



## Article

# A DDoS Detection and Prevention System for IoT Devices and Its Application to Smart Home Environment

Khalid Al-Begain <sup>1,\*</sup>, Murad Khan <sup>2</sup>, Basil Alothman <sup>2</sup>, Chibli Joumaa <sup>2</sup> and Ebrahim Alrashed <sup>3</sup><sup>1</sup> Kuwait College of Science and Technology, Kuwait City 35001, Kuwait<sup>2</sup> Department of Computer Science and Engineering, Kuwait College of Science and Technology, Kuwait City 35001, Kuwait<sup>3</sup> Department of Computer Engineering, Kuwait University, Kuwait City 12037, Kuwait

\* Correspondence: k.albegain@kcst.edu.kw

**Abstract:** The Internet of Things (IoT) has become an integral part of our daily life as it is growing in many fields, such as engineering, e-health, smart homes, smart buildings, agriculture, weather forecasting, etc. However, the growing number of IoT devices and their weak configuration raise many security challenges such as designing protocols to protect these devices from various types of attacks such as using them as bots for DDoS attacks on target servers. In order to protect IoT devices from enslavement as bots in a home environment, we develop a lightweight security model consisting of various security countermeasures. The working mechanism of the proposed security model is presented in a two-part experimental scenario. Firstly, we describe the working mechanism of how an attacker infects an IoT device and then spreads the infection to the entire network. Secondly, we propose a set of mechanisms consisting of filtration, detection of abnormal traffic generated from IoT devices, screening, and publishing the abnormal traffic patterns to the rest of the home routers on the network. We tested the proposed scheme by infecting an IoT device with malicious code. The infected device then infects the rest of the IoT devices in its network and launches a DDoS attack by receiving attack-triggering commands from the botmaster. Finally, the proposed detection mechanism is used to detect the abnormal traffic and block the connection of infected devices in the network. The results reveal that the proposed system blocks abnormal traffic if the packets from an IoT device exceeded a threshold of 50 packets. Similarly, the network packet statistics show that, in the event of an unwanted situation, the detection mechanism runs smoothly and avoids any possible delays in the network.



**Citation:** Al-Begain, K.; Khan, M.; Alothman, B.; Joumaa, C.; Alrashed, E. A DDoS Detection and Prevention System for IoT Devices and Its Application to Smart Home Environment. *Appl. Sci.* **2022**, *12*, 11853. <https://doi.org/10.3390/app122211853>

Academic Editor: Luis Javier Garcia Villalba

Received: 18 October 2022

Accepted: 17 November 2022

Published: 21 November 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

**Keywords:** Internet of Things; smart homes; DDoS; botnet

## 1. Introduction

The IoT comprises a selection of physical devices including sensors and microprocessors connected over a network and exchanging information. Depending on the application and setup, IoT devices can be found in a variety of places, such as homes, offices, industrial plants, and many other environments. These devices are deployed in a specific environment to provide several services, such as controlling home appliances in a smart home, remotely monitoring temperature and humidity in an agricultural farm, and supervising the operation of robots in manufacturing and production plants. This wide range of applications requires a variety of specific services, making ubiquity and inconspicuousness two important characteristics of these devices [1]. It is estimated that over 50 billion IoT devices will be connected in the next few years [2]. The simplicity of setting up an IoT system and the ease of extracting the needed information make it suitable for a lot of needed services. At the same time, this allows attackers to target these devices to take advantage of the information and services provided. Similarly, the recent literature shows that the number of Distributed Denial of Service (DDoS) attacks has significantly increased in the

last decade. One of the reasons for this increase is that more devices are connected to the internet every day. Thus, it is easy for the attacker to attack the weakly configured IoT devices and take them over in a form of enslavement in order to launch DDoS attacks. Therefore, sophisticated security mechanisms are needed to provide enough security to these devices.

The aim of a security mechanism is to protect the IoT devices in a smart home setup from enslavement, where a bot program—malware—is installed by the attacker on one of the devices connected to the home network. The bot program can operate as a bot master infecting other devices on the same network. It will turn infected appliances into zombie devices and will perform malicious actions, such as flooding attacks i.e., DDoS attacks. Several solutions to protect IoT devices from DDoS attacks are described in the literature. Some security mechanisms employ deep learning models to train a neural network that will be used to detect malicious activity patterns in the network traffic originating from IoT devices. However, such solutions are only favorable in those situations when the IoT devices are provided with high processing power and enough memory [3]. Another proposed solution is to use a cluster of Raspberry Pis to control the entire home IoT network [4,5]. However, in this case, the management of the cluster would be challenging to maintain. Lightweight security is a favorable solution for protecting IoT devices from enslavement. One of the reasons for this is that IoT devices have limited memory and processing power, making lightweight security protocols a better option [6,7]. This solution is designed for IoT devices that require extra management on the network layer. Therefore, modifying the network layer protocols requires supplementary work and management.

The work presented in this article proposes a security framework for a home environment. The proposed solution scheme is described in a two-part experimental scenario. First, an attack is launched to enslave an IoT device by installing malicious code. The enslaved IoT device will be transformed into a bot that will infect the other connected IoT devices with the same malicious code. The infected devices turned into bots will launch a staged DDoS attack on the targeted server. Second, a security mechanism is devised to monitor traffic and detect the unusual traffic activity on the edge router from the IoT devices now turned into bots. The edge router is programmed to block unusual traffic from the bots. Then all open ports on the IoT devices will be blocked to prevent future malicious connections. Finally, the edge router will distribute the information about the unusual traffic to the rest of the routers on the same network.

The rest of the paper is divided into the following parts. Section 2 presents a thorough literature study of the current botnet and related security mechanisms. Section 3 presents the problems available in the current IoT environments. Similarly, Section 4 presents the proposed scheme with various phases. Section 5 presents the results and discussion of the experimentation study. Finally, the conclusions are given in Section 6.

## 2. Related Works

Recently, IoT networks have faced a number of security challenges due to the hardware limitations and processing capabilities of IoT devices and home edge routers. Simultaneously, in the last decade, the number of security attacks on the IoT network has increased at an unprecedented rate. Many of these attacks enslave IoT devices in a home environment and use them as bots for attacking a victim server. For example, a well-known botnet called Mirai was launched in October 2016; infected IoT devices bombarded a victim DNS server with 1.2 Tbps of data [8,9]. Similarly, another IoT botnet called BashLite was used to launch a DDoS attack against the victim servers [10,11]. The BashLite botnet searched for possible credentials among 6 generic usernames and 14 generic passwords. The conceptual idea of these botnets is to brute force all the possible combinations and try them until they get access to the IoT device. Once a successful connection is established, the botmaster installs the malicious code in the botnet and later uses them in a DDoS attack on the victim server. Since IoT devices are weakly configured and even sometimes available for connection

without proper authentication, the attacker finds it easy to load the malicious code onto them.

A number of solutions are used in the current literature for detecting abnormal traffic generated by IoT devices [12–15]. These methods are mainly based on the idea of using machine and deep learning techniques to check the network traffic on the home edge router and classify them as normal and abnormal. However, such solutions require high memory and processing requirements, which are not available in most cases. In addition, fulfilling these requirements and implementing the solutions on the home edge router may lead to delaying the processing of urgent and necessary tasks. For example, in [16], a study covers a two-fold machine learning system to detect IoT botnet attacks. Once the IoT devices get compromised and start DDoS attacks, the machine learning system generates 33 types of deep learning models using ResNet-18 to scan 60 DDoS attacks from publicly available datasets. The authors also discussed the lifecycle of the botnet, which they divided into five stages: first, scanning, second, malware injection, third, botnet connection, fourth, command execution, and, lastly, maintenance and up-gradation. Similar research in [17] proposed a hybrid approach to detect IoT botnet attacks using supervised and unsupervised machine learning techniques using a publicly available BaIoT dataset [18]. N-BaIoT stands for Network-Based Anomaly Internet of Things, which uses deep learning techniques to perform anomaly detection. The authors examined two botnets, Mirai and BashLite, to collect traffic data before and after the infections of the IoT devices. The authors evaluated transfer learning techniques by assessing the accuracy of models trained on specific devices when they are applied to identical devices [18]. Nguyen et al. [19] created a novel graph-based approach to detect IoT botnets in a lightweight method using PSI-graph that discovers botnet lifecycle with deep learning. They trained different IoT botnet datasets using Mirai, BashLite, Benign, and other botnets with a total of 10,000 samples. The authors clarify the botnet lifecycle and how IoT devices are infected with a botnet. They also explained the working of a bot when it logs into a device: the bot infects a loader, which is used to download and execute the corresponding binary version of the botnet, typically via FTP and HTTP [19]. Recently, researchers studied different IoT systems' major attacks within the network and physical layer [20]. Similarly, they compared with normal botnets, IoT botnets such as Mirai, and other existing tools to detect botnets. Further, they divided IoT botnet detection into two parts: Host-Based and Network-Based Detection Techniques. In [20], the authors created a sandbox called V-sandbox to use for IoT botnet dynamic analysis with more than 9000 IoT botnet samples. These samples were divided into 6000 IoT bots and 2900 IoT benign samples [21]. Similarly, HKK Idriss et al., in [22], discussed the Mirai botnet and how a DDoS attack detection could be performed using malware hash detection and behavior-based detection. The authors performed the scanning in an IoT networking environment using shodan.io [23] and other vulnerable search engines. In a similar research work in [24], the authors analyzed Mirai botnet traffic using Wireshark and pyshark. Further, they simulated botnet attacks using six local testbeds.

In recent literature, we have noticed several DDoS attacks on IoT devices in an IoT environment [25]. For example, an attack is launched on an IoT sensor that is connected to healthcare data, which may result in a great loss to the patient. In [26], the authors proposed a system to provide security to the IoT systems that are further connected to a healthcare system. The proposed system efficiently overcomes the security challenges of the healthcare system using encoding and two-stage security mechanisms. Finally, the test results show that the proposed system can efficiently protect healthcare data from any sort of security breach. Similarly, another security system for an IoT environment is proposed for protecting the data generated from parking sensors [27]. The proposed system uses a cloud-based infrastructure to protect the data from leaking during the transmission to the users and parking management administrator. However, such a system greatly relies on the use of a cloud-based system, which results in a high cost for installation and maintenance. Therefore, security solutions based on Quantum Computing are essential

to provide security to a cloud system [28]. Also, the security parameter settings in such systems are required to ensure efficient and fast security [29].

Concluding the above literature, we noticed that there is still a need for a system that can be used to detect and prevent DDoS attacks in IoT environments. Also, there is room for research to use lightweight security solutions for IoT networks where the devices have limited battery power and memory.

### 3. Problem Statement

The IoT devices and router attached to a home always have limited memory and processing power. One of the reasons is that IoT devices are not meant to process large amounts of data and, hence, are supplied with limited requirements of memory and processing power to reduce cost. Therefore, providing solutions based on extensive algorithms such as machine and deep learning require high amounts of memory and data processing. Therefore, such solutions are inefficient and may cause a large delay in processing the normal traffic generated from these devices. Further, IoT devices are rarely configured with great care, and therefore, a weak configuration is always possible. These challenges provide a favorable environment for the attacker. The attacker and intruders use these weakly configured IoT devices to launch various types of attacks on different servers. Recent studies have shown that these devices are used by attackers by infecting them with malicious codes and controlling them with a C&C server. The attacker, which is sometimes referred to as a botmaster, controls these weakly configured IoT devices and launches and manages attacks on various servers. Therefore, in this study, we identified the weakly configured devices and provided enough security mechanisms to control the abnormal traffic generated from them. Also, the home router is provided with a detection mechanism that requires less memory and processing time.

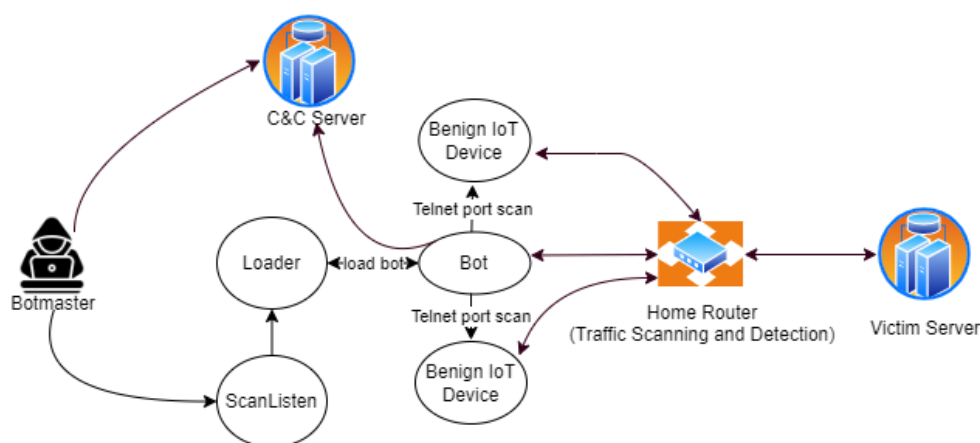
### 4. Proposed Scheme

The experimental scenario of the proposed scheme is divided into the following two main phases: Setting up the botnet attack phase, and the detection and prevention mechanism.

#### 4.1. Overview of the Proposed Scheme

In this article, our aim is twofold. Firstly, we will show how to infect a weakly configured IoT device by downloading the malware into it, and secondly, how to detect abnormal traffic i.e., a DDoS attack generated by the infected IoT devices. Further, the botmaster is responsible for communicating with the bot using the Command-and-Control Server (C&C). As soon as a new victim is ready, the botmaster downloads the malicious code onto it. After downloading the malicious code into the IoT device, the botmaster sends the necessary commands to launch an attack on a particular server. Finally, the victim bot sends a huge amount of data to the server to prevent it from providing services to benign users.

As soon as an attack is launched, the proposed scheme uses a detection mechanism to block the incoming traffic from the infected devices. In the detection mechanism, we are using a threshold on the number of packets sent to a destination IP address for a specific duration. For example, if the number of packets from all the IoT devices exceeds a predefined threshold, the home router will block the connections from all those IoT devices. Similarly, the home router is also configured to manage the weakly configured IoT devices by closing their open ports. Figure 1 shows the overall architecture of the proposed scheme.



**Figure 1.** The working mechanism of the proposed scheme.

#### 4.2. Setting up the Botnet on Victim IoT Devices

In order to set a bot in the victim machine, it is necessary to gain access to the victim machine. The idea of setting a bot in the victim machine is to first infect a weakly configured IoT device. For this purpose, we first select an IoT device and scan its open ports using the brute force technique. However, this is not as simple as it may sound, as the devices in the IoT environment always use some basic security protocols to stop anonymous incoming connections. For example, if an IoT device is using an SSH shell to establish a connection to it, then it is not possible to make a connection from a device that is unknown on the network. Similarly, any new connection to the device must have the credentials to form a connection to the device. Normally, all the IoT devices on the network use similar security credentials; therefore, if an attacker somehow gains access to one of them, it might be easy to infect the rest of them. However, in the case of a botnet, the botmaster always checks the status of the victim machines using the C&C server. Therefore, it is necessary for each infected IoT device to send its status to the C&C server. As soon as the botmaster is assured all the IoT devices on the network are infected, then it is ready to send a command to launch an attack on a server. Finally, these devices launch an attack on the server to somehow stop it from providing services to legitimate users on the network.

This entire process is divided into the following steps.

**Step 1:** Weakly configured IoT devices are identified using the brute force attack to download and infect bots into it. The brute force mechanism works on the principle of scanning all the available ports of an IoT device. These ports are normally left open as a result of the weak configuration and are thus exploited by the attacker.

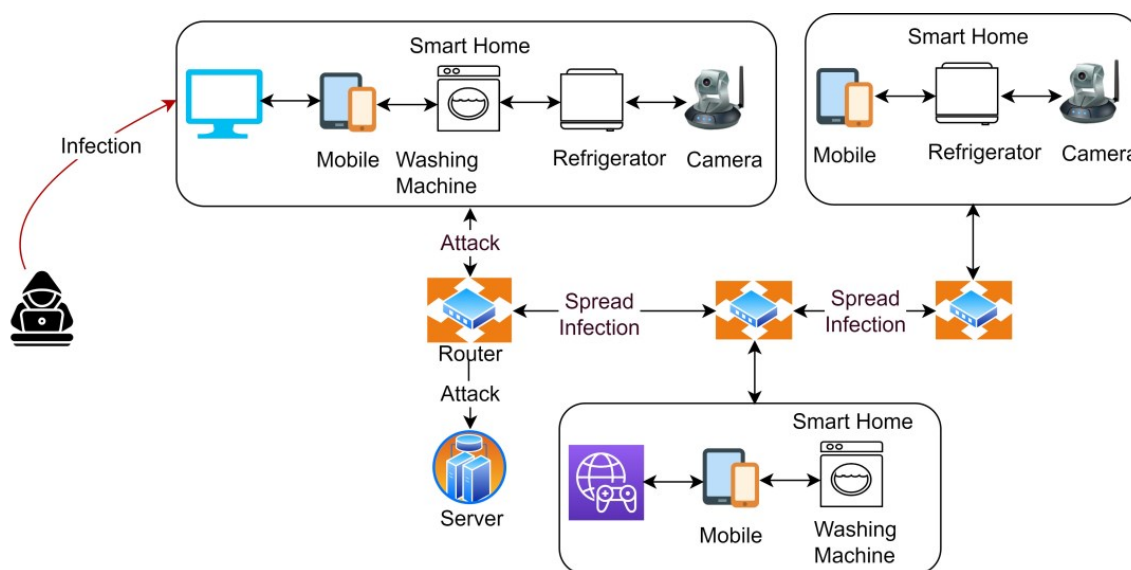
**Step 2:** Once the device is infected, the bot accesses the interface (either shell or graphical user interface) and collects various characteristics of the device. The bot shares this information with the reporting server using a different port. Therefore, it becomes difficult to check the abnormal activities of the bot. Similarly, the attacker can communicate with the bot using the C&C server to avoid being located.

**Step 3:** Once the information is loaded to the server, the botmaster regularly checks on new victims' machines and the status of the current bots attached to the server. This also helps the botmaster to regularly check on all the bots connected to the C&C server. The moment all the bots are connected, the botmaster can easily guide them to launch an attack.

**Step 4:** As soon as the botmaster receives the information of new weakly configured devices, it infects them by downloading bots to them. These new bots then connect to the rest of the IoT devices on the same network to spread the infection.

**Step 5:** Finally, the botmaster sends the attack-triggering command to the bots to launch a DDoS attack against a server by flooding the server with unnecessary information.

This entire attack process is presented in Figure 2.



**Figure 2.** Attack Scenario.

#### 4.3. Detection and Mitigation of DDoS Attack at Home Router

In order to detect abnormal traffic, i.e., a DDoS attack, the home router is configured with multiple security mechanisms. These include filtration, detection, screening, and publishing. These mechanisms are discussed in the following subsections.

##### 4.3.1. Filtration

Before detecting abnormal network traffic, it is important to filter the network traffic. For example, if a device is uploading a huge amount of traffic after each time frame, then it is necessary to check whether this traffic is legitimate or not. A video camera might send video traffic that is, legitimately, heavier in size for a particular amount of time. Therefore, we cannot classify video traffic as abnormal traffic. Thus, the IoT devices connected to the network generate dynamic traffic, and therefore, it is not easy to classify them. Similarly, in the recent literature, we have seen many IDS techniques using signature and anomaly-based filtration. However, such techniques may perform inadequately in the case of overhead traffic. In normal circumstances, most of the traffic generated from IoT devices is benign; therefore, it is better to filter such traffic in advance. One of the reasons for such filtration is to reduce target traffic for the detection phase. Traffic filtration is the first step in ensuring the effectiveness of the system. In the proposed filtration mechanism, we used the list-based mechanism, i.e., a blacklist and a whitelist, to classify the traffic as normal and abnormal.

##### Blacklist-Based Filtration

We will first create a blacklist of all the IP addresses that were previously involved in malicious activities. If any incoming traffic from the IoT devices is matched with the blacklisted IP addresses, the packets will be handled according to the predefined network security rules. These rules will be updated over time to ensure the unblocking of traffic if, in the future, a blacklisted IP address becomes a whitelisted IP address.

##### Whitelist-Based Filtration

The traffic from whitelisted IP addresses can go directly to the corresponding server. Such filtration helps in reducing the burden of checking each packet generated from every IoT device on the network. Further, the creation of a whitelist in our scheme is performed and managed by the network administrator.

In addition to the filtration discussed above, it is necessary to sample the packets to reduce the number of packets that will pass to the detection phase. In the literature, two

schemes are widely used to sample the network traffic, (1) packet-based and (2) flow-based. In the case of the proposed scheme, we will use packet-based sampling techniques. From our experience, flow-based techniques are better where hundreds and thousands of nodes are connected to the network. The packet-based sampling will further help in reducing the number of packets to be scanned by the proposed detection technique.

#### 4.3.2. Detection Phase

Using the filtration mechanism proposed in the previous step, the number of packets for the detection phase becomes easy to handle. In this step, our approach is to scan the rest of the network traffic for abnormal and normal traffic. There is still a possibility that a bot can generate abnormal traffic to a target server. In the home router, we set a threshold for the number of packets distant for a particular server, i.e., destination IP address. For example, if the IoT devices connected to the network send packets over a predefined threshold, the traffic will be considered abnormal traffic, and the IP address of the respective devices will be blacklisted. However, the normal traffic, as separated in step 4.3.1, can still pass the router directly. One of the reasons for using a filtration mechanism is that it can help in reducing the processing on the home router. In addition to setting a threshold for abnormal traffic, we also use statistical analysis on the data passing through the router. This analysis includes checking the frequency of the packet’s generation from a particular device, the average number of packets passing through the home router at a particular time, the average size of the data packets, and the length of a connection time.

In recent studies, we have noticed that the botnet always attacks at once, but they are changing the behavior of the attack by randomly sending the packets from each infected device. This gives botmasters the ability to make the behavior of the network traffic appear benign to the underlying security protocols. For example, the IDS works either on signature or anomaly-based detection mechanisms. Thus, the botmaster controls the traffic generated from each infected device with a random amount of data making it difficult for the IDS to differentiate between normal and abnormal traffic. In contrast, our approach is intelligent enough to handle such traffic by periodically using statistical analysis of the traffic data. However, along with the statistical analysis, the threshold mechanism always provides the first line of defense against a DDoS attack. In addition, the statistical analysis helps us to communicate the DDoS attack traffic with the rest of the routers attached to the current router. The detection mechanism is shown in Figure 3.

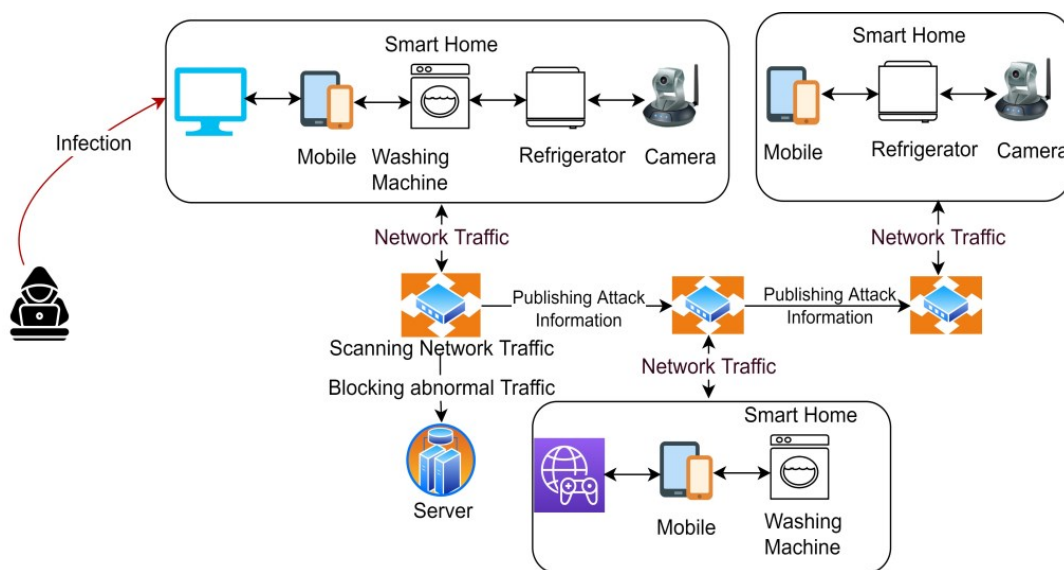


Figure 3. The detection scenario.

#### 4.3.3. Screening Phase

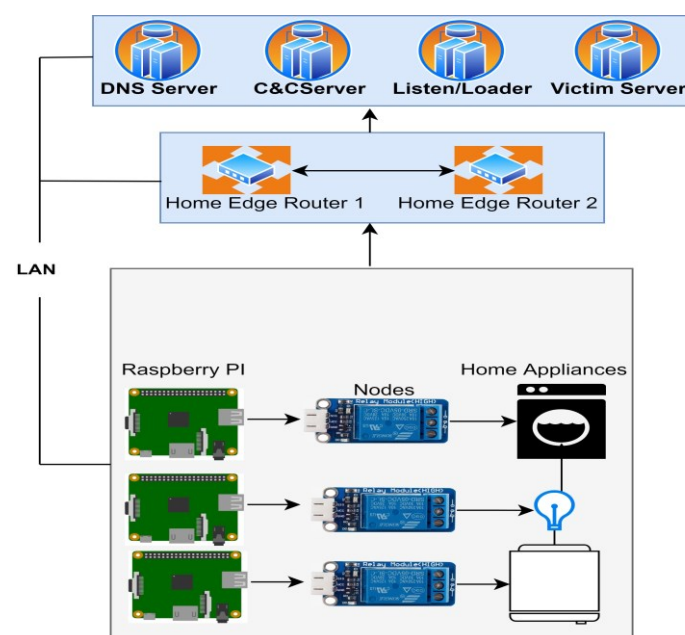
The screening phase is added to provide another line of defense. Over time, new IoT devices might be added to the home network. Therefore, the screening phase helps us to generate a benign profile for each new device. In the screening phase, the profile of each IoT device is generated based on the results of the statistical analysis of step 4.1.2. This helps us to reduce the maintenance and management of IoT devices in the future. Also, the logs help us to maintain a profile for each IoT device, and it updates over time. The screening phase does not help directly in identifying the normal and abnormal traffic, but it is useful in generating a legitimate profile of the devices. Similarly, it can greatly help in the future to add new devices to the network by assigning legitimate profiles of benign IoT devices.

#### 4.3.4. Publishing of Malicious Information with Other Home Routers

In the screening and detection phase, a router logs information on malicious activities of the IoT devices. We use this information by sharing it with the rest of the home routers. This significantly reduces the processing time for detection and screening for the rest of the routers. In order to share the knowledge of normal and abnormal traffic with the rest of the home routers, we have developed a sharing mechanism by constructing a shared memory among all the available routers. This shared memory concept is an entirely novel concept proposed in this research. First, we will develop a safe memory location in the current router, and the routers attached to it will periodically check the current router for malicious information. For example, if the other home router is adding a new device to its network, it can communicate with the current router and obtain specific configuration information. We will share detailed information about this concept in our future research.

### 5. Results and Discussion

The proposed scheme is tested in an environment where a number of IoT devices are attached to a home router. These devices, the home router, C&C server, DNS server, victim server, etc. are all attached to the same LAN. However, in practice, this is not the case, and they could be attached to different LANs. However, the location of these networking entities does not affect the overall concept of the proposed system. The experimental setup of the proposed scheme is depicted in Figure 4.



**Figure 4.** Testbed used in experimenting with the proposed system.



As the C&C server is available on the same LAN, we download the malicious code to one of the IoT devices to infect it. This is simply done by running malicious code on any of the IoT devices. Once the IoT device is infected with malicious code, it is then connected to the home router via the gateway. In order to design a common gateway to format the traffic to be understandable over the internet, we used a switch and configured it as a gateway. This switch is further attached to the home router. Each home appliance is attached to a Raspberry Pi 4.0 module with 4 GB of RAM and a Quad-core 64-bit processor. Similarly, the specifications of the home router used for the experiment are: dual Gigabit Ethernet ports, dual USB 3.0 ports, Raspberry Pi Compute Module 4, 4 GB RAM, 32 GB eMMC, and a Quad-core 64-bit processor.

The traffic passing through the home router is captured with Wireshark at different times. Figure 5 shows the main open dataset representation, which was captured after the infected IoT devices launched a DDoS attack on the victim server. As depicted in Figure 5, the infected IoT devices send a huge amount of SYN packets to the victim server without waiting for the acknowledgment message from the server. This slows down the working of the victim sever in providing services to legitimate IoT devices. In order to check that the victim server failed to fulfill the services of the benign IoT devices, a benign IoT device is selected to send a request to the victim server to obtain the required services. However, as the victim server is busy responding to the infected devices, the benign IoT device will never get a chance to receive the services from the victim server. Finally, the victim server will reset and stop listening to new requests once its cache is clear and ready to receive new messages. In the case of the IoT scenario, the data generated from each device have high preferences. For example, if a camera detects an abnormal behavior of the main gate and tries to send this urgent data to the homeowner. However, the router is busy providing the services to the infected device, which may cause a delay in sending the information to the homeowner on time. Finally, it may result in something unwanted occurring, or damage to the house.

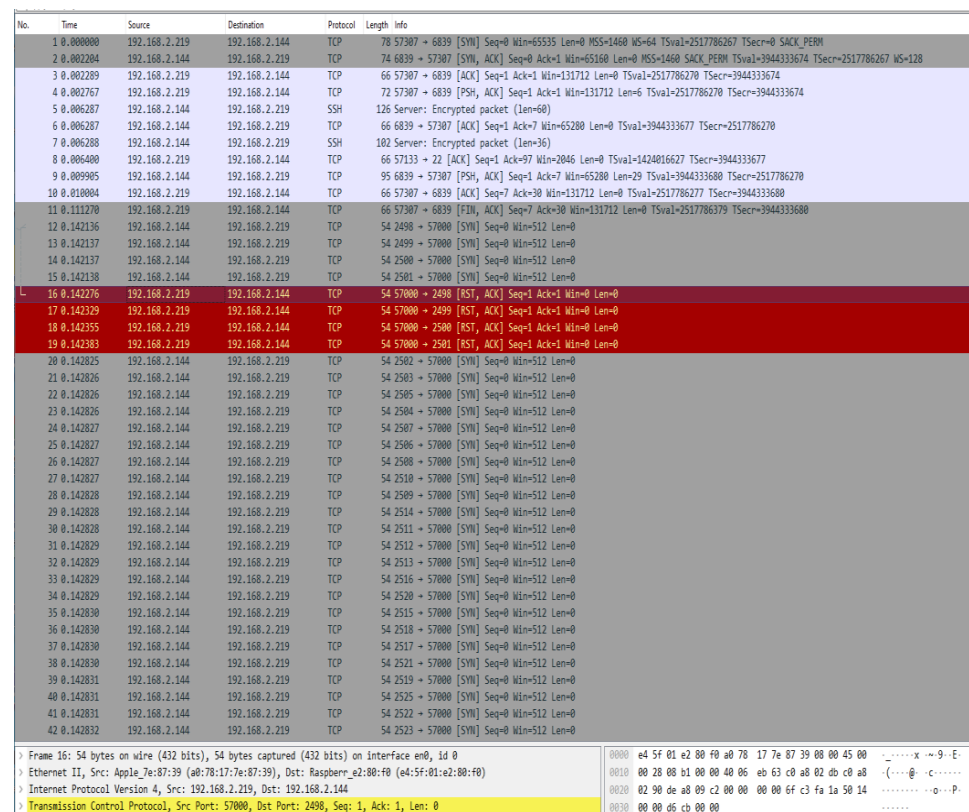


Figure 5. A SYN flood attack on the victim server.

The statistics of the packets generated during the attack are shown in Figure 6. This statistic helps us in screening the packets in the future. In addition, a profile for each IoT device could be built by saving the statistics of each IoT device traffic in the log. For example, based on the packet statistics of each IoT device, we can build an appropriate and suitable traffic generation profile. The traffic generation profile can help in reducing the limitation of the system in terms of blocking normal traffic. For example, we can set a high packet generation threshold for a camera during a specific time of the day. The packet size of streaming data is always greater than that of normal packets and thus we can set the camera to transmit information in a specific period. For example, if a camera is scheduled to send the packet during a specific time, then the router rules could be changed to avoid scanning packets based on the pre-defined threshold of 50 packets. In the case of a camera, the threshold could be increased to 100, etc. Similarly, a refrigerator is continuously creating data, thus such a profile could be built for a refrigerator so that the router does not block the traffic from the refrigerator if the packet transmission exceeded 50 packets.

Statistics	
Measurement	Captured
Packets	160557
Time span, s	5.760
Average pps	27876.3
Average packet size, B	54
Bytes	8671049
Average bytes/s	1505 k
Average bits/s	12 M

Figure 6. Source-destination-wise packet statistics.

In order to validate the working mechanism of the proposed detection system, we have shown a scenario in Figure 7 where the connection from a home appliance is blocked by sending many packets to a victim server. As we can see in Figure 7, the network traffic originating from the source IP address, i.e., 192.168.2.144, is rejected by the router due to the sending of abnormal traffic to the victim server.

Pkts.	Traffic	Target	Prot.	In	Out	Source	Destination	Options	Comment
0	0 B	<a href="#">output_lan_rule</a>	all	*	*	0.0.0.0/0	0.0.0.0/0	-	Custom lan output rule chain
0	0 B	<a href="#">zone_lan_dest_REJECT</a>	all	*	*	192.168.2.144	0.0.0.0/0	-	Reject LAN traffic
0	0 B	<a href="#">zone_lan_dest_ACCEPT</a>	all	*	*	0.0.0.0/0	0.0.0.0/0	-	-

Figure 7. Abnormal traffic originating from a malicious device is blocked by the router.

Finally, we show the TCP port status of the source node sending packets to the destination (Figure 8). As we can see in Figure 8, the traffic from the source IP address i.e., malicious IoT devices, is slowly increasing over time.

Address	Port	Packets	Bytes	Tx Packets	Tx Bytes	Rx Packets	Rx Bytes
192.168.2.144	0	4 bytes	216 bytes	2 bytes	108 bytes	2 bytes	108 bytes
192.168.2.144	1	4 bytes	216 bytes	2 bytes	108 bytes	2 bytes	108 bytes
192.168.2.144	2	2 bytes	108 bytes	1 bytes	54 bytes	1 bytes	54 bytes
192.168.2.144	3	3 bytes	162 bytes	2 bytes	108 bytes	1 bytes	54 bytes
192.168.2.144	4	4 bytes	216 bytes	2 bytes	108 bytes	2 bytes	108 bytes
192.168.2.144	5	4 bytes	216 bytes	2 bytes	108 bytes	2 bytes	108 bytes
192.168.2.144	7	2 bytes	108 bytes	2 bytes	108 bytes	0 bytes	0 bytes
192.168.2.144	8	1 bytes	54 bytes	1 bytes	54 bytes	0 bytes	0 bytes
192.168.2.144	9	3 bytes	162 bytes	2 bytes	108 bytes	1 bytes	54 bytes
192.168.2.144	10	4 bytes	216 bytes	2 bytes	108 bytes	2 bytes	108 bytes
192.168.2.144	11	2 bytes	108 bytes	1 bytes	54 bytes	1 bytes	54 bytes
192.168.2.144	13	2 bytes	108 bytes	1 bytes	54 bytes	1 bytes	54 bytes
192.168.2.144	14	3 bytes	162 bytes	2 bytes	108 bytes	1 bytes	54 bytes
192.168.2.144	15	4 bytes	216 bytes	2 bytes	108 bytes	2 bytes	108 bytes
192.168.2.144	16	4 bytes	216 bytes	2 bytes	108 bytes	2 bytes	108 bytes
192.168.2.144	17	3 bytes	162 bytes	2 bytes	108 bytes	1 bytes	54 bytes
192.168.2.144	18	4 bytes	216 bytes	2 bytes	108 bytes	2 bytes	108 bytes
192.168.2.144	19	3 bytes	162 bytes	2 bytes	108 bytes	1 bytes	54 bytes
192.168.2.144	20	4 bytes	216 bytes	2 bytes	108 bytes	2 bytes	108 bytes
192.168.2.144	22	16 bytes	1.633 KiB	9 bytes	1.170 KiB	7 bytes	474 bytes
192.168.2.144	23	3 bytes	162 bytes	2 bytes	108 bytes	1 bytes	54 bytes
192.168.2.144	24	4 bytes	216 bytes	2 bytes	108 bytes	2 bytes	108 bytes
192.168.2.144	25	1 bytes	54 bytes	1 bytes	54 bytes	0 bytes	0 bytes
192.168.2.144	26	3 bytes	162 bytes	2 bytes	108 bytes	1 bytes	54 bytes
192.168.2.144	28	2 bytes	108 bytes	1 bytes	54 bytes	1 bytes	54 bytes
192.168.2.144	29	4 bytes	216 bytes	2 bytes	108 bytes	2 bytes	108 bytes
192.168.2.144	30	4 bytes	216 bytes	2 bytes	108 bytes	2 bytes	108 bytes
192.168.2.144	32	3 bytes	162 bytes	2 bytes	108 bytes	1 bytes	54 bytes
192.168.2.144	34	4 bytes	216 bytes	2 bytes	108 bytes	2 bytes	108 bytes
192.168.2.144	35	3 bytes	162 bytes	2 bytes	108 bytes	1 bytes	54 bytes
192.168.2.144	36	3 bytes	162 bytes	2 bytes	108 bytes	1 bytes	54 bytes
192.168.2.144	37	3 bytes	162 bytes	2 bytes	108 bytes	1 bytes	54 bytes
192.168.2.144	38	4 bytes	216 bytes	2 bytes	108 bytes	2 bytes	108 bytes
192.168.2.144	40	2 bytes	108 bytes	2 bytes	108 bytes	0 bytes	0 bytes
192.168.2.144	41	2 bytes	108 bytes	2 bytes	108 bytes	0 bytes	0 bytes
192.168.2.144	42	2 bytes	108 bytes	1 bytes	54 bytes	1 bytes	54 bytes
192.168.2.144	43	3 bytes	162 bytes	2 bytes	108 bytes	1 bytes	54 bytes
192.168.2.144	45	4 bytes	216 bytes	2 bytes	108 bytes	2 bytes	108 bytes
192.168.2.144	46	3 bytes	162 bytes	2 bytes	108 bytes	1 bytes	54 bytes
192.168.2.144	48	3 bytes	162 bytes	2 bytes	108 bytes	1 bytes	54 bytes
192.168.2.144	49	3 bytes	162 bytes	2 bytes	108 bytes	1 bytes	54 bytes
192.168.2.144	50	1 bytes	54 bytes	1 bytes	54 bytes	0 bytes	0 bytes
192.168.2.144	51	2 bytes	108 bytes	1 bytes	54 bytes	1 bytes	54 bytes
192.168.2.144	53	2 bytes	108 bytes	1 bytes	54 bytes	1 bytes	54 bytes
192.168.2.144	54	3 bytes	162 bytes	2 bytes	108 bytes	1 bytes	54 bytes

Figure 8. TCP port-based status.

### 6. Conclusions

The security of IoT devices is highly essential due to many recent security attacks. However, the solutions presented in the current literature are mainly dependent on time-consuming algorithms and methods such as deep learning models. Therefore, there is a need for lightweight security models for IoT environments. In this article, we proposed an IoT botnet setup and demonstrated its detection and prevention mechanisms in a real testbed environment. The experimental scenario is divided into two phases; in phase 1, a botnet is set and configured in a home environment. The botmaster controls the bot by infecting it with malicious code. The infected IoT device spreads the code to the rest of the IoT devices by continuously scanning their open ports using the brute force phenomenon. Similarly, once all the devices are infected, the botmaster triggers the launch of an attack by sending a huge number of packets to a victim server. In phase 2, we present a detection mechanism to detect the abnormal traffic generated by the bots to prevent them from sending the DDoS traffic to the victim server by blocking their connection. Further, we show the analysis of the proposed scheme by capturing the network traffic during the DDoS attack. Finally, the network traffic statistics are also presented after detection and prevention are carried out. We have shown in the experiments that the connection from malicious nodes is identified and blocked for a certain period of time. In addition, the packet statistics show that the proposed system does not cause much delay to normal traffic even in the event of a security attack.

**Author Contributions:** K.A.-B. and E.A. conducted the research into the academic landscape and drafted and supervised the research. M.K. and B.A. did the flowchart implementation design while C.J. and E.A. work on the formal analysis of the article. K.A.-B. and M.K. work on implementation and M.K., B.A., and C.J. prepare the initial draft of the paper. The paper was written jointly by all the authors. All authors have read and agreed to the published version of the manuscript.

**Funding:** The work in this paper was funded by the Kuwait Foundation for Advancement of Science (KFAS) under Project No. (PR17-18QI-03).

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Silva, B.N.; Khan, M.; Han, K. Towards sustainable smart cities: A review of trends, architectures, components, and open challenges in smart cities. *Sustain. Cities Soc.* **2018**, *38*, 697–713. [CrossRef]
2. Research, J. Iot Connections to Grow 140% to Hit 50 Billion By 2022, as Edge Computing Accelerates Roi. Available online: <https://www.juniperresearch.com/press/iot-connections-to-grow-140pc-to-50-billion-2022> (accessed on 31 August 2022).
3. Jan, B.; Farman, H.; Khan, M.; Imran, M.; Islam, I.U.; Ahmad, A.; Ali, S.; Jeon, G. Deep learning in big data analytics: A comparative study. *Comput. Electr. Eng.* **2019**, *75*, 275–287. [CrossRef]
4. Doucet, K.; Zhang, J. Learning cluster computing by creating a Raspberry Pi cluster. In Proceedings of the SouthEast Conference, Kennesaw, GA, USA, 7–8 October 2017.
5. Coelho, J.; Nogueira, L. Enabling Processing Power Scalability with Internet of Things (IoT) Clusters. *Electronics* **2021**, *11*, 81. [CrossRef]
6. Fotohi, R.; Pakdel, H. A lightweight and scalable physical layer attack detection mechanism for the internet of things (IoT) using hybrid security schema. *Wirel. Pers. Commun.* **2021**, *119*, 3089–3106. [CrossRef]
7. Rao, V.; Prema, K.V. A review on lightweight cryptography for Internet-of-Things based applications. *J. Ambient. Intell. Humaniz. Comput.* **2021**, *12*, 8835–8857. [CrossRef]
8. Koliass, C.; Kambourakis, G.; Stavrou, A.; Voas, J. DDoS in the IoT: Mirai and other botnets. *Computer* **2017**, *50*, 80–84. [CrossRef]
9. Antonakakis, M.; April, T.; Bailey, M.; Bernhard, M.; Bursztein, E.; Cochran, J.; Zhou, Y. Understanding the mirai botnet. In Proceedings of the 26th USENIX security symposium (USENIX Security 17), Vancouver, BC, USA, 23 May 2017; pp. 1093–1110.
10. Marzano, A.; Alexander, D.; Fonseca, O.; Fazzion, E.; Hoepers, C.; Steding-Jessen, K.; Chaves, M.H.P.C.; Cunha, Í.; Guedes, D.; Meira, W. The evolution of bashlite and mirai iot botnets. In Proceedings of the 2018 IEEE Symposium on Computers and Communications (ISCC), Natal, Brazil, 25–28 June 2018; pp. 813–818.
11. Tanabe, R.; Tamai, T.; Fujita, A.; Isawa, R.; Yoshioka, K.; Matsumoto, T.; Gañán, G.; Van Eeten, M. Disposable botnets: Examining the anatomy of iot botnet infrastructure. In Proceedings of the 15th International Conference on Availability, Reliability and Security, Dublin, Ireland, 25–28 August 2020; pp. 1–10.
12. Alani, M.M. BotStop: Packet-based efficient and explainable IoT botnet detection using machine learning. *Comput. Commun.* **2022**, *193*, 53–62. [CrossRef]
13. Kumar, A.; Shridhar, M.; Swaminathan, S.; Lim, T.J. Machine learning-based early detection of IoT botnets using network-edge traffic. *Comput. Secur.* **2022**, *117*, 102693. [CrossRef]
14. Maurya, S.; Kumar, S.; Garg, U.; Kumar, M. An efficient framework for detection and classification of iot botnet traffic. *ECS Sens. Plus* **2022**, *1*, 026401. [CrossRef]
15. Nguyen, T.N.; Ngo, Q.D.; Nguyen, H.T.; Nguyen, G.L. An advanced computing approach for IoT-botnet detection in industrial Internet of Things. *IEEE Trans. Ind. Inform.* **2022**, *18*, 8298–8306. [CrossRef]
16. Hussain, F.; Abbas, S.G.; Pires, I.M.; Tanveer, S.; Fayyaz, U.U.; Garcia, N.M.; Shah, G.A.; Shahzad, F. A Two-Fold Machine Learning Approach to Prevent and Detect IoT Botnet Attacks. *IEEE Access* **2021**, *9*, 163412–163430. [CrossRef]
17. Desai, M.G.; Shi, Y.; Suo, K. A Hybrid Approach for IoT Botnet Attack Detection. In Proceedings of the 2021 IEEE 12th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), Vancouver, BC, Canada, 27–30 October 2021; pp. 590–592. [CrossRef]
18. Meidan, Y.; Bohadana, M.; Mathov, Y.; Mirsky, Y.; Breitenbacher, D.; Shabtai, A.; Elovici, Y. N-BaIoT: Network-based Detection of IoT Botnet Attacks Using Deep Autoencoders. *IEEE Pervasive Comput.* **2018**, *17*, 12–22. [CrossRef]
19. Nguyen, H.T.; Ngo, Q.D.; Le, V.H. A novel graph-based approach for IoT botnet detection. *Int. J. Inf. Secur.* **2020**, *19*, 567–577. [CrossRef]
20. Dange, S.; Chatterjee, M. IoT Botnet: The Largest Threat to the IoT Network. In *Advances in Intelligent Systems and Computing*; Jain, L., Tsihrintzis, G., Balas, V., Sharma, D., Eds.; Data Communication and Networks; Springer: Singapore, 2020; Volume 1049. [CrossRef]
21. Le, H.-V.; Ngo, Q.-D. V-sandbox for dynamic analysis IoT botnet. *IEEE Access* **2020**, *8*, 145768–145786. [CrossRef]
22. Idriss, H.K. Mirai Botnet in Lebanon. In Proceedings of the 2020 8th International Symposium on Digital Forensics and Security (ISDFS), Beirut, Lebanon, 1–2 June 2020; pp. 1–6. [CrossRef]
23. Shodan. Shodan Search Engine. 2022. Available online: <https://www.shodan.io/> (accessed on 10 October 2022).

24. Gallopeni, G.; Rodrigues, B.; Franco, M.; Stiller, B. A Practical Analysis on Mirai Botnet Traffic. In Proceedings of the 2020 IFIP Networking Conference (Networking), Paris, France, 22–26 June 2020; pp. 667–668.
25. Schiller, E.; Aidoo, A.; Fuhrer, J.; Stahl, J.; Ziörjen, M.; Stiller, B. Landscape of IoT security. *Comput. Sci. Rev.* **2022**, *44*, 100467. [[CrossRef](#)]
26. Kasat, K.; Rani, D.L.; Khan, B.; Ashok, J.; Kirubakaran, M.K.; Malathi, P. A novel security framework for healthcare data obtained by IOT sensors. *Meas. Sens.* **2022**, *24*, 100535. [[CrossRef](#)]
27. Tiwari, R.; Sharma, H.K.; Upadhyay, S.; Sachan, S.; Sharma, A. Automated Parking System-Cloud and IoT based. *Int. J. Eng. Adv. Technol.* **2019**, *8*, 116–123.
28. Hu, Z.; Gnatyuk, S.; Okhrimenko, T.; Tynymbayev, S.; Iavich, M. High-Speed and Secure PRNG for Cryptographic Applications. *Int. J. Comput. Netw. Inf. Secur.* **2020**, *12*, 1–10. [[CrossRef](#)]
29. Hu, Z.; Khokhlachova, Y.; Sydorenko, V.; Opirskyy, I. Method for optimization of information security systems behavior under conditions of influences. *Int. J. Intell. Syst. Appl.* **2017**, *9*, 46. [[CrossRef](#)]