Brazilian Journal of Development

DoS attack detection and prevention in fog-based intelligent environments

Detecção e prevenção de ataques DoS em ambientes inteligentes baseados em nevoeiro

DOI:10.34117/bjdv5n11-089

Recebimento dos originais: 14/10/2019
Aceitação para publicação: 08/11/2019

João Vitor Cardoso
Undergraduate student in Computer Science at Federal University of Santa Catarina
Institution: Federal University of Santa Catarina
Address: Caixa Postal 476, 88040-970, Florianópolis, SC, Brazil
E-mail: joao.vitor.brasill@gmail.com

Hugo Vaz Sampaio
Ph.D. student in Computer Science at Federal University of Santa Catarina
Institution: Federal University of Santa Catarina
Address: Caixa Postal 476, 88040-970, Florianópolis, SC, Brazil
E-mail: hvazsampaio@gmail.com

Cristiano Antonio de Souza
Ph.D. student in Computer Science at Federal University of Santa Catarina
Institution: Federal University of Santa Catarina
Address: Caixa Postal 476, 88040-970, Florianópolis, SC, Brazil
E-mail: cristianoantonio.souza10@gmail.com

Carlos Becker Westphall
Professor in Computer Science at Federal University of Santa Catarina
Institution: Federal University of Santa Catarina
Address: Caixa Postal 476, 88040-970, Florianópolis, SC, Brazil
E-mail: carlosbwestphall@gmail.com

ABSTRACT

The Internet of Things and Fog Computing are technologies currently used in many areas. They can be applied to provide a residential automation environment, for example, fire alarm applications, gas leak alarms, among others. Security-related searches for these fog-based environments are still in the early stages. Also, the fact that these environments are connected to the Internet makes them vulnerable to various threats, such as Denial of Service (DoS) attacks. In this work, we propose a module for detection and prevention of DoS attacks, that operates in the system's fog layer, to protect the system from external attacks. Practical experiments were carried out with the proposed module, considering a Raspberry Pi 3B as our fog server. The results obtained demonstrates that the approach is capable of detecting external attacks, as well as blocking the IPs from attackers, using less than 20% of cpu and less than 1% of RAM memory usage.

Brazilian Journal of Development

Key words: Dos Attack; Fog Computing; Intelligent Environments.

RESUMO

A Internet das Coisas e a Computação por Névoa são tecnologias usadas atualmente em muitas áreas. Eles podem ser aplicados para fornecer um ambiente de automação residencial, por exemplo, aplicações de alarme de incêndio, alarmes de vazamento de gás, entre outros. As pesquisas relacionadas à segurança desses ambientes baseados em neblina ainda estão nos estágios iniciais. Além disso, o fato de esses ambientes estarem conectados à Internet os torna vulneráveis a várias ameaças, como ataques de negação de serviço (DoS). Neste trabalho, propomos um módulo para detecção e prevenção de ataques de negação de serviço, que opera na camada de neblina do sistema, para proteger o sistema contra ataques externos. Experimentos práticos foram realizados com o módulo proposto, considerando um Raspberry Pi 3B como nosso servidor de neblina. Os resultados obtidos demonstram que a abordagem é capaz de detectar ataques externos, além de bloquear os IPs dos invasores, usando menos de 20% da CPU e menos de 1% do uso de memória RAM.

Palavras-Chave: Dos Attack; Computação em nevoeiro; Ambientes inteligentes.

1. INTRODUCTION

Internet of Things (IoT) and Fog Computing can be used in several new applications for multiple areas such as medicine, agriculture, tourism, homes and smart cities, contributing to improvements in survival, and in the quality of life of people. They can also be used in applications to provide a residential automation environment. Example applications include fire alarm applications, gas leak alarm, temperature control, access control, etc.

IoT environments are composed of inexpensive and small devices. These devices usually have resource restrictions. This fact allied to a large amount of data generated by these devices caused the need for greater computational capacity. Cloud computing offers a practical solution to resolve these limitations. However, this centralization of the processing and storage resources entails a large separation between the physical devices and the cloud. This fact may imply problems related to high latency. Fog computing is a distributed computing paradigm that operates in an intermediate layer between the cloud and IoT devices, offering services with lower latency compared to the cloud.

Due to fog being a relatively recent technology, the research related to security in fog and IoT environment is still at an early stage. Because the devices in these environments are connected to the Internet, they are vulnerable to various threats, such as Denial of Service (DoS) attack. The DoS attack happens when the attacker sends many requests to the target machine in an attempt to overload it until it becomes unavailable.

A recent incident precedent involving IoT devices occurred in October 2016. Where an attack against the Dyn service provider has knocked down hundreds of sites, including Twitter, Netflix, Reddit and GitHub, for several hours [Kolias et al. 2017]. This makes special security techniques indispensable in modern computational systems.

DoS attacks on home automation systems can be very dangerous because they handle information and manage equipment that is in people's everyday lives. Having fire alarm systems and gas leakage with a malfunctioning or unavailable may result in tragedies. Therefore, it is of fundamental importance that research is carried out to identify and propose solutions to detect and prevent DoS attacks in fog computing and IoT based home automation environments.

Intrusion Prevention Systems (IPS) are security tools that aim to detect an attack and execute countermeasures. There are several traditional intrusion detection and prevention tools, however, they do not take into account the resource constraints of devices present in home automation environments.

In this work, we propose a module of detection and prevention of DoS attacks that operates in the layer of fog computing to protect the ecosystem of residential automation against external attacks. The proposed module analyzes the TCP/IP protocol packets arriving on the device from the Fog computing layer. In addition to identifying DoS attacks in the context of a smart home, the proposed module predicts the blocking of the attacker's IP. It stores the IP in a table, named blacklist.

The objective of the proposed module, in addition to detecting the attacks of external DoS, is to accomplish this task by spending few resources in relation to traditional tools.

The main contributions of this article are as follows: (i) light module for detection and prevention of external attacks in the Fog computing layer for residential automation environments, through the analysis of TCP/IP protocol packets; (ii) Blocking module of the attackers IP, through blacklist mechanism.

This work is divided into 6 sections. Section 2 presents the main technologies and concepts involved in the scope of this work. In section 3 is contextualized the state of the art related to the detection and prevention of DoS attacks in residential automation environments. Section 4 introduces the autonomic fire detection system and the DoS detection and prevention approach to the proposed fog-computing layer. The experiments for the validation of the proposal are described in Section 5. Also, in section 5, the results obtained are presented and

discussed. Finally, in section 6 the conclusions of this work are presented and some future studies are described.

2. BACKGROUND

This section presents the main fundamental concepts involved in this work.

2.1. INTERNET OF THINGS

In According to [Xia et al. 2012], the Internet of Things (IoT) refers to the interconnectedness of objects used in everyday life, for example, refrigerators, lamps, television, among others, equipped with ubiquitous intelligence. In IoT, the sensors and actuators blend with the environment that surrounds us and the information is shared on platforms to develop a common operating image. IoT connects physical devices to the Internet, makes everyone connected and acting intelligently.

The rapid growth of telecommunication systems, which allows access to the Internet anywhere and at any time, collaborated with the development of IoT applications [Lu and Xu 2019]. IoT emerged from the advances of various areas such as embedded systems, microelectronics, communication, and sensing. IoT is understood as the set of wireless sensor networks (RSSF), Radio Frequency Identification (RFID) devices as well as Vehicular ad-hoc networks (VANETs).

The authors [Gubbi et al. 2013] report that IoT engages a set of paradigms: Object-oriented (middleware), things (sensors) and semantics (knowledge). They define IoT as:

- The worldwide network of interconnected unique address objects according to standard communication protocols.
- The "things" are active participants, information and social processes where they can interact and communicate with each other and with the environment.
- An intelligent environment that uses information and communication technologies to make it more conscious, interactive and efficient.

From a conceptual standpoint, IoT is based on three basic principles related to the characteristics of intelligent objects: being identifiable, communicable and capable of interacting with the environment in which they are inserted [Miorandi et al. 2012]. IoT devices are physical objects that are typically small in size, with limited processing and storage capabilities.

Brazilian Journal of Development

IoT applications have the potential to sensitively improve people's lives, the way they live, work, learn, and have fun. For example, smart houses can provide to the residents, automatic garage opening, coffee preparation, control of HVAC systems, among others [Al-Fuqaha et al. 2015].

The authors [Dragomir et al. 2016] represent a generic IoT system in three main layers. Figure 1 shows the three IoT layers, which are: perception, transport, and application.

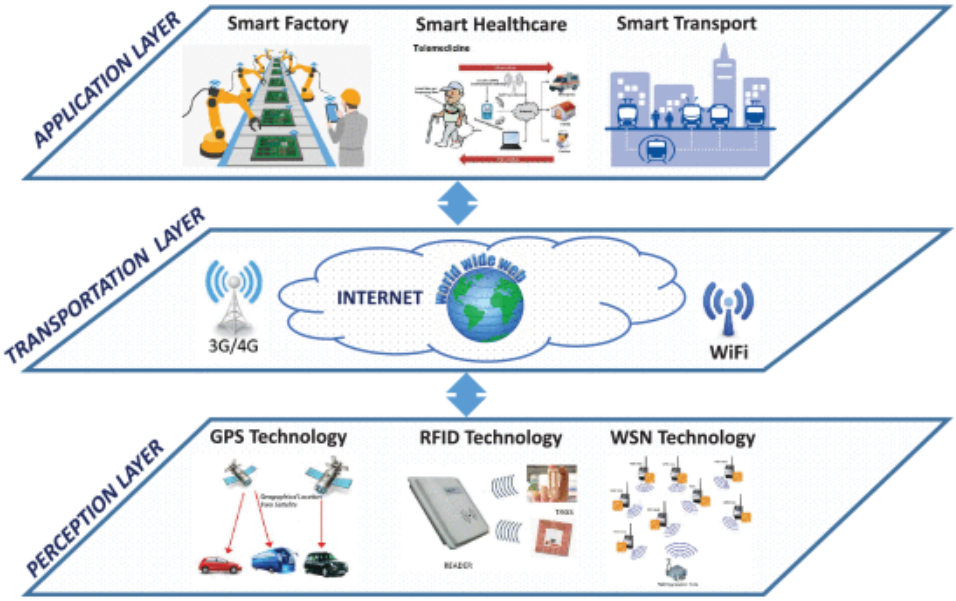

Figure 1. IoT layers. Fonte: [Frustaci et al. 2018]

The perception layer is related to intelligent objects such as sensors and actuators. They have the function of collecting and processing data in different technologies. This layer presents threats in physical attacks, in which the attacker focuses on the system hardware components, denial of service attacks, exploring the low processing capacity of the devices, among others.

The Transport layer provides an environment of access to the perception layer. It aims to transmit information collected from the perception layer to processing systems through different communication technologies, such as Wifi and 3G. This layer introduces threats such as routing attacks, denial of service attacks, and data transmission attacks.

The application layer is responsible for providing services to users. Smart, high-quality services can be implemented to meet the needs of the user, providing a smart environment in IoT. It has as threats the data leakage, customer data theft and denial of service attack.

Brazilian Journal of Development

As mentioned above, IoT devices have resource constraints. This fact allied to a large amount of data generated by these devices caused the need for greater computational capacity. In addition, the amount of devices connected to the Internet continues to grow. Cloud computing offers a practical solution to resolve these limitations.

2.2. CLOUD COMPUTING

[Buyya et al. 2009] Define Cloud Computing as a type of parallel and distributed system, consisting of a collection of interconnected and virtualized computers. These computers are dynamically provisioned and presented as one or more unified computing resources, considering service level agreements (SLA).

The National Institute of Standards and Technology (NIST) defines cloud computing as a model for enabling ubiquitous and convenient network access to a shared set of computing resources. These features are configurable and can be quickly provisioned and released with minimal effort to manage or interact with service providers [Mell et al. 2011]. Therefore, it is a model that provides computer services, hardware, software, and storage resources that can be accessed from any platform anywhere.

The aforementioned basic features of cloud computing make it an important processing resource for IoT applications. Mainly, those composed of a large number of sensors, which deal with large extractions of information from the environments in which they are inserted.

While Cloud Computing is an important ally of IoT applications, the use of cloud computing also has disadvantages, because this centralization of the processing and storage resources leads to a large separation between the devices Physical and cloud. This fact, according to [Satyanarayanan 2015], may imply problems related to high latency.

2.3. FOG COMPUTING

In 2012, Cisco introduced the concept of fog computing. To address the latency obstacles faced by IoT applications that use cloud computing for data access, processing, and storage. Fog computing is a distributed computing paradigm that operates in an intermediate layer between the cloud and IoT devices [Bonomi et al. 2012]. Fog Computing extends the cloud computing paradigm to the edge of the network to provide efficient data access, processing and storage, enabling a new generation of applications and services [Mukherjee et al. 2018].

Brazilian Journal of Development

The NIST Institute [Iorga et al. 2018], defines Fog Computing as a layered model, shown in Figure 2, which enables ubiquitous access to scalable and shared computational resources. Fog consists of fog, virtual or physical nodes, aware of context, that serve applications and services, considering the latency limitations. A fog system can be distributed and organized into clusters.

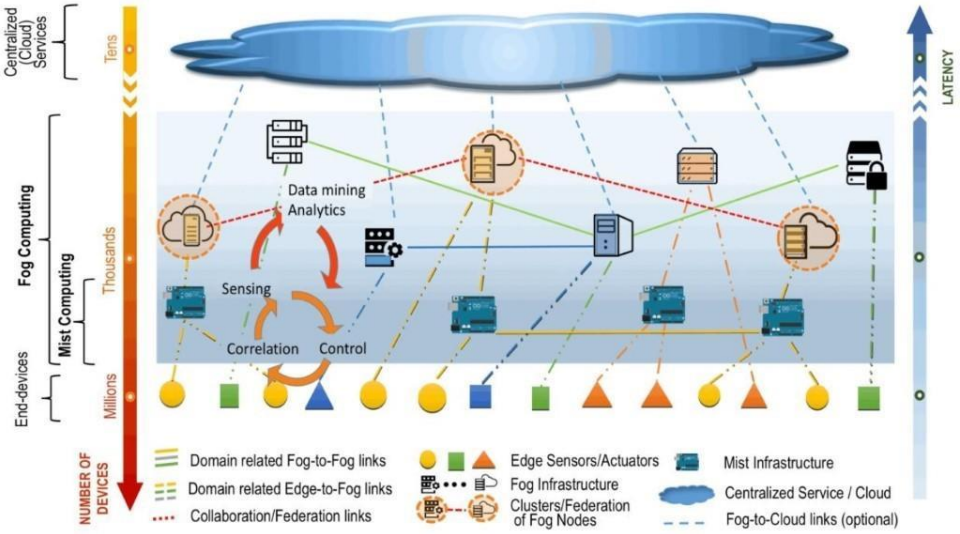

Figure 2. Fog model [Iorga et al. 2018].

For [Alrawais et al. 2017] Fog is an intermediate layer where the nodes, physical or virtual, can be distributed physically close to users. Nodes can be local network servers or gateways, also called the edge of the network. With Fog Computing, large-scale data processing, such as Big Data, coming from IoT devices, is done in a distributed manner, physically closer to the source of information, decreasing the response time to the final device. Fog also supports mobility, location awareness, heterogeneity, and wide scalability. In the lower layer are devices of users such as mobile phones, embedded devices, IoT, etc.

Due to fog being a relatively recent technology, the research related to security in fog and IoT environment is still at an early stage. In the next section are exposed to some security aspects related to fog computing and IoT.

2.4. SECURITY

This section presents concepts related to IoT environment security.

Brazilian Journal of Development

2.4.1 Threats in smart environments

The following are discussed the threats in which IoT systems are susceptible. Table 1 presents the main threats on each tier of an IoT system model.

Table 2. Threats in IoT System Model.

Layer	Main Threats
Application Level	Data Leakage
	DoS Attacks
	Malicious Code Injection
Transportation Level	Routing Attacks
	DoS Attacks
	Data Transit Attacks
Perception Level	Physical Attacks
	Dos Attacks
	Data Transit Attacks

It is possible to identify which denial of service attack is a major threat, appearing in the three layers. The DoS attack happens when the attacker sends many requests to the servers in an attempt to overload the target system until the services are unavailable. The difference between this and a Distributed Denial of Service (DDoS) attack is that instead of a single machine, in DDoS there are several machines performing requests.

The types of DoS and DDoS attacks involved in the theme of this work are SYN Flood, Ping Flood, and UDP Flood.

Basically, in the SYN Flood attack, the attacker explores the operation of the TCP protocol to overwhelm the system. TCP is a reliable protocol used to establish secure connections between client and server. It is used in different services, such as Web servers, e-mail, file transfer, among others. This protocol uses the Three-Way Handshake mechanism to

establish the connection between client and server before sending the data. In this mechanism, the client first sends a packet with the SYN message to the server. The server responds with the message SYN + ACK and the client sends ACK messages to confirm the establishment of the connection. In the SYN Flood attack, the attacker sends several SYN messages and does not send the ACK message to terminate the connection establishment. For each SYN message, the server creates a record in a table and waits for the client's ACK message.

The attacker continues to send SYN messages until the server table becomes full, so the server will not be able to establish new connections with the clients, leaving the service unavailable, as illustrated in Figure 3 [Wang et al. 2002].

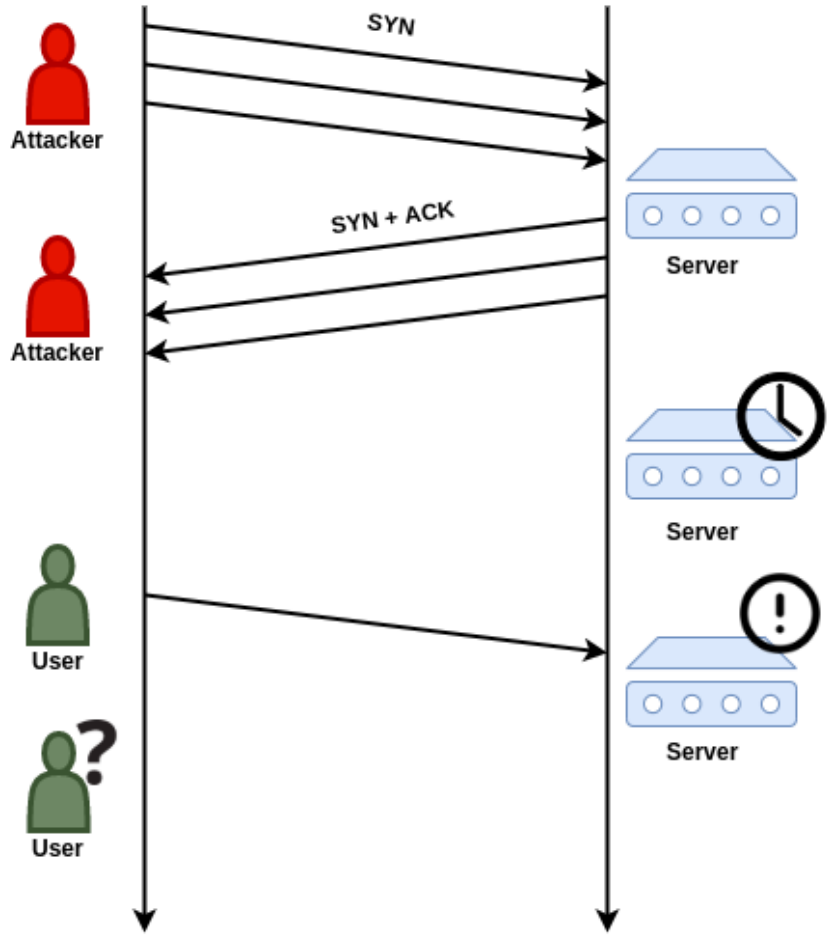

Figure 3. SYN Flood attack.

The Ping Flood attack happens when the attacker sends several echo-Request messages from the ICMP protocol. These messages are intended to test connectivity between devices, for example, when a user uses the ping command on their machine. In this attack, the server is wasting time and resources to process these messages [Nazario 2008].

Finally, in the UDP Flood attack, the attacker sends several packets with UDP datagram to the servers. The UDP protocol does not need to make any connection establishment before sending the data. Also, it does not guarantee that the data arrives correctly to the destination, so it is an untrusted protocol. This protocol has little control information, making it smaller and faster, so it is widely used in real-time applications and also in the communication between IoT objects [Cabrera et al. 2001].

The following are the concepts related to IDS, an important security mechanism.

2.4.2. Intrusion Detection and Prevention

The first step in securing a networked system is to detect the attack, even if it is unable to prevent it, so intrusion detection can be considered the first line of defense in any security system [Kabiri and Ghorbani 2005]. Intrusion Detection Systems (IDS) are security tools [Garcia-Teodoro et al. 2009] that aim to defend a system, executing countermeasures or generating alerts for an entity capable of performing appropriate actions, when an attack occurs [Axelsson 2000].

Depending on the type of analysis performed, IDS can be classified into anomaly or signature detection. In anomaly detection, IDS defines a default behavior and signals all abnormal behavior. In signature detection, IDS compares the monitored actions with signatures previously defined in the system. Signature-based detection tends to be effective only in known attacks, in contrast, anomaly-based detection has the potential to detect intrusive events not previously seen [Axelsson 2000, Garcia-Teodoro et al. 2009].

After identification of the intrusion, the IDS can behave passively, where is informed the event detected to the user responsible, through messages consoles, e-mail, reports, among others. Besides, IDS can take active action by taking proactive and corrective actions on an identified critical event, correcting a system vulnerability, reconfiguring the firewall, among others [Jackson et al. 1999]

An intrusion prevention system (IPS) has the same abilities as an IDS, where it is capable of detecting real-time attacks. However, also, it can isolate invasions and protect the network actively [Chandre et al. 2018, Zhang et al. 2004].

Brazilian Journal of Development

3. RELATED WORKS

This section presents a contextualization regarding the current state of the art related to the safe residential automation environment. To construct a good perspective on the state of the art, a bibliographic review was carried out on the subject.

In the review carried out, studies published in English were researched in the period 2009 to 2019, available online in the following databases: IEEEXplore, ACM Digital Library, Elsevier Science Direct, and Springer Link.

The following is an overview of intrusion detection in home automation environments. Also, the related works are exposed, obtained through the review, as well as their main contributions, the approaches used and the deficiencies observed.

Smart houses are becoming a trend in IoT. In the authors ' article [Vikram et al. 2017], a low-cost residential automation system based on wireless sensor networks incorporating IoT was developed. The system has temperature and humidity sensors, gas leakage system, fire alarm, intrusion alarm, rain detection, among others. In addition, it has an application for Android devices to provide a user management interface. As mentioned, this application handles a lot of information captured from the environment, however, it does not handle any security issues. Protecting home automation applications from denial of service attacks is extremely important, as disrupting critical systems such as fire and gas leak detectors can be extremely dangerous.

The authors [Kasinathan et al. 2013] Propose an architecture to protect from DoS attacks within IoT environments. In the proposed architecture is used the traditional open-source IDS called Suricata. The approach analyzes network-level traffic and signature attacks to identify DoS attacks. However, using a traditional IDS that is not designed for the purpose of operating in the IoT context can impair the performance of the IoT application.

The method proposed by the authors [Maphats ' OE and Masinde 2016] seeks to identify DoS attacks originated in the internal network of sensors. In this type of attack, the malicious nodes can change from Id to pass a new node and flood the gateway with registration packets. The proposed method records the frequency that each node changes from Id to register at the gateway. It utilizes fuzzy logic to decide if the frequency is high enough to trigger an alarm. As mentioned, this work seeks to detect internal attacks, our work focuses on blocking attacks caused by external attackers.

The authors [Nikam and Ambawade 2018] have proposed the intrusion detection mechanism based on counterattacks malicious opinion metrics to the routing protocol in the

IoT network. This mechanism uses opinion metrics to calculate average belief values for each node in the network and to detect the malicious node based on this value in a favorable way to energy. This work is therefore to detect denial-of-service attacks caused by internal malicious devices on the network.

The proposal of the Intrusion Detection and Prevention System (IDPS) based on fog computing for IoT networks, by the authors [Shafi et al. 2018], employs the machine learning classifiers: Recurrent Neural Network (RNN), Perceptron Multi-layer (MLP) and Tree of Alternative decision (ADT), they constitute the multi-classifier. The multi-classifier is capable of detecting DoS threats. In this work, experiments were carried out with the database UNSW-NB15. However, no attacks were performed against the system. Moreover, our method has as a differential the fact that it aims to be light in relation to the other traditional approaches.

Finally, in their work [Aldaej 2019], a preventive approach is proposed to increase the cybersecurity of IoT devices and networks against bandwidth-consuming DDoS attacks on IoT devices. This work assumes that there is already an IDS operating on the network and only works with the logs generated to implement countermeasures.

Based on the current state of the art related to intrusion detection and prevention for residential automation environments, we propose a module for detecting and preventing denial-of-service attacks. The proposed module operates in the Fog computing layer for residential automation environments. In the next section, we describe more details about it.

4. PROPOSED APPROACH

In the context of smart houses, a fire alarm system was proposed by [Sampaio et al. 2019], using an IoT device with a battery that is managed in the FOG layer. The IoT node is responsible for capturing the data for the environment in which it is located, and the fog device performs data processing and storage. The IoT node used the Arduíno Uno microcontroller as the main hardware and three sensors, being they a temperature and humidity sensor of the air, a gas and smoke sensor MQ-2, and a flame sensor. IoT and fog devices also have a Zigbee antenna module for wireless communication.

The interface of the fire alarm system of [Sampaio et al. 2019] is shown in Figure 4. In addition to displaying sensor values, the system displays a message about the current system state, as well as the device's battery estimation. The IoT node has a working cycle, so we can manage the same hibernation time, between the data capture intervals, for energy saving. In

the Sleep Time field, we can indicate an integer value that represents the duration of the node's hibernation, in seconds. The alarm field is used to identify whether the alarm is active or not.

Figure 4. Fire alarm fog system. Adapted from [Sampaio et al. 2019].

When modifying the values of the alarm or sleep time fields, a Zigbee protocol message is generated and forwarded to the Zigbee antenna connected to the serial port. The antenna will then send the Zigbee package to the IoT device.

The interface of the fire system can be accessed via the Web through smartphones, computers, etc. This communication occurs through requests to the server hosted on the Fog device,

The Fog node used was a Raspberry Pi 3, which acts as the system server, with database server services and the Web server. Figure 5 introduces the system's architecture. The Fog node has a Zigbee antenna module plugged into a Serial port. It has an application, called Central IoT (CIoT), which manages the different services, as well as communication with the Serial ports. The fog node reads the packets received by the Zigbee network, extracting the information and storing it in a database and displaying the information on a Web site.

Brazilian Journal of Development

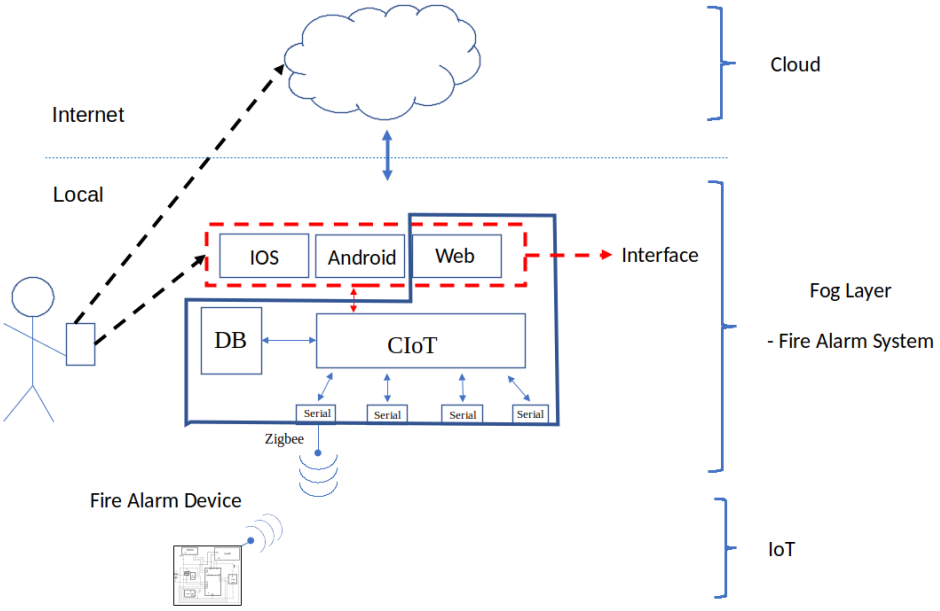

Figure 5. Example of Fire alarm fog system.

We conducted a security analysis of the system of [Sampaio et al. 2019] To verify the weaknesses that may preclude its correct functioning. A possible point of system threats are the requests to the Web server hosted on the fog device. Multiple requests can be made external to the system's network, such as malicious devices, with aim of overloading the fog's resources, delaying the services, resulting in low QoS. Therefore, it is important to protect the fog layer devices against external attacks, because the interruption of the services of this manager leaves the entire system nonfunctioning.

Finally, in this article we propose an approach of detection and prevention of external attacks, a light approach, in order to use few resources, to protect this environment from residential automation. The light proposal for detecting and preventing intrusion of external attacks is presented below.

4.1. DOS DETECTION AND PREVENTION MODULE

As mentioned above, the denial of service attack is a fairly common type of threat. In this work, parameters were implemented to try identifying this type of attack for intelligent homes context. In addition to detecting the attack, the proposed approach predicts the blocking of the attackers' IP through the system firewall. It stores the IP's in a blocked IP table, named blacklist.

Brazilian Journal of Development

Considering the context of smart houses, as well as the security challenges and threats presented, it was developed in this work an IDS to analyze all TCP/IP protocol packets from the different interfaces of the fog device. In a way, to allow control of the input stream of external source data, as well as assist in the management of IoT devices that communicate through this protocol.

Figure 6 presents an illustration of the location of the detection and prevention module proposed in the context of the residential automation system of [Sampaio et al. 2019].

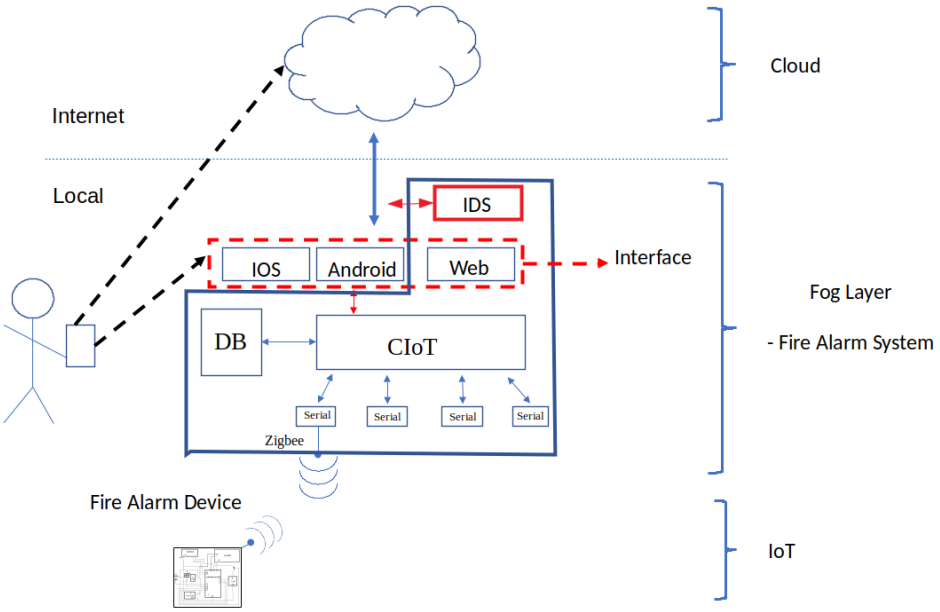

Figure 6. Illustration of the proposed module in the fire alarm system architecture.

The package analyzer was developed in the C++ language based on the Libtins library. Libtins is an open-source library developed by Matias Fontanini. It aims to provide the developer with a way, efficiently, regardless of the platform, send, receive, and manipulate network packets. This library was used to get the network packets received on the device.

As described, denial of service attack is a type of threat that appears in the three layers of a generic IoT system. So in this software were implemented parameters to try to identify this type of attack for the intelligent home context. The types of attacks that the approach identifies are SYN Flood, Ping Flood, and UDP Flood.

To identify the SYN Flood attack the developed Packet Analyzer registers the source IP address in a temporary table and performs monitoring of the number of SYN packets received over IP. The IP is in this table until you answer packets with ACK. The software is configured as a maximum limit of packets per time interval to be considered an attack. Thus, an attacker will be blocked if it reaches the imposed limit of SYN packets sent. The IP's of the attackers are blocked in the system firewall and stored in a table containing the blocked IP, called blacklist. The temporary table is cleared every time the user-defined time interval ends, because with this it is possible to block the IP's that attempt to send a large number of SYN packets in a small interval of time, signaling an attack.

The module developed to detect Ping Flood attacks limits the amount of echo-Request messages that each IP can send in a given time interval. If the IP reaches the tax limit of these sent messages, it is blocked by the firewall and inserted into the blacklist table.

Finally, the detection module limits the number of packets with UDP datagram received from each IP in a given time interval, blocking the IP that reaches the proposed limit. A maximum packet size can be set, in bytes, that can be received. Thus, if the size of the packets received extrapolates the established limit, the packet may be discarded and the IP blocked. With this, it is also possible to detect attacks of the type of letter soup where the attacker sends large packets filled with random texts.

The detection module has a parameter to set the maximum time of the parse interval. At this interval, the program performs all the operations mentioned and when this interval is over, all IPs contained in the blacklist (IPs blocked) are saved and recorded in a log file also containing the date and time the file was saved. If no IP is blocked in this time interval, the log file is not generated. When the interval is over, the temporary tables are also restarted and a new analysis cycle starts.

In order not to block authorized IPs of origin packets that frequently communicate with the system, a white list is proposed where the devices that should not be blocked are registered.

In the next section, the experiments carried out to evaluate the performance of the proposed approach are described.

5. Evaluation

This section describes the experiments carried out to evaluate the performance of the proposed approach. It was considered the context of smart houses and was used the Raspberry-PI 3B as

a fog device, because of its size and high processing power. In it was implemented the module of detection and prevention of denial of service attacks developed. Attempts were made of DoS and DDoS attacks to verify the functioning of the approach and the resources used when being processed. The check was made on the Wi-Fi network interface of the Raspberry-Pi 3B, named Wlan0.

When the module is put into execution a message appears in the terminal signaling that the monitoring has been initialized. From that all packets received, in the monitored interface, are analyzed by the program.

Stress tests have been performed to simulate DoS and DDoS denial of service attacks to find out how much more resource is used to process each of the three attack types.

Figure 7 presents an illustration of the experimental environment used. The proposed denial of service attack detection and Prevention module operated on the Raspberry-PI 3B device, which consists of the fog layer.

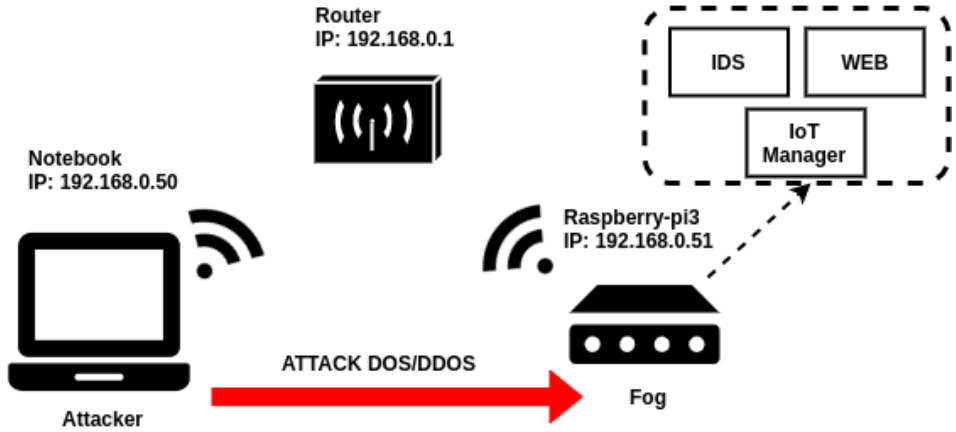

Figure 7. Illustration of the experimental environment.

As can be observed, an attacking machine was used, where the HPING3 program was used to perform the attacks. Hping3 is a tool for sending custom TCP/IP packets and displaying target response. With it, you can manipulate the fragmentation, size, and body of arbitrary packets and transfer encapsulated files with supported protocols. To perform DoS and DDoS attacks, the following commands were used.

- SYN Flood:
    - (1) hping3 --flood --syn -a 1.2.3.4 -p 22 192.168.0.51
    - (2) hping3 --flood --syn --rand-source -p 22 192.168.0.51

- Ping Flood:
    - (3) hping3 --flood --icmp -a 1.2.3.4 192.168.0.51
    - (4) hping3 --flood --icmp --rand-source 1.2.3.4 192.168.0.51
- UDP Flood:
    - (5) hping3 --flood --udp -a 1.2.3.4 -p 22 192.168.0.51
    - (6) hping3 --flood --udp --rand-source -p 22 192.168.0.51

In the commands using the --flood option specifies that packets will be sent as soon as possible. The --SYN option Specifies that packets will be from the TCP protocol with the syn flag Active. The -a 1.2.3.4 OPTION Specifies the source IP of the package. The -P 22 192.168.0.50 option specifies the port number followed by the destination host address, which in this case is the Raspberry-pi. This command (1) specifies that it will be sending multiple packets of type SYN to the host with IP 192.168.0.51 and the source of the package is 1.2.3.4. At the command (3), the option -icmp specific that will send an echo-request message of the ICMP protocol. In the command (5), the option --UDP specific that will be sent a UDP datagram. The --Rand-source option serves to send packets with IP's of random origin. This option was used in the commands (2), (4), (6) to simulate a DDoS attack of each type of attack.

The results obtained with the execution of the experiments are presented below.

While there are no network packets to be analyzed, the program uses less than 0.1% of the processor and 0.8% of memory.

In the stress tests performed, it was possible to simulate DoS and DDoS denial of service attacks to find out how much more resource is used to process each of the three types of attack.

Table 2 shows the test results with the proposed detection and prevention module for each type of attack. Each test was approximately 1 minute long. The first column of the table specifies the type of attack. In sequence the number of the command used, the use (on average) of the processor and RAM and the numbers of blocked IP's.

Table 2. Features Used by the Developed Packets Analyzer

Attack	Command	CPU (%)	RAM (%)	Number of blocked IP
SYN Flood (DoS)	**1**	**6**	**0.9**	**1**
SYN Flood (DDoS)	**2**	**15**	**0.9**	**1485**
Ping Flood (DoS)	**3**	**4**	**0.8**	**1**
Ping Flood (DDoS)	**4**	**18.5**	**0.9**	**2816**
UDP Flood (DoS)	**5**	**5.5**	**0.9**	**1**
UDP Flood (DDoS)	**6**	**19**	**0.9**	**1980**

In table 2 It is possible to note that DoS type attacks have only one blocked IP address because a single machine is an attacker, and they spend between 4% and 6% CPU for packet processing. Once the attack is identified, the attacker's address is blocked by the firewall and the following packets are discarded. DDoS-type attacks spend on average between 15% and 19% of processing since each packet arriving is a new address to be verified and registered, this causes blacklist to have linear growth in relation to the number of packets that arrive at the fog device.

It is important to report that only blocking the packets in the firewall of the Raspberry-PI3 is not enough to combat the attack, as there is still a large stream of packets flooding the network and arriving in the Fog device. This can lead to a lot of loss of user packages that are using the system. Therefore it is recommended to take the records of the IPs recorded in the log and block them on the router, before entering the local network, to avoid a flood in the network.

6. Conclusion and future work

This work presented a module of detection and prevention of external attacks for environments based on IoT and fog computing. The solution falls within the context of residential

automation systems. The module analyzes the TCP/IP protocol packets arriving on the device from the Fog computing layer and can identify DoS attacks in the smart home context. Also, the module predicts the blocking of the attacker's IP. The proposed approach presented relevant results, being able to detect and block DoS attacks in environments based on IoT and fog computing.

As future works, we can cite the proposal of a detection module capable of analyzing the data sent by the sensor nodes to the serial ports of the fog device. The goal is to detect DoS attacks caused by internal devices.

7. Acknowledgements

The authors wish to sincerely thank the Federal University of Santa Catarina (UFSC). In addition, the work was financed in parts by: Foundation for Research and Innovation of the state of Santa Catarina (FAPESC), coordination of improvement of higher education personnel-Brazil (CAPES)-Finance Code 001 and National Development Council Scientific and technological (CNPq).

References

Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., and Ayyash, M. (2015).Internet of things: A survey on enabling technologies, protocols, and applications.*IEEE Communications Surveys & Tutorials*, 17(4):2347–2376.

Aldaej, A. (2019). Enhancing cyber security in modern internet of things (iot) using intrusion prevention algorithm for iot (ipai). *IEEE Access*, pages 1–1.

Alrawais, A., Alhothaily, A., Hu, C., and Cheng, X. (2017). Fog Computing for the Internet of Things: Security and Privacy Issues. *IEEE Internet Computing*, 21(2):34–42.

Axelsson, S. (2000). Intrusion detection systems: A survey and taxonomy. Technical report, Technical report.

Bonomi, F., Milito, R., Zhu, J., and Addepalli, S. (2012). Fog computing and its role in the internet of things. In *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, pages 13–16. ACM.

Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J., and Brandic, I. (2009). Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation Computer Systems*, 25(6):599–616.

Cabrera, J. B., Lewis, L., Qin, X., Lee, W., Prasanth, R. K., Ravichandran, B., and Mehra,R. K. (2001). Proactive detection of distributed denial of service attacks using mib traffic variables-a feasibility study. In *2001 IEEE/IFIP International Symposium on Integrated Network Management Proceedings. Integrated Network Management VII. Integrated Management Strategies for the New Millennium (Cat. No. 01EX470)*, pages609–622. IEEE.

Chandre, P. R., Mahalle, P. N., and Shinde, G. R. (2018). Machine learning based novel approach for intrusion detection and prevention system: A tool based verification. In *2018 IEEE Global Conference on Wireless Computing and Networking (GCWCN)*, pages 135–140. IEEE.

Dragomir, D., Gheorghe, L., Costea, S., and Radovici, A. (2016). A survey on secure communication protocols for iot systems. *In 2016 International Workshop on Secure Internet of Things (SIoT)*, pages 47–62.

Frustaci, M., Pace, P., Aloi, G., and Fortino, G. (2018). Evaluating critical security issues of the iot world: Present and future challenges. *IEEE Internet of Things Journal*, 5(4):2483–2495.

Garcia-Teodoro, P., Diaz-Verdejo, J., Maciá-Fernández, G., and Vázquez, E. (2009). Anomaly-based network intrusion detection: *Techniques, systems and challenges. computers & security*, 28(1-2):18–28.

Gubbi, J., Buyya, R., Marusic, S., and Palaniswami, M. (2013). Internet of things (iot): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29(7):1645 – 1660.

Iorga, M., Feldman, L., Barton, R., Martin, M. J., Goren, N. S., and Mahmoudi, C. (2018). Fog computing conceptual model. Technical report.

Jackson, K. A. et al. (1999). Intrusion detection system (ids) product survey. *Los Alamos National Laboratory*.

Kabiri, P. and Ghorbani, A. A. (2005). Research on intrusion detection and response: *A survey. IJ Network Security*, 1(2):84–102.

Kasinathan, P., Pastrone, C., Spirito, M. A., and Vinkovits, M. (2013). Denial-of-service detection in 6lowpan based internet of things. In 2013 *IEEE 9th International Conference*

on Wireless and Mobile Computing, Networking and Communications (WiMob), pages 600–607.

Kolias, C., Kambourakis, G., Stavrou, A., and Voas, J. (2017). Ddos in the iot: Mirai and other botnets. *Computer*, 50(7):80–84.

Lu, Y. and Xu, L. D. (2019). Internet of things (iot) cybersecurity research: A review of current research topics. *IEEE Internet of Things Journal*, 6(2):2103–2115.

Maphats'oe, T. and Masinde, M. (2016). A security algorithm for wireless sensor networks in the internet of things paradigm. In 2016 *IST-Africa Week Conference*, pages 1–10.

Mell, P., Grance, T., et al. (2011). The nist definition of cloud computing.

Miorandi, D., Sicari, S., De Pellegrini, F., and Chlamtac, I. (2012). Internet of things: Vision, applications and research challenges. *Ad hoc networks*, 10(7):1497–1516.

Mukherjee, M., Shu, L., and Wang, D. (2018). Survey of fog computing: Fundamental, network applications, and research challenges. *IEEE Communications Surveys Tutorials*, 20(3):1826–1857.

Nazario, J. (2008). Ddos attack evolution. *Network Security*, 2008(7):7–10.

Nikam, A. and Ambawade, D. (2018). Opinion metric based intrusion detection mechanism for rpl protocol in iot. In *3rd International Conference for Convergence in Technology (I2CT)*, pages 1–6.

Sampaio, H. V., de Jesus, A. L. C., do Nascimento Boing, R., and Westphall, C. B. (2019). Autonomic IoT Battery Management with Fog Computing. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11484 LNCS(Cc):89–103.

Satyanarayanan, M. (2015). A brief history of cloud offload: A personal journey from odyssey through cyber foraging to cloudlets. *GetMobile: Mobile Computing and Communications*, 18(4):19–23.

Shafi, Q., Basit, A., Qaisar, S., Koay, A., and Welch, I. (2018). Fog-assisted sdn controlled framework for enduring anomaly detection in an iot network. *IEEE Access*, PP:1–1.

Vikram, N., Harish, K. S., Nihaal, M. S., Umesh, R., Shetty, A., and Kumar, A. (2017). A low cost home automation system using wi-fi based wireless sensor network incorporating

internet of things (iot). In *2017 IEEE 7th International Advance Computing Conference (IACC)*, pages 174–178.

Wang, H., Zhang, D., and Shin, K. G. (2002). Detecting syn flooding attacks. In *Proceedings. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 3, pages 1530–1539. IEEE.

Xia, F., Yang, L., Wang, L., and Vinel, A. (2012). *Internet of things. International Journal of Communication Systems*, 25.

Zhang, X., Li, C., and Zheng, W. (2004). Intrusion prevention system design. In *The Fourth International Conference on Computer and Information Technology, 2004. CIT'04.,* pages 386–390. IEEE.