# How to Improve the IoT Security Implementing IDS/IPS Tool using Raspberry Pi 3B+

Ruíz-Lagunas Juan Jesús[1]
Departamento de Sistemas y Computación, TecNM/I.T. Morelia and Universidad Vasco de Quiroga
Morelia, México

Reyes-Gutiérrez Mauricio René[3]
Facultad de Ingeniería Eléctrica-UMSNH/Departamento de Sistemas y Computación TecNM/I.T. Morelia
Morelia, México

Antolino-Hernández Anastacio[2], Torres-Millarez Cristhian[5], Paniagua-Villagómez Omar[6]
Departamento de Sistemas y Computación
TecNM/I.T. Morelia, Morelia, México

Ferreira-Medina Heberto[4]
Unidad de TICs, IIES-UNAM and Departamento de Sistemas y Computación, TecNM/I.T. Morelia
Morelia, México

***Abstract*—This work shows a methodology of implementation and testing of the system is proposed and tested with a prototype; it is constructed with sensors and actuators that allow monitoring the behavior of the system in an environment under threats. We used an IDS / IPS as a protection tool for IoT systems, based on Raspberry Pi and Raspbian operating system. It is described in a block diagram the testing method used. We implemented the IDS/IPS Snort tool in an embedded platform Raspberry. It presents also the state of the art of cloud frameworks that have the same objective of protecting. The main contribution is the implemented testing method for Snort that can be used with security rules in other applications of embedded IoT devices.**

***Keywords*—Security IoT; IDS/IPS software; Pentesting tools; smart cities; prototype Raspberry***

## I. Introduction

Today, Information Technology (IT) is immersed in the use and exploitation of telecommunications networks, to which more devices are connecting every day to communicate with each other (Peer to Peer) and to a central device (client-server). Internet of Things (IoT) offers wide range state of the art solutions, using sensors and/or devices, which monitor to control certain events, giving rise to new challenges to IT security, since information gathered can be compromised by this variety of components. Internet of Things (IoT) is a concept defined by Kevin Ashton in 1999, which describes a network that connects people and objects [1]. These objects, right now, allow computers to have "sensors", which facilitate them, not only to process information but gather more information through these devices, allowing applications to be even more "intelligent", since it is possible to make decisions in real-time, based on a major quantity of information. It is possible to assure that since the first implementation of IoT up until date there are millions of sensors interconnected.

The concept of "smart" devices is inherent in connectivity to obtain benefits from the information [2]. Industry automation, and monitoring systems are the main reasons for this technology's success, so data networks are being unified with production networks to achieve these benefits. It has been estimated that in 2018 there were more than 7 billion IoT devices [3]. It is estimated that by 2020 there will be more than 10 billion and more than 22 billion by 2025. This worldwide trend is due to the growing demand to connect devices to the networks.

With this trend, has been observed clearly that the way of the information interchange with technology will change, but at what cost? Due to the demand for interconnection, many IoT developers do not consider the security in communication for many reasons: Amongst these, we have processing costs, training and algorithm implementation.

IoT devices are considered to have many weaknesses in information security since their development. The following are examples described below:

- Passwords stored in plain text.
- Outdated firmware and not encryption.
- Video streaming without encryption.
- Communication between devices and servers in plain text.
- Over-shared data (influence of cloud utilization).
- Development bugs in the firmware.
- Use of default passwords.
- Devices have a direct interface to the internal network, but they can be connected to the Internet, making increase an attack risk exponentially.

Sponsors TecNM/I.T. Morelia and IIES-UNAM, México

In addition, it has known that hackers are exploiting these vulnerabilities with current tools and techniques of their own to achieve that goal. One of the most recent tools to detect vulnerabilities in IoT devices is *Autosploit* [4], since it uses artificial intelligence in its algorithms [5].

The main contribution of this paper is to present a testing method for IDS/IPS and the comparison of its response implemented on the Raspberry platform to Nmap and Metasploit of network attacks. The paper is organized as follows: Section II deals with actual security in IoT, in Section III describes IDS/IPS tools, in Section IV describes the methodology, Section V shows the results about this work, and finally, in Section VI shows conclusions.

## II. Security in IoT

As a response to exposed security problems, the main software developers propose various strategies to guarantee security in interconnection components. According to [6] the expansion of IoT, it has been developing in the following areas, see Table I.

TABLE. I. IoT Expansion Areas based on [6]

| **IoT development areas** | **Description** | | |
|---|---|---|---|
| | ***Elements*** | ***Opportunities*** | ***Challenges*** |
| Smart life | - Health care.<br>- Consumer and Retail businesses.<br>- Bank Convergence.<br>- Security.<br>- Public services | Technologies that promote simplifications in the lives of the users. | Ensure secure information and data exchange. |
| Smart mobility | - Intelligent vehicles<br>- Urban mobility.<br>- Intercity mobility.<br>- Rate management and payment solutions.<br>- Distribution and logistics.<br>-Fleet management. | Real-time solutions that make mobility simpler and transport reliable. | Secure interconnection and secure real-time monitoring and activation. |
| Smart Cities | - Intelligent infrastructure management.<br>- Cross-agency collaboration using the cloud.<br>- Data collection in real time and quickly.<br>- Better planning of cities.<br>-Network utilities.<br>-Construction Development. | The innovations will aim to improve the quality of life in the city. Using sensors and systems that help in decision-making. | Ensure that users exchange information in real time and keep their data protected against hackers. |
| Smart manufacturing | - Machine learning.<br>-Communication between machines.<br>- Network interconnection.<br>-Optimization of processes.<br>-Proactive asset management.<br>- Improve infrastructure integration. | Smart solutions to optimize production processes, controls and quality. | Keep process information safe, interconnections between machines using secure protocols. |

## III. Frameworks for Security in IoT

### A. Frameworks in the Cloud

According to [7] the challenge in IoT, security will be to build services that can be integrated into different software solutions. In platforms described in [8], [9], [10], [11], [12] and [13] a set of services oriented to software and hardware solutions to offer layers of security at different levels, working from the platform (**PaaS**, Platform as a Service), and services (**SaaS**, Software as a Service), which are offered with auto-service, cost and on-demand schemes. Table II shows the main features of these proposals that are a reference as a framework and cloud computing.

The list of platforms shown in Table III is compared by the type of framework and the security mechanisms they offer with encryption method, in a mobile application (App) or any API.

Despite the implementation of different security schemes both in the device, in the interconnection and in the cloud, these do not guarantee that the IoT components are free from attacks using the different layers of the communication protocols [20].

TABLE. II. MAIN FRAMEWORKS IN THE CLOUD THAT OFFER SECURITY FOR IOT

| **Framework** | Description | Elements | Security |
|---|---|---|---|
| Amazon Web Services IoT [8] | It offers a cloud platform, as well as IoT hardware, operating system, software and cloud connectivity services, security layers, monitoring, and administration software for PaaS services. | - Software for devices.<br>- Service control.<br>- Data services.<br>- Secure interconnection services. | Oriented to platforms with clients and servers using secure interconnection mechanisms, it offers cryptography and monitoring services. |
| Microsoft Azure IoT Hub [9] | They offer a cloud platform with open and flexible services to connect securely, monitor and manage IoT devices and develop applications using open source SDK (Development Kit) and multiple protocols. Working under a SaaS scheme. | - Device layer.<br>-Interconnection layer.<br>- Cloud access layer.<br>- Hub layer.<br>- Back-End for Apps. | It offers an exchange of information with the devices, using languages such as NodeJS, .Net, Java, Python, Android, IOS and C. It establishes layers of security for the connection. |
| Oracle Internet of Things Cloud Services [10] | It offers a PaaS scheme, which allows you to connect IoT devices to the cloud, analyze data in real time to integrate it into business applications, allows you to establish web services and any other Oracle proprietary service. | - Software for devices.<br>- Hub for access to cloud services.<br>- Offers a wide range of SaaS. | It offers connectivity with IOS, Android and any device that uses Java, Posix C, and the RESTful protocol offers a cryptography scheme. |
| Watson IoT Platform [11] | It offers a PaaS-based connectivity scheme, offers a firmware to be installed on different platforms and achieve connectivity and the use of services. | - Use of IBM cloud.<br>- Device management.<br>-Platform services.<br>- Administration services.<br>-Blockchain services. | Uses protocols for secure interconnection using a gateway, uses node.js, java, and JS languages. Offers blockchain and crypto services. |
| Xively, Google [12] | Google Cloud IoT offers a set of tools to, connect, process, store and analyze data both in the perimeter and in the cloud, they are PaaS services. | - Software for sensors<br>- Cloud Connection (Edge).<br>- Android, CPU, GPU and TPU support.<br>- Offers real-time analysis service.<br>-Data usage services. | It offers scalable and managed services, integrates Artificial Intelligence functions, and uses a communication protocol with security schemes (MQTT, Machine-to-Machine protocol). |
| Samsung Artik [13] | It offers connectivity services based on PaaS, to connected smart devices, as well as connected homes and smart cities, allows connectivity under the concept known as D3 (Data Driven Development). | - Software for sensors and devices.<br>- Storage services and location of services (brokerage).<br>- Support for third-party users and Apps. | It offers Big Data usage scheme, it uses REST, Web Sockets, MQTT and CoAP protocols (Protocol for restricted devices) to exchange information. |

TABLE. III. PLATFORM COMPARISON FOR IOT

| No. | Framework | Firmware owner | Security mechanism |
|---|---|---|---|
| 1 | Amazon Web Services IoT | Yes | Encryption |
| 2 | Azure IoT Hub | No | In app |
| 3 | Oracle IoT Cloud Services | No | In app |
| 4 | Watson IoT Platform | Yes | Blockchain |
| 5 | Xively | No | In app |
| 6 | Samsung Artik | Si | Encryption |
| 7 | Carriots [14] | Si | API keys |
| 8 | Adafruit.io [15] | No | In app |
| 9 | Ubidots [16] | No | In app |
| 10 | MyDevices Cayenne [17] | No | In app |
| 11 | Macchina IO [18] | No | In app |
| 12 | ThingSpeak [19] | No | In app |
| 13 | Arduino IoT Cloud [20] | Si | Encryption |

## IV. Intrusion Prevention and Detection Systems (IDS/IPS)

The IDS and IPS are complements to improve security on host systems (HIDS/IPS), mainly for embedded IoT devices, to establish which of the current tools is most suitable for IoT, a comparison was made between the main open source IDS/IPS offered on the Internet. It is based on the sum of the indicators achieved (functionality, usability, reliability, performance, supportability), each indicator contributes 5 points; 0 doesn't accomplish, 1 is mentioned, 2 slightly accomplished, 3 accomplished, 4 very accomplished, 5 Extremely accomplished. Fig. 1 shows this comparison adding the indicators achieved for the following tools IDS and/or IPS: Snort [21], Suricata [22], Broids [23], OSSec [24], OS Tripwire [25], Aide [26], Samhain [27], Fail2ban [28], Sagan [29].

As we can observe Snort and Sagan, are the best tools evaluated.

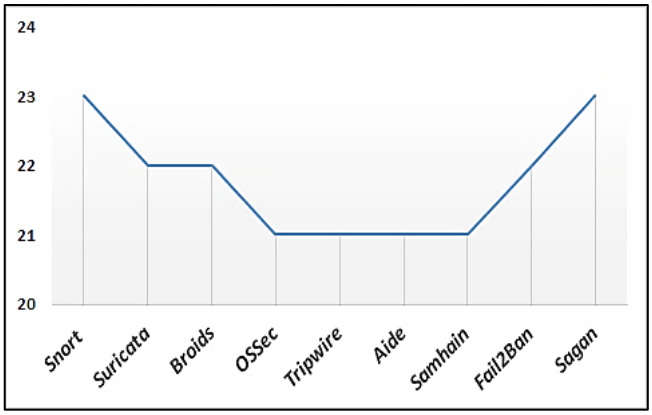



Fig. 1. Comparison between the main IDS/IPS using FURPS.

## V. Methodology

The research methodology used is based on an experimental and applied method, therefore the process compromises several steps, which they are described in next lines. The implementation of an IDS/IPS as a security scheme on a Raspberry Pi3B+ card, is a relatively simple process, however, it is necessary to evaluate the operation of the system, to develop adequate detection rules using Snort and Sagan, to improve the embedded system in the management and monitoring the network traffic and the internal state of the device.

Fig. 2 shows the methodology in block diagram used to design the prototype and pentesting probes. The methodology proposes a reviewing of the state of the art in IoT security context, and related or similar projects, then a prototype is implemented with the IDS/IPS tool installed for pentesting and monitoring threat behavior in these devices. At the end, feedback is proposed to improve the prototype's components and software tools for security.

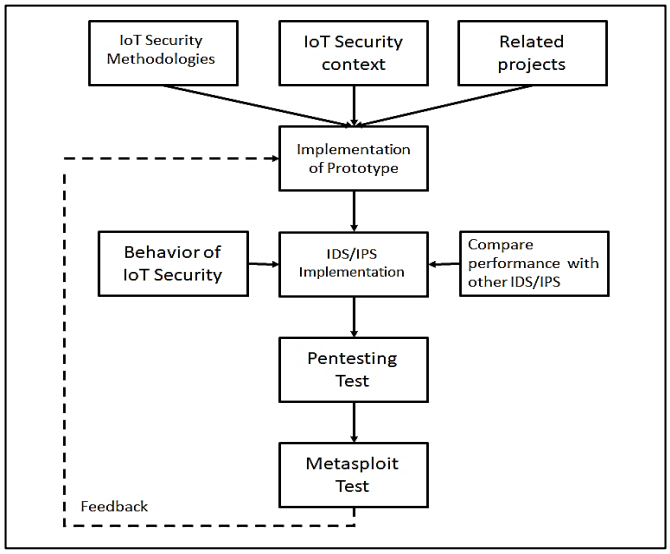



Fig. 2. Methodology in Block Diagram Implemented.

### A. IoT Prototype Implementation

A system based on Raspberry Pi3 + was built, with the Raspbian Operating System, using Python language and compatible components for Raspberry card [30]. They were assembled to verify system performance against attacks. The components used are described in Table IV.

The IDS / IPS system was installed and configured in the prototype and the functional tests of each of the sensors and actuators were performed, as shown in the diagram in Fig. 3.

TABLE. IV. IoT Components Assembled in the Prototype

| No. | Components | Type | Function |
|---|---|---|---|
| 1 | Raspberry Pi3+ | Card | Host IoT system with OS Raspbian 4.19 |
| 2 | Adc1 | Sensor | $I^2C$, a signal for detecting AC *I* |
| 3 | Adc2 | Sensor | $I^2C$, for detecting DC *I* |
| 4 | Adc3 | Sensor | $I^2C$, for detecting AC Voltage |
| 5 | Adc4 | Sensor | $I^2C$, , for detecting DC Voltage |
| 6 | BH1750FVI | Sensor | $I^2C$, for detecting light |
| 7 | PIR | Sensor | Detect presence /absence, ON/OFF |
| 8 | FZ0430 | Sensor | Detect DC voltage |
| 9 | MCP3424 | Card | Analog-Digital Converter with $I^2C$ |
| 10 | Relay 2 | Card | Relay 2 canals 5v |
| 11 | LED | Bulb | 127v bulb |
| 12 | Electromagnet | Actuator | Opening device |
| 13 | Motor | Actuator | 12v motor |
| 14 | Ov5647 | Actuator | Infrared night vision camera with IR sensor, 5MP |

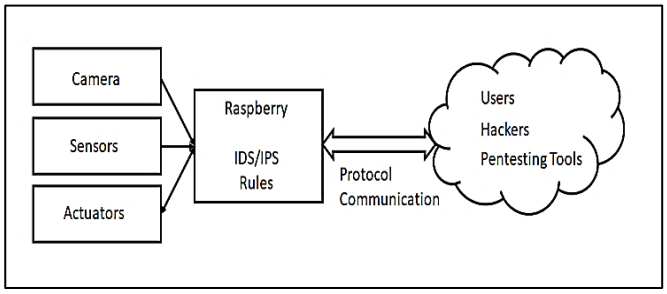


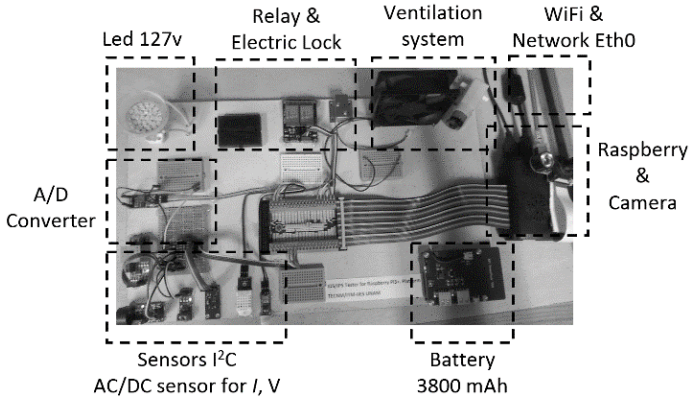

Fig. 3. System Design View and Attack Tests.

The designed prototype was integrated with the sensors and actuators to form an embedded and functional system, as shown in Fig. 4(a)). Tests were carried out with scripts in Python v3 language as shown in Fig. 4(b)).

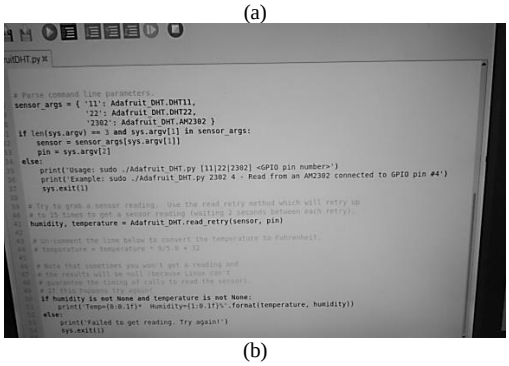
## B. IDS/IPS Implementation

Snort automates and simplifies intruder detection, using rules that describe the behavior of different attacks. The installation procedure in Raspbian is described below (it is important to have the equipment connected to the network):

```
#sudo apt-get update
#sudo apt-get install snort snort-common snort-common-libraries snort-rules-default libpcap-dev
#sudo dpkg-reconfigure snort
// sudo command vi /etc/snort/snort.debian.conf parameters
#sudo vi /etc/snort/snort.conf
// Review the configuration of the nine sections to adapt the operation of the IDS
#sudo rc.d stop snort
#sudo rc.d start snort
```




(a)



(b)

Fig. 4. Prototype Design and Testing, a) Component Integration and b) Sensors Testing in Raspbian with Python Language.

## C. Pentesting

IDS/IPS operation tests were performed using Nmap and Metasploit. With Nmap, the following instructions were applied:

```
#nmap –f –sS –sV –script auth 192.168.0.9
```

Fig. 5 shows the vulnerabilities detected by Nmap, in the active services that use authentication in the prototype.

Fig. 6 shows a list of all the vulnerabilities detected by Nmap in the prototype's active services.

```
#nmap –f –script vuln 192.168.0.9
```

Fig. 7 shows the traffic detected and blocked by Snort against attacks, in the scanning of vulnerabilities with Nmap. The metric used is the type of traffic per app.

## D. Vulnerability Test

Using Metasploit tool, the following exploits were applied: DDoS attack on port 80, which consists of saturating packets to that service, with the goal of denying the service to users.

```
#msfconsole
#use auxiliary/dos/tcp/synflood
#set RPORT 80
#set RHOST 192.168.0.9
#Run
```

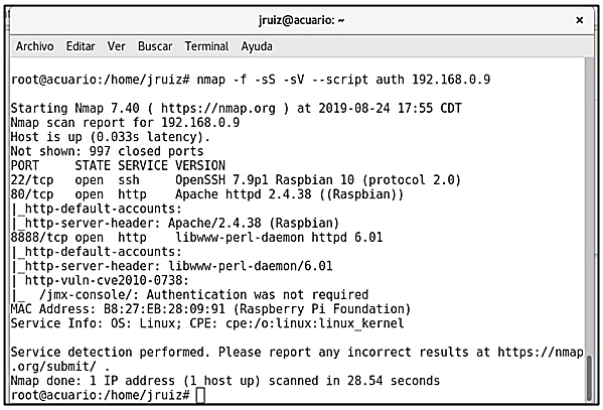



Fig. 5. Scanning Test with Nmap.

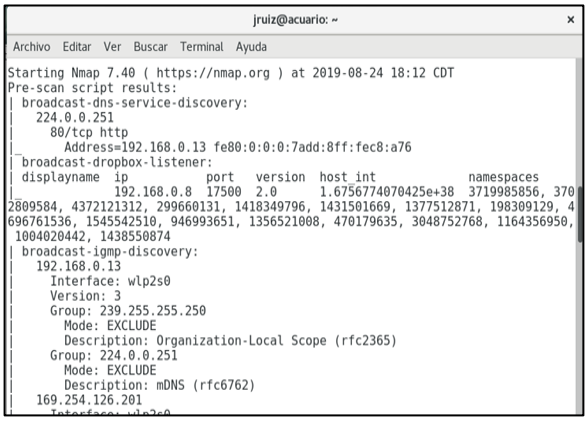



Fig. 6. Nmap Vulnerability Results.

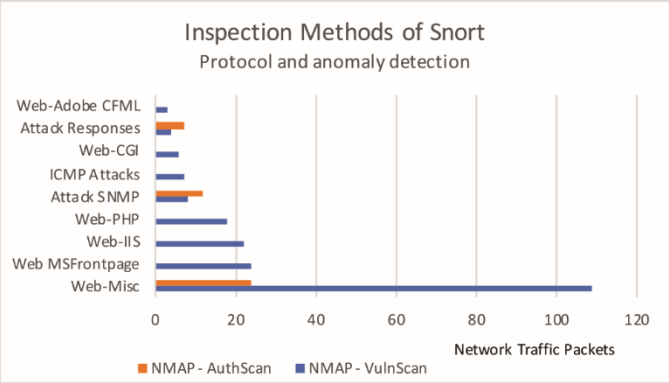



Fig. 7. Traffic Generated and Vulnerabilities Detected by Scanning.

The next step was developing the brute force attack, which consists of using a dictionary attack for breaking the password of a user account in the attacked system.

```
#Msfconsole
#Search ssh
#Use auxiliary/scanner/ssh/ssh_login
#Show options
#Set BLANK_PASSWORDS true
#Set PASS_FILE /root/Escritorio/pass.txt
#Set USER_FILE /root/Escritorio/users.txt
#Set RHOSTS 192.168.0.9
#Run
```

Intrusion attempts, generated by Metasploit, detected and blocked by the Snort tool.

Fig. 8 shows the amount of traffic generated by the attack trying to hack and block the web service this traffic was blocked by the IDS/IPS.

Fig. 9 shows the amount of traffic generated by trying to hack and compromise the ssh service, using a keys dictionary. The metric was the traffic generated by the Metasploit tool.

To configure the IPS is necessary to activate two basic elements, the whitelist (allowed hosts) and the blacklist (banned attacker hosts). Finally, add the preprocessing directives so that the IDS automatically applies the rules:

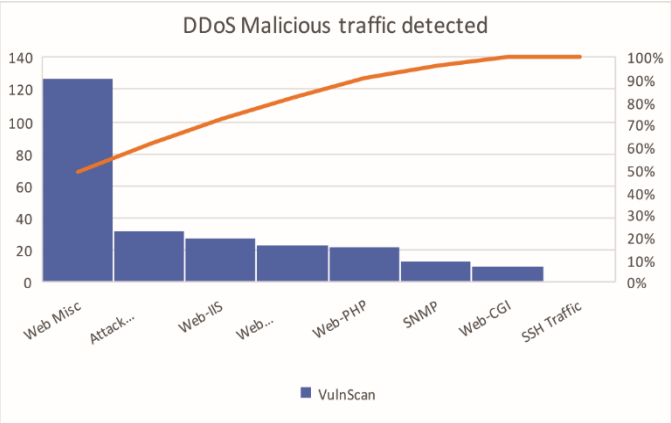



Fig. 8. Result of the DDoS Attack for Web Service Detected with Snort.

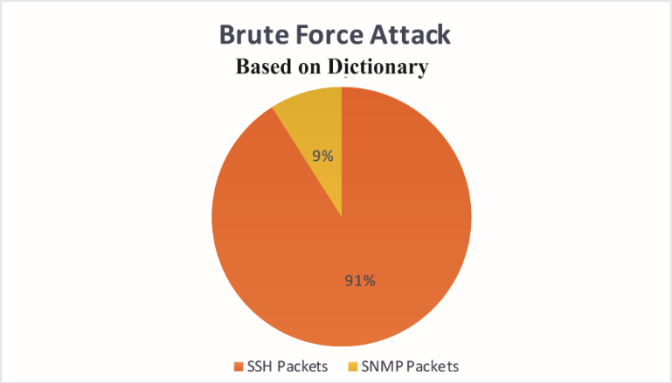



Fig. 9. Attack on SSH Service Detected with Snort Tool.

```
//Configure Snort IPS (edit snort.conf)
#sudo vi /usr/local/etc/snort/snort.conf
Add -ipvar HOME_NET 192.168.0.0/24 –make this match your internal network;
Add -ipvar EXTERNAL_NET !$HOME_NET //IPs of network home
Add -var RULE_PATH rules
Add -var WHITE_LIST_PATH rules //IPs from host allowed
Add -var BLACK_LIST_PATH rules
Add this to the end after “decompress_depth 65535”
max_gzip_mem 104857600
-Add this line -output unified2: filename snort.log, limit 128
-delete or comment out all of the “include $RULE_PATH” lines except:
#include $RULE_PATH/local.rules
#include $RULE_PATH/snort.rules–add after local.rules7.
//Now the following rules are uncommented, for:
preprocessor normalize_ip4
preprocessor normalize_tcp: ips ecn stream
preprocessor normalize_icmp4
preprocessor normalize_ip6
preprocessor normalize_icmp6.
```

## VI. Results

As described, the trend of using IoT components in industry 4.0 is perhaps the most complex challenge for security, it is changing the way that information is generated and exchanged. The problem observed is that due to the rapidity with which IoT devices are produced and used, due to demand, the communication security between the components is not properly established. The origin of the threats in the IoT derives from lack of training, investment, staff capacity, and security schemes.

It has demonstrated that the components, which were integrated into an IoT system, in their wild they were weak about security characteristics. They were not built considerating security parameters. So we made some penetration testing to demonstrate their behavior under some kind of attacks. We used at the same time an IDS/IPS tools in these tests, to demonstrate that is necessary to support and help an IoT system. The attacks went into the system in a direct way, only the IDS/IPS system helped to detect them. With Nmap, we obtained a list of vulnerabilities in the prototype then we did a DDoS attack on port 80, we also used brute force attack trying to guess a user’s password over SSH service. In these attacks that were tested, Snort detected the unusual traffic and behavior and it sent messages and warnings.

It is worth mentioning that initially in the development of the project, the manufacturing state of the prototype components was analyzed and the sensors and actuators were implemented. With this, it was determined that despite these components, it was advisable to improve its security with specialized tools; this phase of work was done with the Snort software. The Sagan tool proved to be more demanding in the use of memory and processing, recommending its implementation in multi-threaded architectures that can support the demand.

## VII. Conclusion and Future Work

In this work, it has demostrated how strong or weak are the IoT components against some common attacks which can be found on the Internet. We found that the components of an IoT system were built with no considerations on security schemas and response against common attacks. So, is recommend installing an IDS/IPS to secure an IoT system, to prevent and warning against some cyber attacks.

The implementation of Snort as an intruder detection system allowed real-time detection of port scanning and attempts to breach the system from other hosts; providing an opportunity to measure how systems are compromised. This provides a new opportunity for an investigation to model this behavior. Finally, it is important considering the stabilization of rules for IDS/IPS so that the system permits secure communication without repudiation.

Our future works can treat over new penetration tests, set up new rules in the IDS/IPS, encrypted messages among wireless components and integrate all components and tools to implement a platform of IoT secure scheme.

## Acknowledgments

We thank Tecnológico Nacional de México/Instituto Tecnológico de Morelia, as well as Universidad Nacional Autónoma de México, for the support granted for this research, together with project 5774.19-P “Development and implementation of a secure IoT architecture, based on penetration tests, using fuzz models and detection systems with an IDS and IPS” Spanish translate “Desarrollo e implementación de una arquitectura IoT segura, basado en pruebas de penetración, utilizando modelos fuzz y sistemas de detección con un IDS e IPS”. We also thank professors Abel A. Pintor from ITMorelia and from IIES-UNAM who helped the development of the prototype, Diego Cabrer, Atzimba López, Alberto Valencia, and Yumi Tzib.

## References

[1] Haroon A., Naeem W., Shah M. A., Kamram M., Asim Y. & Javaid Q. “Constraints in the IoT: The World in 2020 and Beyond”. International Journal of Advanced Computer Science and Applications,Vol. 7, No. 11, 2016.

[2] Santoso F. K. & Vun N. C. H. “Securing IoT for smart home system”. International Symposium on Consumer Electronics (ISCE), 2015. IEEE. ISBN: 978-1-4673-7365-4. DOI: 10.1109/ISCE.2015.7177843.

[3] Qinghe D., Houbing S. & Xuejie Z. “Social-Feature Enabled Communications Among Devices Toward the Smart IoT Community”. IEEE Communications Magazine. Volume 57 Issue 1. January-2019. DOI: 10.1109/MCOM.2018.1700563.

[4] Rouhiainen Tuukka. “Scanning the Internet to find security loopholes”. Proceedings of the Seminar in Computer Science: Internet, Data and Things (CS-E4000). Computer Science at Aalto University. 2018.

[5] Mosca, D. “Hacking the internet of things just got easier – it’s time to look at your security”. [Online]. Available: https://www.computer weekly.com/opinion/Hacking-the-Internet-of-Things-just-got-easier-its-time-to-look-at-your-security. [Accessed April, 2019].

[6] Rishi Rahul & Saluja Rajeev. “Future IoT”. Ernst & Young Associates LLP, Published in India. 2019

[7] Lovejoy C., Watson R. & Pizzala J.. “Internet of Things and Operating Technology Security”. [Online]. Available: https://www.ey.com/en_gl/advisory/iot-operating-technology-security. [Accessed August 2019].

[8] Amazon Web Services, “Internet de las cosas, Plataforma como servicio AWS IoT,” 2019. [Online]. Available: https://aws.amazon.com/es/iot/. [Accessed: 23-Apr-2019].

[9] Microsoft. (2019). IoT Hub | Microsoft Azure. Retrieved April 23, 2019, from https://azure.microsoft.com/es-mx/services/iot-hub/

[10] Oracle, “Internet of Things | Oracle Cloud,” 2019. [Online]. Available: https://cloud.oracle.com/iot. [Accessed: 23-Apr-2019].

[11] IBM, “IBM Watson Internet of Things (IoT)”, 2019. [Online]. Available: https://www.ibm.com/mx-es/internet-of-things. [Accessed: 23-Apr-2019]

[12] G. I. Xively, “IoT Platform for Connected Devices”, 2019. [Online]. Available: https://xively.com/. [Accessed: 23-Apr-2019].

[13] Samsung Co., “IoT Cloud Platform, Samsung ARTIK cloud services”, 2019. [Online]. Available: https://artik.cloud/. [Accessed: 02-May-2019].

[14] Altair Engineering Inc., “Altair SmartWorks” 2019. [Online]. Available: https://www.altairsmartworks.com/index.php/. [Accessed: 02-May-2019].

[15] Adafruit, “Welcome to Adafruit IO” 2019. [Online]. Available: https://io.adafruit.com/. [Accessed: 02-May-2019].

[16] Ubidots, “IoT platform Ubidots” 2019. [Online]. Available: https://ubidots.com/. [Accessed: 02-May-2019].

[17] MyDevices, “The IoT Solutions Company” 2019. [Online]. Available: https://mydevices.com/. [Accessed: 02-May-2019].

[18] Macchina.io, “IoT Edge Device Software Development and Secure Remote Access Solutions”, 2019. [Online]. Available: https://macchina.io/. [Accessed: 02-May-2019].

[19] ThingSpeak, “IoT Analytics, ThingSpeak Internet of Things”, 2019. [Online]. Available: https://thingspeak.com/. [Accessed: 02-May-2019].

[20] Arduino, “Arduino” 2019. [Online]. Available: https://www.arduino.cc/en/IoT/HomePage. [Accessed: 02-May-2019].

[21] Tomas Zitta, “Penetration Testing of Intrusion Detection and Prevention System in Low-Performance Embedded IoT Device” (2018). IEEE-Xplore. Retrieve: https://ieeexplore.ieee.org/document/8624734. [Accessed: July-2019].

[22] Cisco Systems, “Snort - Network Intrusion Detection & Prevention System.” [Online]. Available: https://snort.org/. [Accessed: 13-Jul-2019].

[23] Project Suricata, “Suricata, Open Source IDS/IPS/NSM engine.” [Online]. Available: https://suricata-ids.org/. [Accessed: 13-Jul-2019].

[24] The Zeek Network Security Monitor, “The Zeek Network Security Monitor.” [Online]. Available: https://www.zeek.org/index.html. [Accessed: 13-Jul-2019].

[25] OSSEC Project Team, “OSSEC -World’s Most Widely Used Host Intrusion Detection System-” [Online]. Available: https://www.ossec.net/. [Accessed: 13-Jul-2019].

[26] Tripwire, “Cybersecurity and Compliance Solutions”. [Online]. Available: https://www.tripwire.com/. [Accessed: 14-Jul-2019].

[27] Linux, “Intrusion detection with AIDE”. [Online]. Available: https://www.linux.com/news/intrusion-detection-aide. [Accessed: 14-Jul-2019].

[28] Samhain design labs, “Samhain Labs” [Online]. Available: https://www .la-samhna.de/samhain/index.html. [Accessed: 18-Jul-2019].

[29] Fail2ban Project, “Fail2ban.” [Online]. Available: https://www.fail2ban .org/wiki/index.php/Main_Page. [Accessed: 19-Jul-2019].

[30] Q. I. S. Sagan Project, “The Sagan Log Analysis Engine | Quadrant Information Security.” [Online]. Available: https://quadrantsec.com/sagan_log_analysis_engine/. [Accessed: 19-Jul-2019].