓		✓	✓		✓			✓		✓		✓	
Amber Baig			✓		✓		✓			✓				
Muhammad Asim Ali					✓		✓			✓		✓		✓
Shankar Karuppayah					✓		✓			✓		✓		✓

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal Analysis

I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting

Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject Administration
Fu : **Fu**nding Acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

The supporting data of this study are openly available on Github at https://github.com/Mithiiran05/Sality_pcap.git.

REFERENCES

[1] Y. Li and Q. Liu, "A comprehensive review study of cyber-attacks and cyber security; Emerging trends and recent developments," *Energy Reports*, vol. 7, pp. 8176–8186, 2021, doi: 10.1016/j.egyr.2021.08.126.

[2] S. Khattak, N. R. Ramay, K. R. Khan, A. A. Syed and S. A. Khayam, "A Taxonomy of Botnet Behavior, Detection, and Defense," in *IEEE Communications Surveys and Tutorials*, vol. 16, no. 2, pp. 898-924, 2014, doi: 10.1109/SURV.2013.091213.00134.

[3] L. Bilge, D. Balzarotti, W. Robertson, E. Kirda, and C. Kruegel, "DISCLOSURE: Detecting Botnet Command and Control Servers Through Large-Scale NetFlow Analysis," in *ACSAC '12: Proceedings of the 28th Annual Computer Security Applications Conferenc*, 2012, pp. 129–138, doi: 10.1145/2420950.2420969.

[4] S. Karuppayah, M. Fischer, C. Rossow, and M. Muhlhauser, "On advanced monitoring in resilient and unstructured P2P botnets," *2014 IEEE International Conference on Communications (ICC)*, Sydney, NSW, Australia, 2014, pp. 871–877, doi: 10.1109/ICC.2014.6883429.

[5] G. Stringhini, O. Hohlfeld, C. Kruegel, and G. Vigna, "The harvester, the botmaster, and the spammer," in *ASIA CCS '14: Proceedings of the 9th ACM Symposium on Information, Computer and Communications Security*, 2014, pp. 353–364, doi: 10.1145/2590296.2590302.

[6] E. Simkhada, E. Shrestha, S. Pandit, U. Sherchand, and A. M. Dissanayaka, "Security Threats/Attacks via Botnets and Botnet Detection and Prevention Techniques In Computer Networks: A Review," *Midwest Instruction and Computing Symposium (MICS 2019)*, vol. 52, pp. 1–15, 2019.

[7] S. G. Abbas, S. Zahid, F. Hussain, G. A. Shah, and M. Husnain, "A Threat Modelling Approach to Analyze and Mitigate Botnet Attacks in Smart Home Use Case," *2020 IEEE 14th International Conference on Big Data Science and Engineering (BigDataSE)*, pp. 122–129, 2020, doi: 10.1109/BigDataSE50710.2020.00024.

[8] I. Stellios, P. Kotzanikolaou, M. Psarakis, C. Alcaraz, and J. Lopez, "A survey of iot-enabled cyberattacks: Assessing attack paths to critical infrastructures and services," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 3453–3495, 2018, doi: 10.1109/COMST.2018.2855563.

[9] P. S. Gosavi, A. A. Dange, and B. B. Meshram, "Critical Infrastructure and Botnet," *International Journal of Engineering Research and Applications (IJERA)*, vol. 3, no. 2, pp. 1695–1701, 2013.

[10] B. Kerbs, "KrebsOnSecurity Hit With Record DDoS – Krebs on Security," Krebs On Security. [Online]. Available: https://krebsonsecurity.com/2016/09/krebsonsecurity-hit-with-record-ddos/. (Date accessed: Mar. 06, 2024).

[11] B. Kerbs, "Did the Mirai Botnet Really Take Liberia Offline?," Krebs on Security. [Online]. Available: https://krebsonsecurity.com/2016/11/did-the-mirai-botnet-really-take-liberia-offline/. (Date accessed: Mar. 06, 2024).

[12] T. Cruz *et al.*, "CWMP extensions for enhanced management of domestic network services," *IEEE Local Computer Network Conference*, Denver, CO, USA, 2010, pp. 180–183, doi: 10.1109/LCN.2010.5735695.

[13] T.-H. Nguyen, T. Nguyen, and M. Yoo, "Analysis of deployment approaches for virtual customer premises equipment," in *2018 International Conference on Information Networking (ICOIN)*, Chiang Mai, Thailand, 2018, pp. 289–291, doi: 10.1109/ICOIN.2018.8343127.

[14] M. Antonakakis *et al.*, "Understanding the mirai botnet," in *Proceedings of the 26th USENIX Security Symposium*, 2017, pp. 1093–1110.

[15] F. Haddadi, D. Runkel, A. N. Zincir-Heywood, and M. I. Heywood, "On botnet behaviour analysis using GP and C4.5," in *GECCO Comp '14: Proceedings of the Companion Publication of the 2014 Annual Conference on Genetic and Evolutionary Computation*, 2014, pp. 1253–1260, doi: 10.1145/2598394.2605435.

[16] S. Soltani, S. Amin, H. Seno, M. Nezhadkamali, and R. Budirato, "A survey on real world botnets and detection mechanisms," *International Journal of Information & Network Security (IJINS)*, vol. 3, no. 2, pp. 116–127, 2014.

[17] D. Andriesse, C. Rossow, B. Stone-Gross, D. Plohmann, and H. Bos, "Highly resilient peer-to-peer botnets are here: An analysis of Gameover Zeus," in *2013 8th International Conference on Malicious and Unwanted Software: "The Americas" (MALWARE*, Fajardo, PR, USA, 2013, pp. 116–123, doi: 10.1109/MALWARE.2013.6703693.

[18] A. Bakshtein, "Hospitals Hit by DDoS Attacks as Killnet Group Targets the Healthcare Sector - What You Need to do Now — Imperva," Imperva. [Online]. Available: https://www.imperva.com/blog/hospitals-hit-by-ddos-attacks/. (Date accessed: Mar. 06, 2024).

[19] T. Krause, R. Ernst, B. Klaer, I. Hacker, and M. Henze, "Cybersecurity in power grids: Challenges and opportunities," *Sensors*, vol. 21, no. 18, pp. 1–19, 2021, doi: 10.3390/s21186225.

[20] A. Mascellino, "Researchers Uncover Major Surge in Global Botnet Activity - Infosecurity Magazine," Infosecurity Magazine. [Online]. Available: https://www.infosecurity-magazine.com/news/hundredfold-surge-global-botnet/. (Date accessed: Mar. 06, 2024).

[21] K. B. Wellington, "Cyberattacks on Medical Devices and Hospital Networks: Legal Gaps and Regulatory Solutions, 30 Santa Clara High Tech," *The Santa Clara High Technology Law Journal*, vol. 30, no. 2, pp. 139-198, 2014.

[22] M. Yin, X. Chen, Q. Wang, W. Wang, and Y. Wang, "Dynamics on Hybrid Complex Network: Botnet Modeling and Analysis of Medical IoT," *Security and Communication Networks*, vol. 2019, pp. 1–14, Aug. 2019, doi: 10.1155/2019/6803801.

[23] F. A. G. Muzzi, P. R. D. M. Cardoso, D. F. Pigatto, and K. R. L. J. C. Branco, "Using Botnets to provide security for safety critical embedded systems - A case study focused on UAVs," *Journal of Physics: Conference Series, 4th International Conference on Mathematical Modeling in Physical Sciences (IC-MSquare2015)*, Mykonos, Greece, 2015, vol. 633, no. 1, doi: 10.1088/1742-6596/633/1/012053.

[24] S. Karuppayah, *Advanced Monitoring in P2P Botnets: A Dual Perspective*, Springer, 2018, doi: 10.1007/978-981-10-9050-9.

[25] H. R. Zeidanloo and A. A. Manaf, "Botnet command and control mechanisms," *2009 Second International Conference on Computer and Electrical Engineering*, vol. 1, pp. 564–568, 2009, doi: 10.1109/ICCEE.2009.151.

[26] P. Amini, R. Azmi, and M. A. Araghizadeh, "Analysis of Network Traffic Flows for Centralized Botnet Detection," *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, vol. 11, no. 2, pp. 7–17, 2019.

[27] Y. Xing, H. Shu, H. Zhao, D. Li, and L. Guo, "Survey on Botnet Detection Techniques: Classification, Methods, and Evaluation," *Mathematical Problems in Engineering*, vol. 2021, no. 1, 2021, doi: 10.1155/2021/6640499.

[28] Y. Zhang and S. B. Venkatakrishnan, "Kadabra: Adapting Kademlia for the Decentralized Web," *arXiv*, 2022, doi: 10.48550/arXiv.2210.12858.

[29] S. Herwig, K. Harvey, G. Hughey, R. Roberts, and D. Levin, "Measurement and Analysis of Hajime, a Peer-to-peer IoT Botnet," *Network and Distributed Systems Security (NDSS) Symposium 2019*, 2019, pp. 1-15, doi: 10.14722/ndss.2019.23488.

[30] A. J. Ganesh, A.-M. Kermarrec, and L. Massoulie, "Peer-to-peer membership management for gossip-based protocols," *IEEE Transactions on Computers*, vol. 52, no. 2, pp. 139–149, Feb. 2003, doi: 10.1109/TC.2003.1176982.

[31] T. Holz, M. Steiner, F. Dahl, E. Biersack, and F. Frederic, "Measurements and Mitigation of Peer-to-Peer-based Botnets: A Case Study on Storm Worm," *Leet*, vol. 8, pp. 1–9, 2008.

[32] A. Beauchaine, O. Collins, and M. Yun, "BotsideP2P: A Peer-to-Peer Botnet Testbed," in *2021 IEEE 12th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, New York, NY, USA, 2021, pp. 0236–0242, doi: 10.1109/UEMCON53757.2021.9666641.

[33] G. Vormayr, T. Zseby, and J. Fabini, "Botnet Communication Patterns," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2768–2796, 2017, doi: 10.1109/COMST.2017.2749442.

[34] R. S. Rawat, M. Diwakar, and P. Verma, "ZeroAccess botnet investigation and analysis," *International Journal of Information Technology*, vol. 13, no. 5, pp. 2091–2099, Oct. 2021, doi: 10.1007/s41870-021-00693-z.

[35] L. Böck, E. Vasilomanolakis, M. Mühlhäuser, and S. Karuppayah, "Next generation P2P botnets: Monitoring under adverse conditions," *Research in Attacks, Intrusions, and Defenses: 21st International Symposium, RAID 2018*, vol. 11050, pp. 511–531, 2018, doi: 10.1007/978-3-030-00470-5-24.

[36] A. Shafee, "Botnets and their detection techniques," in *2020 International Symposium on Networks, Computers and Communications (ISNCC)*, Montreal, QC, Canada, pp. 1–6, Oct. 2020, doi: 10.1109/ISNCC49221.2020.9297307.

[37] O. Pomorova, O. Savenko, S. Lysenko, A. Kryshchuk, and K. Bobrovnikova, "Anti-evasion Technique for the Botnets Detection Based on the Passive DNS Monitoring and Active DNS Probing," in *Computer Networks: 23rd International Conference, CN 2016*, vol. 608, pp. 83–95, 2016, doi: 10.1007/978-3-319-39207-3-8.

[38] J. Domingo-Pascual, Y. Shavitt, and S. Uhlig, "Traffic Monitoring and Analysis," *Third International Workshop, TMA 2011*, 2011, doi: 10.1007/978-3-642-20305-3.

[39] P. Wainwright and H. Kettani, "An analysis of botnet models," *ICCDA '19: Proceedings of the 2019 3rd International Conference on Compute and Data Analysis*, pp. 116–121, 2019, doi: 10.1145/3314545.3314562.

[40] P. Wang, L. Wu, B. Aslam, and C. C. Zou, "Analysis of peer-to-peer botnet attacks and defenses," *Propagation Phenomena in Real World Networks*, vol. 85, pp. 183–214, 2015, doi: 10.1007/978-3-319-15916-4-8.

[41] J. Calvet *et al.*, "The case for in-the-lab botnet experimentation: Creating and taking down a 3000-node botnet," *ACSAC '10: Proceedings of the 26th Annual Computer Security Applications Conference*, 2010, pp. 141–150, doi: 10.1145/1920261.1920284.

[42] M. Stevanovic, K. Revsbech, J. M. Pedersen, R. Sharp, and C. D. Jensen, “A Collaborative Approach to Botnet Protection,” in *Multidisciplinary Research and Practice for Information Systems: IFIP WG 8.4, 8.9/TC 5 International Cross-Domain Conference and Workshop on Availability, Reliability, and Security, CD-ARES 2012)*, vol. 7465, pp. 624–638, 2012, doi: 10.1007/978-3-642-32498-7_47.

[43] S. Dange and M. Chatterjee, “IoT Botnet: The Largest Threat to the IoT Network,” in *Data Communication and Networks: Proceedings of GUCON 2019*, vol. 1049, pp. 137–157, 2020, doi: 10.1007/978-981-15-0132-6_10.

[44] D. Acarali, M. Rajarajan, N. Komninos, and I. Herwono, “Survey of approaches and features for the identification of HTTP-based botnet traffic,” *Journal of Network and Computer Applications*, vol. 76, pp. 1–15, Dec. 2016, doi: 10.1016/j.jnca.2016.10.007.

[45] C. P. Lee, “Framework for botnet emulation and analysis,” Ph.D. dissertation, School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, Georgia, 2009.

[46] P. Maymounkov and D. Mazières, “Kademlia: A Peer-to-Peer Information System Based on the XOR Metric,” in *International Workshop on Peer-To-Peer Systems*, vol. 2429, pp. 53–65, 2002, doi: 10.1007/3-540-45748-8_5.

[47] C. Hattingh, *Using Asyncio in Python: understanding Python’s asynchronous programming features*, O’Reilly Media, Inc., 2020.

[48] S. Kenlon, “Getting Started with the Raspberry Pi,” in *Developing Games on the Raspberry Pi*, Berkeley, CA: Apress, 2019, pp. 1–18, doi: 10.1007/978-1-4842-4170-7_1.

[49] G. Sinclair, C. Nunnery, and B. B. Kang, “The waledac protocol: The how and why,” in *2009 4th International Conference on Malicious and Unwanted Software (MALWARE)*, Montreal, QC, Canada, 2009, pp. 69–77, doi: 10.1109/MALWARE.2009.5403015.

[50] K. Wang, J. Rao, and C.-Z. Xu, “Rethink the virtual machine template,” *ACM SIGPLAN Notices*, vol. 46, no. 7, pp. 39–50, Jul. 2011, doi: 10.1145/2007477.1952690.

[51] L. Cao and X. Qiu, “Defence against Botnets: A Formal Definition and a General Framework,” in *2013 IEEE Eighth International Conference on Networking, Architecture and Storage*, Xi’an, China, 2013, pp. 237–241, doi: 10.1109/NAS.2013.37.

[52] C. Mou, “DNS is the Internet Pivotal Basics and Fundamental,” *International Journal of Advanced Network, Monitoring and Controls*, vol. 7, no. 2, pp. 11–23, Jan. 2022, doi: 10.2478/ijanmc-2022-0012.

[53] R. Sureswaran, H. Al Bazar, O. Abouabdalla, A. M. Manasrah, and H. El-Taj, “Active e-mail system SMTP protocol monitoring algorithm,” in *2009 2nd IEEE International Conference on Broadband Network & Multimedia Technology*, Beijing, China, 2009, pp. 257–260, doi: 10.1109/ICBNMT.2009.5348490.

[54] O. S. Younes, “A Secure DHCP Protocol to Mitigate LAN Attacks,” *Journal of Computer and Communications*, vol. 4, no. 1, pp. 39–50, 2016, doi: 10.4236/jcc.2016.41005.

[55] B. Jung, T. Kim, and E. G. Im, “Malware classification using byte sequence information,” *RACS ’18: Proceedings of the 2018 Conference on Research in Adaptive and Convergent Systems*, 2018, pp. 143–148, doi: 10.1145/3264746.3264775.

[56] X. Saez-de-Camara, J. L. Flores, C. Arellano, A. Urbieta, and U. Zurutuza, “Gotham Testbed: A Reproducible IoT Testbed for Security Experiments and Dataset Generation,” *IEEE Transactions on Dependable and Secure Computing*, vol. 21, no. 1, pp. 186–203, 2023, doi: 10.1109/TDSC.2023.3247166.

[57] L. Golightly, P. Modesti, and V. Chang, “Deploying Secure Distributed Systems: Comparative Analysis of GNS3 and SEED Internet Emulator,” *Journal of Cybersecurity and Privacy*, vol. 3, no. 3, pp. 464–492, Aug. 2023, doi: 10.3390/jcp3030024.

[58] P. Sharma, L. Chaufournier, P. Shenoy, and Y. C. Tay, “Containers and Virtual Machines at Scale,” in *Middleware ’16: Proceedings of the 17th International Middleware Conference*, Nov. 2016, pp. 1–13, doi: 10.1145/2988336.2988337.

[59] D. Nagasundaram, S. Manickam, S. U. A. Laghari, and S. Karuppayah, “Proposed fog computing-enabled conceptual model for semantic interoperability in internet of things,” *Bulletin of Electrical Engineering and Informatics*, vol. 13, no. 2, pp. 1183–1196, Apr. 2024, doi: 10.11591/eei.v13i2.5748.

[60] A. J. Hintaw, S. Manickam, S. Karuppayah, M. A. Aladaileh, M. F. Aboalmaaly, and S. U. A. Laghari, “A Robust Security Scheme Based on Enhanced Symmetric Algorithm for MQTT in the Internet of Things,” *IEEE Access*, vol. 11, pp. 43019–43040, Mar. 2023, doi: 10.1109/ACCESS.2023.3267718.

[61] A. Kumar and T. J. Lim, “A Secure Contained Testbed for Analyzing IoT Botnets,” *estbeds and Research Infrastructures for the Development of Networks and Communities: 13th EAI International Conference*, vol. 270, 2019, doi: 10.1007/978-3-030-12971-2_8.

[62] L. Campanile, M. Gribaudo, M. Iacono, F. Marulli, and M. Mastroianni, “Computer Network Simulation with ns-3: A Systematic Literature Review,” *Electronics*, vol. 9, no. 2, pp. 1-25, Feb. 2020, doi: 10.3390/electronics9020272.

[63] G. O. Osman, “Emulating the Internet of Things with QEMU,” M.S. thesis, Department of Computer Science and Engineering, Chalmers University of Technology, University of Gothenburg, Gothenburg, Sweden, 2019.

[64] G. Gallopeni, B. Rodrigues, M. Franco, and B. Stiller, “A Practical Analysis on Mirai Botnet Traffic,” *2020 IFIP Networking Conference (Networking)*, Paris, France, 2020, pp. 667-668.

[65] V. Ndatinya, Z. Xiao, V. R. Manepalli, K. Meng, and Y. Xiao, “Network forensics analysis using Wireshark,” *International Journal of Security and Networks*, vol. 10, no. 2, pp. 91-106, 2015, doi: 10.1504/IJSN.2015.070421.

[66] S. R. S. P. Paul, “Proposed Methods of IP Spoofing Detection and Prevention,” *International Journal of Science and Research (IJSR)*, vol. 2, no. 8, pp. 438–444, 2013.

[67] A. Beauchaine, M. Macchiaroli, and M. Yun, “IBoT: IoT botnet testbed,” *2021 16th International Conference on Computer Science & Education (ICCSE)*, Lancaster, United Kingdom, 2021, pp. 822–827, doi: 10.1109/ICCSE51940.2021.9569298.

[68] D. Datla *et al.*, “Wireless distributed computing: a survey of research challenges,” *IEEE Communications Magazine*, vol. 50, no. 1, pp. 144–152, Jan. 2012, doi: 10.1109/MCOM.2012.6122545.

[69] P. Urien and M. Dandjinou, “Introducing Smartcard Enabled RADIUS Server,” in *International Symposium on Collaborative Technologies and Systems (CTS’06)*, Las Vegas, NV, USA, 2006, pp. 74–80, doi: 10.1109/CTS.2006.54.

[70] K. A. T. Indah and I. N. K. Wardana, “The implementation of radius server for wifi pass using the mechanism of access point controller in Department of Electrical Engineering building, Bali State Polytechnic,” *Journal of Physics: Conference Series, International Conference on Applied Science and Technology (iCAST on Engineering Science)*, Bali, Indonesia, vol. 1450, no. 1, p. 012073, Feb. 2020, doi: 10.1088/1742-6596/1450/1/012073.

[71] R. F. M. Dollah, F. M. A., F. Arif, M. Z. Mas’ud, and L. K. Xin, “Machine learning for HTTP botnet detection using classifier algorithms,” *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, vol. 10, no. 1–7, pp. 27–30, 2018.

[72] S. Kumar, R. K. Sehgal, and S. Chamotra, "A Framework for Botnet Infection Determination through Multiple Mechanisms Applied on Honeynet Data," in *2016 Second International Conference on Computational Intelligence and Communication Technology (CICT)*, Ghaziabad, India, Feb. 2016, pp. 6–13, doi: 10.1109/CICT.2016.12.

[73] A. Rahimian, R. Ziarati, S. Preda, and M. Debbabi, "On the Reverse Engineering of the Citadel Botnet," in *6th International Symposium, Foundations and Practice of Security*, vol. 8352, pp. 408–425, 2014, doi: 10.1007/978-3-319-05302-8_25.

[74] A. K. Sood, R. J. Enbody, and R. Bansal, "Dissecting SpyEye – Understanding the design of third generation botnets," *Computer Networks*, vol. 57, no. 2, pp. 436–450, Feb. 2013, doi: 10.1016/j.comnet.2012.06.021.

[75] G. Saito and G. Stringhini, "Master of Puppets: Analyzing And Attacking A Botnet For Fun And Profit," *arXiv*, 2015, doi: 10.48550/arXiv.1511.06090.

[76] K. Allix, T. F. Bissyandé, Q. Jérome, J. Klein, R. State, and Y. Le Traon, "Large-scale machine learning-based malware detection," in *CODASPY '14: Proceedings of the 4th ACM Conference on Data and Application Security and Privacy*, Mar. 2014, pp. 163–166, doi: 10.1145/2557547.2557587.

[77] E. Alomari, S. Manickam, B. B. Gupta, P. Singh, and M. Anbar, "Design, deployment and use of HTTP-based botnet (HBB) testbed," *16th International Conference on Advanced Communication Technology*, 2014, pp. 1265–1269, doi: 10.1109/ICACT.2014.6779162.

[78] M. Simon and L. Huraj, "DDoS testbed based on peer-to-peer grid," in *2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES)*, IEEE, Oct. 2016, pp. 1181–1186, doi: 10.1109/SCOPES.2016.7955627.

[79] M. B. Al-Somaidai and N. R. S. Al-Hankawi, "Practical Private Cloud Testbed for Studying The Effects of A Botnet Attack," *Al-Rafidain Engineering Journal (AREJ)*, vol. 22, no. 2, pp. 73–82, 2014, doi: 10.33899/rengj.2014.87324.

[80] M. Bogdanoski, T. Shuminoski, and A. Risteski, "Analysis of the SYN Flood DoS Attack," *International Journal of Computer Network and Information Security (IJCNIS)*, vol. 5, no. 8, pp. 15–11, Jun. 2013, doi: 10.5815/ijcnis.2013.08.01.

[81] Z. Lu and H. Yang, *Unlocking the power of OPNET modeler*, 2012, doi: 10.1017/CBO9780511667572.

[82] J. Calvet *et al.*, "Isolated virtualised clusters: Testbeds for high-risk security experimentation and training," *3rd Workshop on Cyber Security Experimentation and Test (CSET 10)*, 2010.

[83] C. Siaterlis, B. Genge, and M. Hohenadel, "EPIC: A Testbed for Scientifically Rigorous Cyber-Physical Security Experimentation," *IEEE Transactions on Emerging Topics in Computing*, vol. 1, no. 2, pp. 319–330, Dec. 2013, doi: 10.1109/TETC.2013.2287188.

[84] Y.-L. Huang, B. Chen, M.-W. Shih, and C.-Y. Lai, "Security Impacts of Virtualization on a Network Testbed," in *2012 IEEE Sixth International Conference on Software Security and Reliability*, IEEE, Jun. 2012, pp. 71–77, doi: 10.1109/SERE.2012.17.

[85] M. Chernyshev, Z. Baig, O. Bello, and S. Zeadally, "Internet of Things (IoT): Research, Simulators, and Testbeds," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1637–1647, Jun. 2018, doi: 10.1109/JIOT.2017.2786639.

[86] J. Lai, J. Tian, K. Zhang, Z. Yang, and D. Jiang, "Network Emulation as a Service (NEaaS): Towards a Cloud-Based Network Emulation Platform," *Mobile Networks and Applications*, vol. 26, no. 2, pp. 766–780, Apr. 2021, doi: 10.1007/s11036-019-01426-0.

[87] C. Siaterlis, A. P. Garcia, and B. Genge, "On the Use of Emulab Testbeds for Scientifically Rigorous Experiments," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 2, pp. 929–942, 2013, doi: 10.1109/SURV.2012.0601112.00185.

[88] M. H. ElSheikh, M. S. Gadelrab, M. A. Ghoneim, and M. Rashwan, "BoTGen: A new approach for in-lab generation of botnet datasets," in *2014 9th International Conference on Malicious and Unwanted Software: The Americas (MALWARE)*, IEEE, Oct. 2014, pp. 76–84, doi: 10.1109/MALWARE.2014.6999406.

[89] A. M. Scott, C. Forbes, J. Clark, M. Carter, P. Glasziou, and Z. Munn, "Systematic review automation tools improve efficiency but lack of knowledge impedes their adoption: a survey," *Journal of Clinical Epidemiology*, vol. 138, pp. 80–94, Oct. 2021, doi: 10.1016/j.jclinepi.2021.06.030.

[90] J. Howard, "PythonP2PBotnet: A proof of concept P2P botnet written in Python using the Twisted framework," GitHub. [Online]. Available: https://github.com/jhoward321/PythonP2PBotnet. (Date accessed: Mar. 14, 2024).

[91] S. S. H. Hajjaj and K. S. M. Sahari, "Establishing remote networks for ROS applications via Port Forwarding," *International Journal of Advanced Robotic Systems*, vol. 14, no. 3, p. 172988141770335, May 2017, doi: 10.1177/1729881417703355.

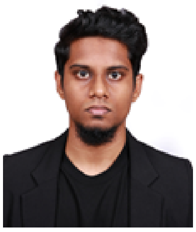
[92] W. Ahmed, *Proxmox Cookbook*, Birmingham, UK: Packt Publishing Ltd, 2015.

[93] B. S. Fagin, B. Klanderman, and M. C. Carlisle, "Making DNS Servers Resistant to Cyber Attacks: An Empirical Study on Formal Methods and Performance," in *2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC)*, IEEE, Jul. 2017, pp. 566–571, doi: 10.1109/COMPSAC.2017.165.

[94] B. Campbell, "Terraform in-depth," in *The Definitive Guide to AWS Infrastructure Automation: Craft Infrastructure-as-Code Solutions*, Berkeley, CA: Apress, 2020, pp. 123–203, doi: 10.1007/978-1-4842-5398-4_4.

[95] M. Zadka, "Salt Stack," in *DevOps* in Python, Berkeley, CA: Apress, 2019, pp. 121–137, doi: 10.1007/978-1-4842-4433-3_10.

BIOGRAPHIES OF AUTHORS



Mithiiran Parthipan is currently pursuing his Master's degree at the esteemed National Advanced IPv6 Centre, Universiti Sains Malaysia, building upon the strong foundation he laid with a Bachelor's degree in Computer Science from the School of Computer Science at the same institution. His academic journey reflects a profound dedication to the field, marked by exceptional achievements and a relentless pursuit of knowledge. His research interests are wide-ranging, encompassing areas such as botnets, malware, IoT, machine learning, deep learning, and distributed systems. His fascination with cybersecurity has led him to delve deep into the intricacies of botnets and malware, seeking innovative solutions for detection and mitigation. He can be contacted at email: mithiiran@student.usm.my.

Dr. Shams Ul Arfeen Laghari is a seasoned computer scientist with over 24 years of experience, is passionate about advancing cybersecurity and network security. Currently serving as an Assistant Professor at the School of ICT, Bahrain Polytechnic, he is committed to nurturing the next generation of cybersecurity experts. He earned his Ph.D. in Cybersecurity from Universiti Sains Malaysia (USM), where he delved into the intricacies of safeguarding digital systems. His academic journey was further enriched by a postdoctoral research position at the National Advanced IPv6 Centre (NAv6) within USM, where he contributed to cutting-edge research in network security. Beyond academia, he has honed his skills through collaborations with leading technology companies, gaining invaluable insights into the practical challenges and opportunities in the field. His industry experience has equipped him with a deep understanding of real-world cybersecurity threats and mitigation strategies. His research interests span a wide range of topics, including cybersecurity, the IoT, industry 4.0, distributed systems, cloud computing, and mobile cloud computing. His contributions to the field are evident in his numerous publications in top-tier international journals. He can be contacted at email: shams.ularfeen@polytechnic.bh.

Ashish Jaisan a distinguished scholar, holds a Master's degree in Cybersecurity from the prestigious National Advanced IPv6 Centre, Universiti Sains Malaysia, complementing his Bachelor's degree in Information Technology from Bharathiar University, Tamil Nadu, India. With a track record of authored and co-authored papers published in esteemed journals, he has demonstrated his scholarly prowess and commitment to impactful research. Engaging actively in industry-related projects, he has tackled performance optimization and cybersecurity challenges with aplomb. His research interests span a broad spectrum, encompassing botnets, malware, IoT, machine learning, deep learning, and distributed systems. He can be contacted at email: jaisan@student.usm.my.

Amber Baig received the B.S. and M.S. degrees in Computer Science from IMCS, University of Sindh, Jamshoro and the M. Phil and Ph.D. degrees in Computer Science from DCS, Isra University, Hyderabad. She is currently working as Associate Professor in the Department of Computer Science, Isra University, Hyderabad, Pakistan. Her research interests include cybersecurity, artificial intelligence, natural language processing, and human computer interaction. She can be contacted at email: amber.baig@isra.edu.pk.

Dr. Muhammad Asim Ali received the M.S. in Digital Communication from University of Kiel Germany in 2005 and Ph.D. in Electrical and Electronics Engineering from the University of Leeds in 2014. He is associated with the Department of Electrical Engineering Sukkur IBA University since 2008. His research interests include cyber-security, IoT protocols, and techniques for 5/6 G networks. He is head of the Digital Fabrication Lab at Sukkur IBA University. His other interests include embedded system design and fabrication. He can be contacted at email: asim.samejo@iba-suk.edu.pk.

Dr. Shankar Karuppayah is currently the Deputy Director and Senior Lecturer in the National Advanced IPv6 Centre, Universiti Sains Malaysia. He obtained his B.Sc. (HONS) Computer Science from Universiti Sains Malaysia in 2009 and his M.Sc. Software Systems Engineering from King Mongkut's University of Technology North Bangkok (KMUTNB) in 2011. In 2016, he obtained his Ph.D. degree. He is currently working actively on several cybersecurity projects and working groups. Till date, he has published in more than 30 articles in cybersecurity journals and conferences. He can be contacted at email: kshankar@usm.my.