DOI: https://doi.org/10.33103/uot.ijccce.23.2.8

Design and Implementation of Security Gatway for IoT Devices Security

Marwan Alaa Hussein[1], Ekhlas Kadhum Hamza[2]
[1,2]*Control and Systems Engineering Department, University of Technology, Baghdad, Iraq*
[1]*cse.19.09@grad.uotechnology.edu.iq*, [2]*ekhlas.k.hamza@uotechnology.edu.iq*

Abstract*— As the Internet of Things (IoT) is growing in popularity globally, which has resulted in a rise in cyber threats, experts are focusing more on its security. The majority of IoT security research to date has concentrated on huge devices, while small IoT devices have received comparably little attention. Our primary purpose is, therefore, to research how to ensure the operation of IoT devices that are small. The security gateway is a Security Settings on the Gateway for RaspberryPi built gateway that may link Internet of Things devices to their private network, safeguarding IoT devices from exposure to external networks. In addition, a variety of Security Settings on the Gateway for RaspberryPi security settings are installed, including fiel2ban and a Security Settings on the Gateway for RaspberryPi firewall, in order to avoid brute force and dictionary attacks. This article also studies the communication between Internet of Things (IoT) devices utilizing various secure communications, including Secure Shell (SSH), and analyzes their performance in a variety of circumstances. The gateway's experimental evaluation reveals that the proposed framework can secure tiny IoT devices.*

Index Terms*— Internet of Things (IoT), Security IoT, Security Gateway, Security Settings on the Gateway for* RaspberryPi, *ESP8266.*

I. INTRODUCTION

The Internet of Things is a network that joins several physical items to the web in order to exchange information and communications for smart location, tracking, administration, and monitoring [1, 2]. Network security has progressively emerged as one of the most important challenges in the Internet industry, and network security breaches are growing globally as more devices are connected to the Internet of Things in order to provide novel and interconnected services [3, 4]. The security of key Internet of Things (IoT) devices, such as linked cars, smart homes, connected technology, wearable equipment, and devices with remote monitoring skills, is the primary focus of current research [5]. Researchers have paid very little attention to the security of tiny IoT devices. Some makers of IoT devices for small businesses can offer users updates through the official repository. However, some non-experts with an interest in the IoT may employ native tools for the creation of specific systems or projects (using a nodeMCU8266 development board). In this case, they just guarantee that the system's small IoT devices can function as planned, regardless of the devices' security. By building a security gateway to protect IoT devices, this analysis aims to create a safe environment for small IoT devices. This gateway was developed with a *Security Settings on the Gateway for* RaspberryPi and is known as the Security Gateway. Small IoT devices will not be accessible to external networks due to the construction of a private network and the assignment of a private IP address to each device. In addition, we established a number of measures to secure the gate, which will be described in full within the scope of this investigation. Consequently, every company can invest heavily in Internet of Things devices, which can be deployed globally. must thus

Received 31/August/2022; Accepted 09/November/2022

ISSN (Print) 1811-9212 *ISSN (Online) 2617-3352*

DOI: https://doi.org/10.33103/uot.ijccce.23.2.8

assess the connectivity of IoT devices. To ensure that the connection is secure, we analyzed the security of IoT device connections using a number of common methodologies and provided suggestions for a variety of environmental scenarios. Below are the main components discussed in this research, as follows:

A. Raspberry Pi

Raspberry Pi is the name of a series of single-board computers made by the Raspberry Pi Foundation, a UK charity that aims to educate people in computing and create easier access to computing education [6].

The Raspberry Pi launched in 2012, and there have been several iterations and variations released since then. The original Pi had a single-core 700MHz CPU and just 256MB RAM, and the latest model has a quad-core CPU clocking in at over 1.5GHz, and 4GB RAM. The price point for Raspberry Pi has always been under $100 (usually around $35 USD), most notably the Pi Zero, which costs just $5 [6].

All over the world, people use the Raspberry Pi to learn programming skills, build hardware projects, do home automation, implement Kubernetes clusters and Edge computing, and even use them in industrial applications [7].

The Raspberry Pi is a very cheap computer that runs Linux, but it also provides a set of GPIO (general purpose input/output) pins, making controlling electronic parts for physical computing and learning about the Internet of Things (IoT) possible.

B. ESP8266

Chinese manufacturer Espressif has created the ESP8266, a system on a chip (SoC). This device consists of a Tensilica L106, a 32-bit microcontroller (MCU), and a WiFi transceiver. A total of 11 GPIO pins (general-purpose input/output pins) are available, as well as an analog input. As a result, it can be programmed in the same way as an Arduino or other microcontroller [8]. Wi-Fi connectivity is also provided, allowing the server to be used as a wireless access point, connect to the Internet, run a web server, and allow a smartphone to connect to it. There is no end to the possibilities! With so many different modules available, including standalone modules like the AI Thinker's ESP-## series and development boards like the NodeMCU DevKit and WeMos D1, it's no surprise that this chip has become the most popular IoT device to access. It is possible that different boards have defective pins, different Wi-Fi antennas, or varying amounts of flash memory. However, six of the GPIO pins of the ESP8266 chip (6-11) are used to interact with the integrated flash memory chip, as shown in *Fig. 1.*

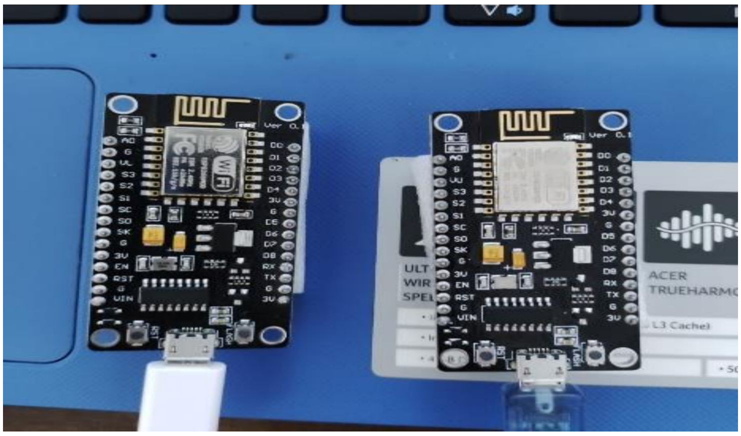


FIG. 1. THE ESP8266 MICROCONTROLLER.

Received 31/August/2022; Accepted 09/November/2022

DOI: https://doi.org/10.33103/uot.ijccce.23.2.8

C. Oracle VirtualBox machine (VM)

A hypervisor that allows users to run various operating systems by generating a virtual machine on their hardware. It is meant to take advantage of advances in the x86 architecture and is an easy-to-install KVM paravirtualization hypervisor. It is a great choice for home and office hypervisors because it can take snapshots and run in a seamless mode.

D. Security issues for IoT devices

These are some of the most prevalent security flaws affecting IoT devices:

- Some online markets and websites sell counterfeit components.[4].
- Weak password: For simplicity of usage, IoT devices employ weak or duplicate passwords. In addition to the fact that users don't like to change default passwords, the attacker uses certain methods to figure out the system password.[8].
- Information leakage: IoT devices are large, interconnected networks that leak a huge amount of information, either mistakenly or maliciously [9], leaving them subject to exploitation and greed on the part of hackers.
- Unauthorized access: An attacker can get into the target system without permission and take control of it [4].
- Remote Code Execution: With insufficient skills in proper coding, developers fail stringent input parameter filtering and validation, leading to remote code activation or command interference during the performance of risky procedures [9].
- Man-in-the-middle (MITM) attack: The attacker is positioned between both sides of the connection and operates as a data exchanger [4, 9].
- Cloud attack: As connected gadgets are increasingly managed in the cloud [9], attackers exploit cloud service flaws to investigate the connections between both the device and the cloud, as well as fabricated data, to perform replay attacks and seize control of the system.

This study safeguards small Internet of Things devices against the aforementioned security threats. The components are collected from official websites, thereby validating their legitimacy. We've built a strong password for the *Security Settings on the Gateway for* RaspberryPi to circumvent the issue of weak passwords. The application file2ban was also utilized to prevent brute-force assaults [10]. We have secured the information to avoid leakage, as well as the *Security Settings on the Gateway for* RaspberryPi and Internet of Things firmware, to prevent unauthorized access. To avoid remote code execution and man-in-the-middle (MITM) attacks, we have implemented SSH port forwarding and two-factor authentication with a token. In conclusion, Wazuh software was used to monitor all portal-related processes [11]. The remaining sections are organized as follows: Section II introduces the literature review. The design of the security gateway framework, interior, and exterior are discussed in Section IV. In Section V, we'll go over the study's configurations and findings, covering topics like the *Security Settings on the Gateway for* RaspberryPi security configurations, the Security Gateway's inside and exterior connections, and the Raspberry's external connection. In Section VI, we assess and summarize the research's results. In the article's concluding section, "Section VII," discuss what comes next.

Received 31/August/2022; Accepted 09/November/2022

DOI: https://doi.org/10.33103/uot.ijccce.23.2.8

II. LITERATURE SURVEY

In order to update, manage, and regulate IoT devices and provide a secure network ecosystem, Hu et al. [13] employ the *Security Settings on the Gateway for* RaspberryPi to develop a basic security gateway named Raspberry Gate and Raspberry Monitor (human community). Raspberry Gate can be used in router mode, bridged mode, or maintenance mode. Modus de routing facilitates the translation of IP data between public and private networks based on the IP address of the sending or receiving node. In this case, *Security* Settings on the Gateway for RaspberryPi Guardian could utilize Raspberry Gate to keep the local network and its attached devices safe. While in bridge mode, packets can be bridged between incoming and outgoing traffic with or without packet filters. Automatic upgrades to the Security Settings on the Gateway for RaspberryPi Gate are performed during this time. Security Settings on the Gateway for RaspberryPi Gate can get the most recent version of Raspberry Guardian's code from GitHub and use it to update itself. H. Sun, et al. [14] They have devised a fog-cloud-enabled IoT architecture that uses the best features of fog and cloud. An ETCORA algorithm has been applied to improve energy consumption and complete applications. The authors showed the simulation results that can reduce the amount of energy used and the time it takes to respond.Tomas Zitta, et al. [15] The author presented the developed detection rules for the LLRP (Low Level Reader Protocol) interface and theimplementation of the IDS (Intrusion Detection System) and IPS (Intrusion Prevention System) RFID (Radio Frequency Identification) and UHF (Ultra High Frequency) reader solutions using the SecuritySettings on the Gateway for RaspberryPi 3 security solution. developed to secure existing LLRP-enabled RFID readers, comparing IPS and IDS solutions, and then selecting the right solutions for the application using the Security Settings on the Gateway for RaspberryPi.

The main contribution is the security of RFID readers, which are usually available on the market without any cryptographic support. It has been stated by V. Teeraratchakarn and Y. Limpiyakorn. [16]H. Sun, et al. [14] They have devised a fog-cloud-enabled IoT architecture that uses the best features of fog and cloud. An ETCORA algorithm has been applied to improve energy consumption and complete applications. The authors have shown the simulation results that are able to minimize energy expenditure and response time.Tomas Zitta, et al. [15] The author presented the developed detection rules for the LLRP (Low Level Reader Protocol) interface and the implementation of the IDS (Intrusion Detection System) and IPS (Intrusion Prevention System) RFID (Radio Frequency Identification) and UHF (Ultra High Frequency) reader solutions using the Security Settings on the Gateway for RaspberryPi3 security solution. developed to secure existing LLRP-enabled RFID readers, comparing IPS and IDS solutions, and then selecting the right solutions for the application using the *Security Settings on the* Gateway for RaspberryPi.

The main contribution is the security of RFID readers, which are usually available on the market without any cryptographic support. It has been stated by V. Teeraratchakarn and Y. Limpiyakorn. [16]The authors explored the capabilities of an open-source monitoring and logging product called Elastic Stack to discern the most common user names used by attackers during a brute-force attack. The ultimate goal was to advise organizations to regulate their internal identities better as an added safety measure. Although their conclusion shares similarities with this dissertation's ultimate conclusion, their exploration of monitoring and logging tools was limited to brute-force attacks due to their focus on identities and user names. While evaluating the current state of IoT applications, Neshenko et al. [17] focused on security flaws in the IoT. The authors organized a wide variety of IoT security flaws, attack methods, and mitigation strategies into manageable groups. According to F. Mulyadi et al. [18], the authors proposed the Elastic Stack Anchor system for security information and event management. Its supporting features are light weight and

Received 31/August/2022; Accepted 09/November/2022

DOI: https://doi.org/10.33103/uot.ijccce.23.2.8

simplicity. One important feature of Elasticsearch is security information. Elasticsearch can function as an intrusion detection system (IDS) in conjunction with log and event management systems such as SIEM. In addition to ELK and Wazuh with docked ELK, efficiency will also be obtained because Docker is based on a lightweight container and more than one instance can be deployed on a single host machine for a given use case. Therefore, it is concluded that the docked ELK has added value and is good to use. F. Balseca-Chávez et al. [19] proposed an architecture for analyzing data through big data tools using events or security logs, allowing for better identification, integration, and correlation of events. Big data processing stages have been used to identify computer threats. The designed technology architecture was based on the integration of Elastic Stack and its key components (Elasticsearch, Logstash, and Kibana), technologies such as Filebeat and Wazuh Security Detection (NIPS and HIDS), and security management of information assets such as communications and data equipment, application servers, database engines, and end-user terminals. Its implementation will allow real-time and date monitoring for quick and effective responses to security alerts and incident status reports.

III. SECURITY GATEWAY FRAMEWORK DESIGN

The architecture and layout of the security gateway are covered here. *Fig. 2* depicts the desired structure. It describes how the Security Settings on the Gateway for RaspberryPi gateway links and connects small Internet of Things and computer devices. Small IoT devices can be assigned a private network IP address by supplying a gateway IP address. Therefore, all small IoT devices are contained within a secure private network. The gateway is Internet-connected to promote communication between small IoT devices.

A. Design Framework

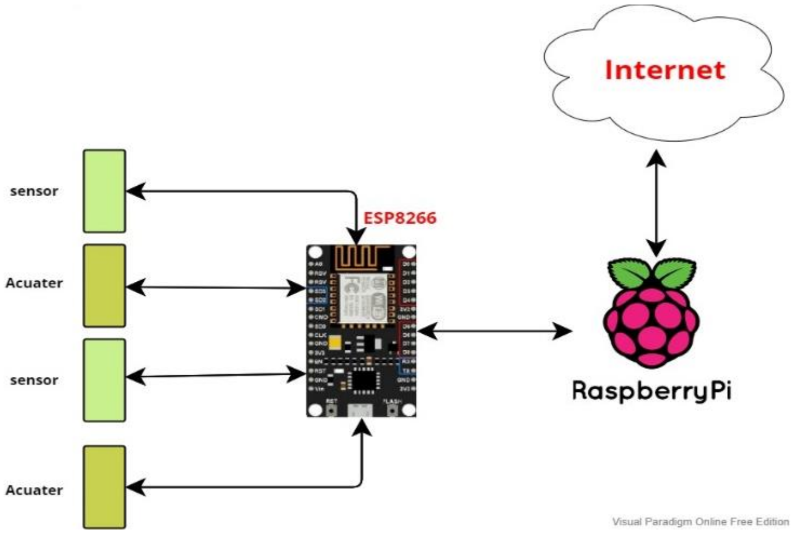


FIG. 2. DESIGN FRAMEWORK.

The Security Settings on the Gateway for RaspberryPi can act as a wireless Ethernet access point if a private network is set up. Therefore, the Security Settings on the Gateway for RaspberryPi is solely responsible for the resulting wireless network. To connect the external network interface represented by the cloud and the internal network represented by IoT devices, we use the Security Settings on the Gateway for RaspberryPi as the security gate. We used the ESP8266 as IoT devices (sensors and actuators) and represented it with

Received 31/August/2022; Accepted 09/November/2022

DOI: https://doi.org/10.33103/uot.ijccce.23.2.8

two nodes; the first was a server and the second was the master, as shown in *Fig. 1*. The Internet of Things devices are connected to the Security Settings on the Gateway for RaspberryPi via WiFi. The home router also has an internet connection. Therefore, the security of low-power IoT devices connected through our portal has been enhanced. Some improvements have also been added to the portal by changing the default SSH port and adding Wazuh, which is one of the best systems for detecting intrusions and is the new addition to this work, as will be explained in the subsequent sections. *Fig. 3* shows how we looked at SSH port routing technologies and cloud services to connect small IoT devices to the cloud in a safe way.

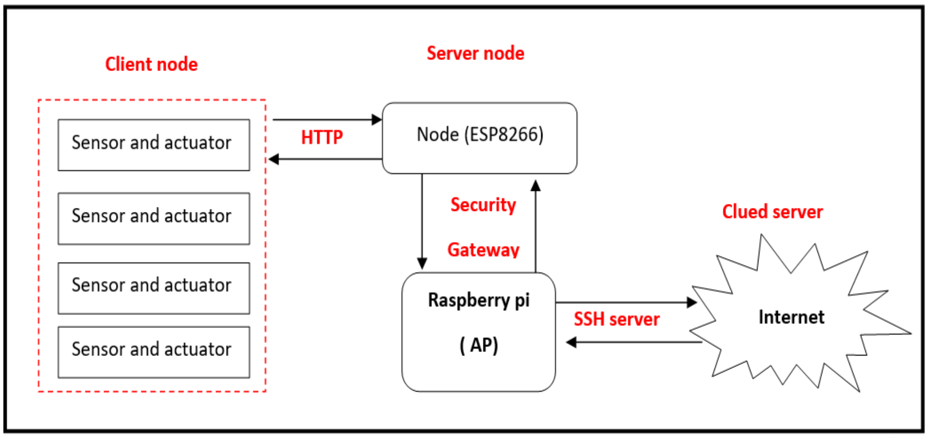


FIG. 3. IMPLEMENTATION FRAMEWORK.

B. Security The Gateway Design

The newest *Security Settings on the Gateway for* RaspberryPi features built-in WiFi that works with a wide variety of devices. This study makes use of the *Security Settings on the Gateway for* RaspberryPi as a gateway device. Our entry point, referred to as the Security Gateway, is a wireless access point that connects small IoT devices to a private network in order to protect those devices. Security The Gateway Design: Interior Setting Utilize two-factor authentication between IoT devices based on the token and time-based one-time password (TOTP) algorithm to safeguard IoT devices in the internal environment and prior to the gateway by installing a server that connects the gateway and IoT devices to the Two-Factor Authentication (TFA) application. Security The Gateway Design: Exterior Setting In this section, various methods were employed to ensure security outside the gate. For example, to monitor the gateway from the external Kibana dashboard, use SSH port forwarding to specify the port, install Fail2ban, turn on the firewall (UFW), and add the Wazuh proxy.

IV. EXPERIMENTAL SETTINGS AND ANALYSIS

The Raspbian operating system employs the same Linux commands and language as Ubuntu. First, "sudo apt-get update –y" was executed to ensure that the operating system was up-to-date. Additionally, the command "sudo apt-get dist-upgrade" was executed to upgrade any package that requires an upgrade. After the update and upgrade process was complete, the device was rebooted to apply all the changes. Next, verify the IP address of each device. When the connection to any WiFi network is complete, type ifconfig to display the device's IP address in the console. If the Internet is connected via WiFi, the network card's name should be listed under wlan0. If they are linked through Ethernet, the network card should be eth0. *Fig. 4* shows that once the IP addresses have been set up correctly,, the device is ready.

Received 31/August/2022; Accepted 09/November/2022

DOI: https://doi.org/10.33103/uot.ijccce.23.2.8

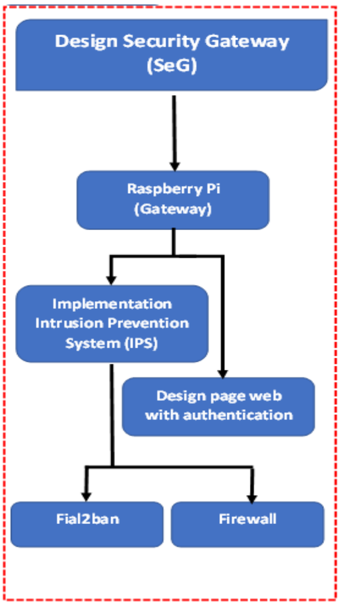
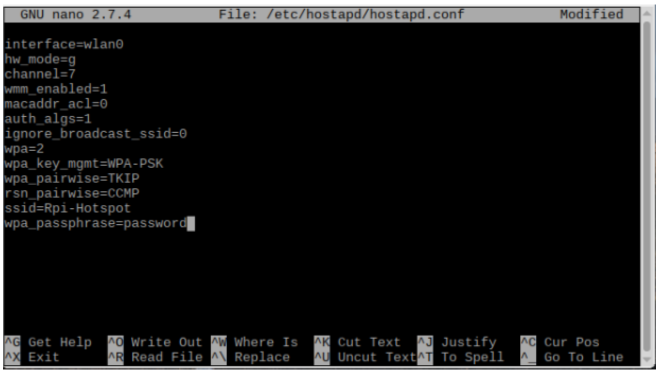

Design Security Gateway (SeG)
Raspberry Pi (Gateway)
Implementation Intrusion Prevention System (IPS)
Design page web with authentication
Fial2ban
Firewall

FIG. 4. THE STEPS OF SEGW INSTALL.

A. General Settings Security Settings on the Gateway for RaspberryPi

Utilize Virtual Box to install the RaspberryPi and operating system on the RaspberryPi in order to construct a private network and create a wireless access point and gateway. Before launching the operating system, plug the Pi into Ethernet network. To use the RaspberryPi as an access point, must first install the hostapd access point [1] (Host access point daemon), a user-space software access point capable of turning normal network interface cards into access points and authentication servers [19], and configure the appropriate settings. The WiFi network name for the RaspberryPi is called the SSID, and it is set in the hostapd configuration file for the gateway [20]. We also create the password that is needed to access the RaspberryPi for the first time while connecting gadgets. When we're done, we'll be able to run and start hostapd, as seen in *Fig. 5*.



```
  GNU nano 2.7.4          File: /etc/hostapd/hostapd.conf                Modified

interface=wlan0
hw_mode=g
channel=7
wmm_enabled=1
macaddr_acl=0
auth_algs=1
ignore_broadcast_ssid=0
wpa=2
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP
ssid=Rpi-Hotspot
wpa_passphrase=password
```

FIG. 5. INSTALLATION HOSTAPD.

The wlan0 connection. A result of Hostapd status checks is shown in *Fig. 3*. To have the RaspberryPi act as a DHCP server for the wireless network, set 192.168.6.1 as the static IP address for the wireless interface (wlan0). In terms of DNS and DHCP configuration, dnsmasq is a lightweight tool. The protocol is suited for local area networks (LANs) and supports DHCP and DNS. With a wlan0 network, IP addresses for DHCP clients will be assigned for a period of 24 hours. Wireless users also

Received 31/August/2022; Accepted 09/November/2022

DOI: https://doi.org/10.33103/uot.ijccce.23.2.8

have access to the RaspberryPi through the Security gateway. Configuring dnsmasq After added routing and masquerade, restart the RaspberryPi and connect a wireless device to see if the wireless access point is working. For the hostapd configuration to work, both the network's SSID and password must be present.

B. Security Settings on the Gateway for Raspberry– Special Settings

In order to make the gateway more secure, we have implemented the following security settings: As shown in *Fig. 6*:

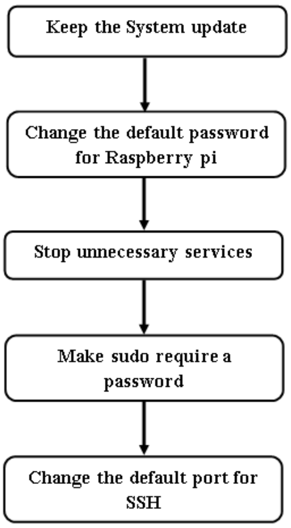

Keep the System update
Change the default password for Raspberry pi
Stop unnecessary services
Make sudo require a password
Change the default port for SSH

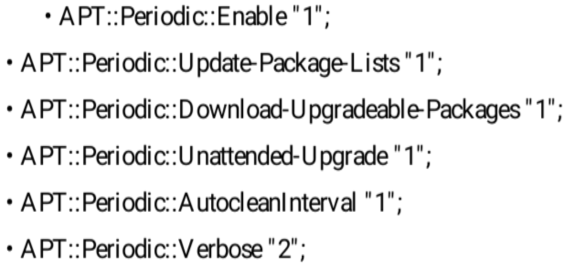
FIG. 6. STEPS INSURANCE FOR RASPBERRY PI OPERATING SYSTEM.

1. *Keep system updated*

- By using the unattended-upgrades package, automatically update the RaspberryPi.

This process permits daily automatic installation of security patches:
# sudo apt update
# sudo apt upgrade

- Launch the setup file: # sudo nano /etc/apt.conf.d/50unttended-upgrades
- Open this file:# sudo nano /etc/apt/apt.conf.d/02periodic
- Paste these lines



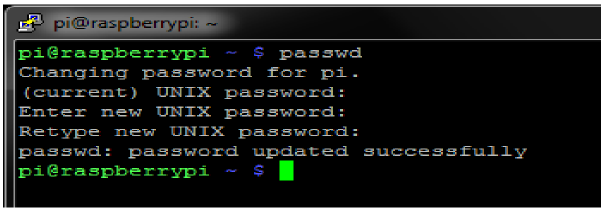
```
• APT::Periodic::Enable "1";
• APT::Periodic::Update-Package-Lists "1";
• APT::Periodic::Download-Upgradeable-Packages "1";
• APT::Periodic::Unattended-Upgrade "1";
• APT::Periodic::AutocleanInterval "1";
• APT::Periodic::Verbose "2";
```

2. *Changing the default password for RaspberryPi*

A frequent error is to keep the default username and password for the user (RaspberryPi). This password will be known to everyone who has previously used a RaspberryPi Numerous persons are monitoring SSH ports and attempting Pi/raspberry logins. Simply log in and enter the command below:
# Passwd

Received 31/August/2022; Accepted 09/November/2022

DOI: https://doi.org/10.33103/uot.ijccce.23.2.8



```
pi@raspberrypi ~ $ passwd
Changing password for pi.
(current) UNIX password:
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
pi@raspberrypi ~ $
```

FIG. 7. CHANGE THE DEFAULT PASSWORD FOR RASPBERRYPI.

a. *Stop unnecessary services*

- Try to uninstall or delete unneeded services and programs:
- List services in operation: # sudo service –status -all To cancel a service: # sudo service <service- name> stop

b. *Add a password prompt to sudo*

- It's common knowledge that sudo doesn't always prompt the user for a password. There is a strong probability that will not have to frequently retype password. It's a fantastic idea from a production standpoint, but a terrible idea from a safety standpoint.
- Edit this file: **# sudo nano /etc/sudoers.d/010 pi-nopasswd**
- Find this line**:# pi ALL=(ALL) NOPASSWD: ALL**
- Replace it with**:# pi ALL=(ALL) PASSWD: ALL**
- Save and exit (**CTRL+O, CTRL+X)**

c. *SSH: Prevent root login*

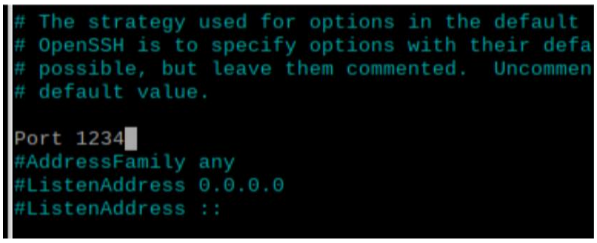
- Users with the root or pi privileges are frequently the primary target of brute-force attacks. This is especially the case when SSH access is allowed [21].
- Consequently, are tasked with ensuring that root does not have direct access to SSH. If require root access, log in as regular user (not pi), and then use sudo to obtain the necessary permissions to become the super-user. Access to the root account is prevented by default.
- A SSH server configuration file must be opened: # sudo nano /etc/ssh/sshd_config
- Find this line: #permitROOTLogin prohibit-password
- If they have something else, comment this line (by adding # at the beginning)

d. *In order to use SSH, must alter the default port.*

- By default, SSH uses port 22. Attackers will likely use automated software to probe for vulnerabilities over this port. Modifying the settings for port 22 can stop this from happening automatically. Change the settings for the SSH server:
  # sudo nano /etc/ssh/sshd.config
- Find this line: **# port22**
- Replace the port with the one want to use, and make sure to uncomment the line: **# Port 1234**
- Restart the server: # sudo service ssh restart.

Received 31/August/2022; Accepted 09/November/2022

DOI: https://doi.org/10.33103/uot.ijccce.23.2.8



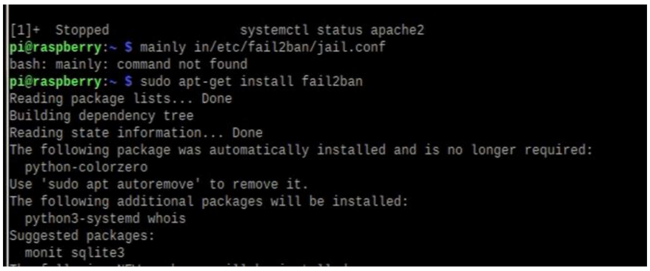
```
# The strategy used for options in the default
# OpenSSH is to specify options with their defa
# possible, but leave them commented.  Uncommen
# default value.

Port 1234
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::
```

FIG. 8. CHANGE THE DEFAULT PORT OF SSH.

C. Install Fail2ban on the Gateway

Protection from brute-force attacks, one of the most frequent types of attacks on IoT devices, is desirable. The goal of brute force attacks is to repeatedly try to guess server credentials [22]. The Fail2ban software monitors login attempts to a server. With fail2ban, the IP address of a device that repeatedly tries to connect to a server without the correct credentials can be blocked. fail2ban can be configured to block any IP address that connects to the server more than three times per day [23]. *Fig. 9* depicts how to install Fail2ban on the gateway.



```
[1]+  Stopped                 systemctl status apache2
pi@raspberry:~ $ mainly in/etc/fail2ban/jail.conf
bash: mainly: command not found
pi@raspberry:~ $ sudo apt-get install fail2ban
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  python-colorzero
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  python3-systemd whois
Suggested packages:
  monit sqlite3
```

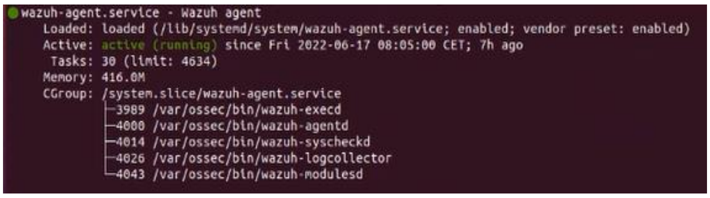
FIG. 9. INSTALLATION FAIL2BAN.

D. Install Wazuh on the Gateway

Wazuh is an open-source security platform classified as an intrusion detection system (IDS). It is a host-based (Wazuh agent) in the sense of endpoint detection and response (EDR), with a primary focus on infrastructure monitoring, security risk detection, and incident response. It consists of numerous components [24]:

- Wazuh Server: is an application suite that aims to analyze logs, generate alerts when detecting a malicious event, add new clients or agents, and send logs to the server.
- Registration service.
- RESTful API (Wazuh API)
- Filebeat.
- Wazuh agent: The Wazuh agent runs on the monitored system and is responsible for collecting log and event data, performing analytics policy monitoring, malware and rootkit detection, and triggering alerts when files are modified.

shown in *Fig. 10*. Install the Wazuh agent on the gateway.



```
● wazuh-agent.service - Wazuh agent
     Loaded: loaded (/lib/systemd/system/wazuh-agent.service; enabled; vendor preset: enabled)
     Active: active (running) since Fri 2022-06-17 08:05:00 CET; 7h ago
      Tasks: 30 (limit: 4634)
     Memory: 416.0M
     CGroup: /system.slice/wazuh-agent.service
             ├─3989 /var/ossec/bin/wazuh-execd
             ├─4000 /var/ossec/bin/wazuh-agentd
             ├─4014 /var/ossec/bin/wazuh-syscheckd
             ├─4026 /var/ossec/bin/wazuh-logcollector
             └─4043 /var/ossec/bin/wazuh-modulesd
```

FIG. 10. INSTALLATION WAZUH AGENT.

Received 31/August/2022; Accepted 09/November/2022

DOI: https://doi.org/10.33103/uot.ijccce.23.2.8

E. In-House Prototyping of Internet-of-Things Devices using RaspberryPi and Other Small Microcontrollers

This study's gateway contains a NodeMCU because of resource constraints (a microcontroller unit for a small IoT device). We constructed an IoT device out of a Micro USB cable, three LEDs, a NodeMCU board, breadboard jumpers, and a breadboard, and we used the Arduino IDE to program the NodeMCU board to control the LEDs' on and off states. Wi-Fi functionality is provided by the ESP8266 chip on the NodeMCU board. Using the TCP/IP protocol, the ESP8266 from Espressif is a cheap WiFi chip [25]. The LEDs represent the (client) sensors, which in turn transmit data from the environment to the NodeMCU board (server), which transmits it to the gateway. The schematic is shown in *Fig. 11*.

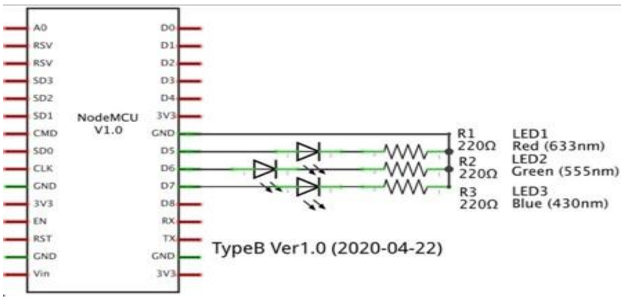
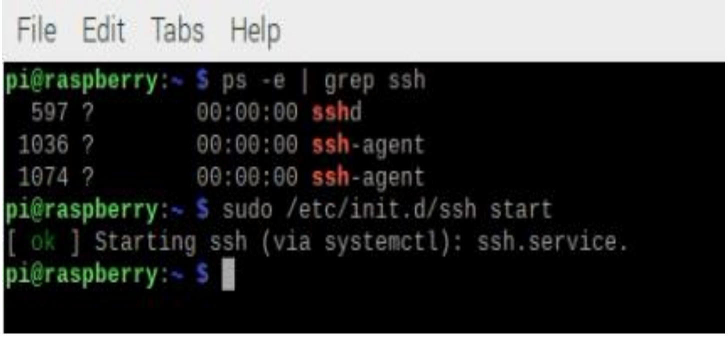


FIG. 11. SCHEMATIC DIAGRAM.

F. H. Using SSH Port Forwarding for Raspberry Pi's Outdoor Connection

This study made use of remote port forwarding, which is very similar to local port forwarding [26]. Remote SSH port forwarding is widely employed when a local machine lacks a public IP address and researchers require remote access to it. *Fig. 12* shows our Secure Shell (SSH) connection from a remote server to a local computer.



```
pi@raspberry:~ $ ps -e | grep ssh
  597 ?        00:00:00 sshd
 1036 ?        00:00:00 ssh-agent
 1074 ?        00:00:00 ssh-agent
pi@raspberry:~ $ sudo /etc/init.d/ssh start
[ ok ] Starting ssh (via systemctl): ssh.service.
pi@raspberry:~ $ 
```

FIG. 1. RESULT OF SSH REMOTE PORT FORWARDING.

V. EVALUATION AND SUMMARY

A. Evaluation of IoT device private network connection

This research was successful in its goal of designing and developing a security gateway that can effectively partition a private network from the Internet. Miniature Internet of Things gadgets can't talk to the wider internet without the RaspberryPi.

As an added security measure, small IoT devices employ dedicated IP addresses for internal networks, meaning that external networks cannot connect to the devices directly.

Received 31/August/2022; Accepted 09/November/2022

DOI: https://doi.org/10.33103/uot.ijccce.23.2.8

Security for over-the-air updates (OTA) is ensured in this study for cloud servers, tiny IoT devices, and the connectivity between cloud and small IoT devices [27].

B. Evaluation of Fail2ban

Brute-force attacks that log SSH attacks are among the most common types of attacks. This study explores fail2ban, and the results suggest that it reduces the number of attacks. An attacker trying to force SSH credentials connects to the system, tries the username and password, and then reconnects if they are incorrect [28]. Depending on the nature of this type of attack, the attacker is likely to attempt a large number of possible combinations; thus, the average number of attacks per IP address is likely to be much higher than for other types of attacks [29]. First, the attack size was reduced by 99.2%, and secondly, the average number of attacks per IP address decreased from 101.1 to 4.6, which is consistent with fail2ban rules that imply that IP addresses will be blocked after three failures. The quest is completed within twenty-four hours. Because of how long it takes to set up an SSH connection and how long it takes to actually block an IP address, some IP addresses may have more than three chances before they are blocked [30].

C. Evaluation of SSH Port Forwarding

Basically, all processes involving SSH data transmission are encrypted using symmetric keys. However, asymmetric encryption is used in the initial connection creation phase and the authentication handshake phase. The difference between asymmetric encryption and symmetric encryption is that in order to send data in a single direction, a related set of keys is required (a public key and a private key) [31]. In this research, we only share the public key and keep the private key strictly confidential and not disclosed to anyone. We have also set a password for the private key used for system authentication in SSH to prevent it from leaking. Therefore, SSH port forwarding is secure.

D. Evaluation of Wazuh

A simple test environment has been created to run tests to see if Wazuh meets the required security needs with the security gate. A Wazuh agent is installed on the security gate (RaspberryPi). This agent monitors the security gate while collecting logs and sending them to the Wazuh Manager. Wazuh Manager is installed on the Azure cloud. Elasticsearch and Kibana are also installed. Elasticsearch is a free and open-source search and analytics engine for all the types of data that Wazuh uses to process its logs. And Kibana is a data visualization and management tool from Elasticsearch, which also has a Wazuh plugin that allows it to visualize Wazuh logs and alerts. In addition, the open-source Filebeat software sends Wazuh logs to Elasticsearch. Additional details of wazuh are shown in *Fig. 13*.

Received 31/August/2022; Accepted 09/November/2022

DOI: https://doi.org/10.33103/uot.ijccce.23.2.8

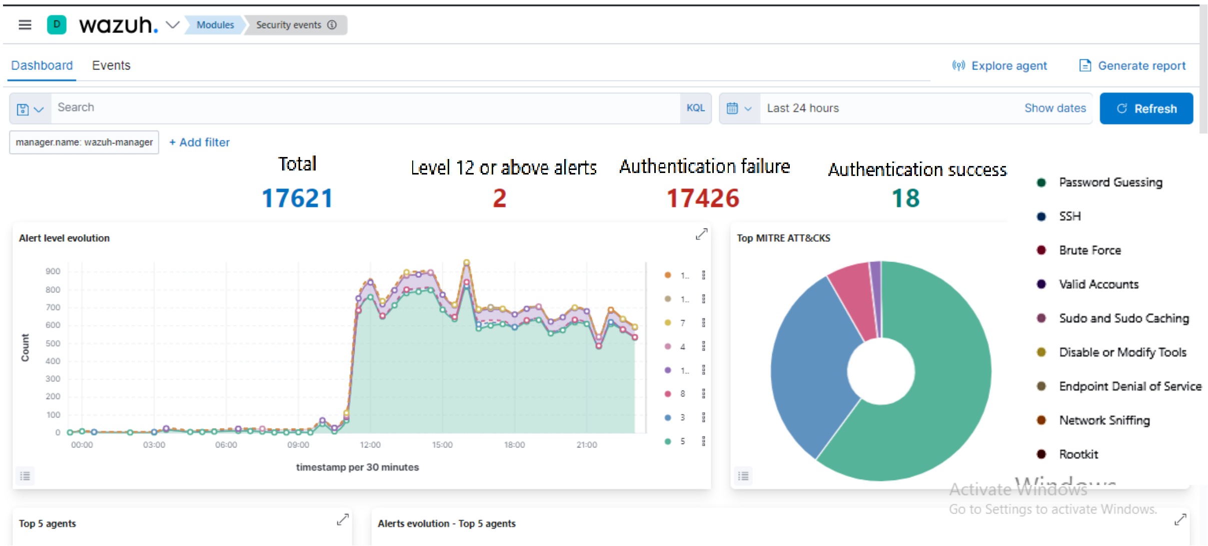


FIG. 13. THE WAZUH INTERFACE.

1. *Detecting an Denial of Service (DoS) attack*

This DoS test is carried out on the Wazuh agent device or on the Wazuh manager, as this attack is considered one of the most dangerous, as it floods the victim with a flood of requests, after which the victim falls and cannot continue to provide his services. It is expected that Wazuh will be able to correctly detect and identify the patterns of this attack. When wazuh alerts are seen, the monitoring and recording tool figures out what kind of attack it is and how serious it is. *Fig. 14* shows how to fill the victim's memory to show how the tool can find and accurately identify an electronic attack. *Fig. 15* also shows some information about the attack and its type.

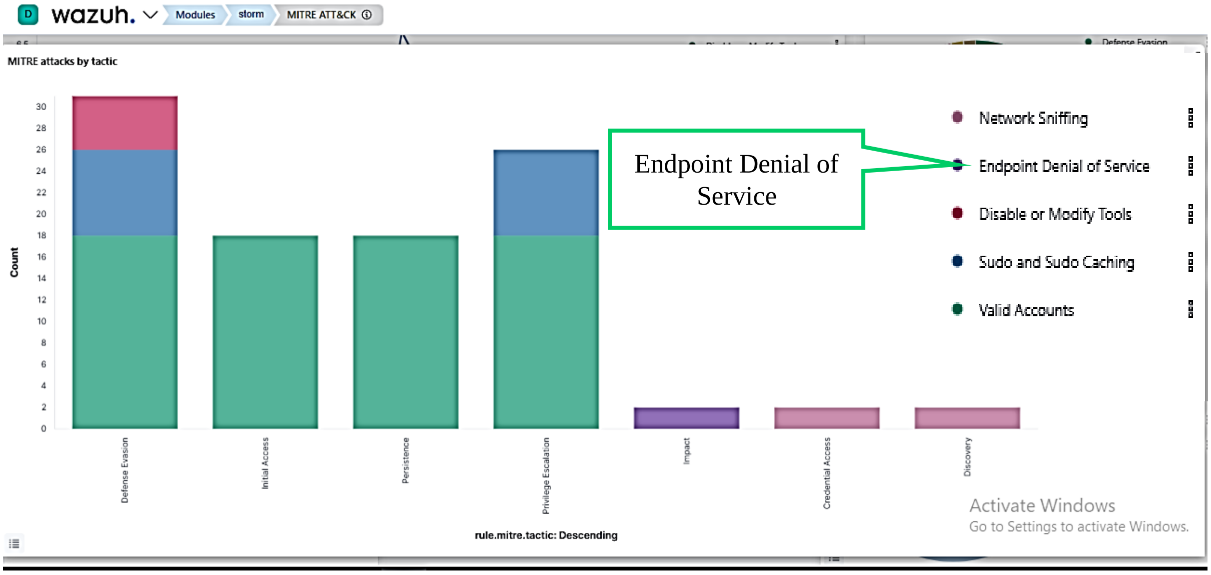
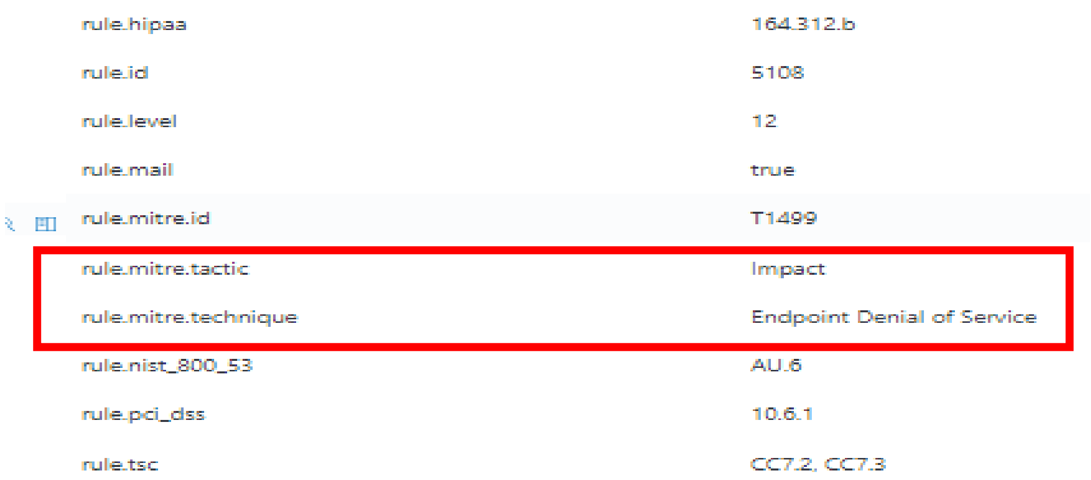


FIG. 14. REPRESENTED DENIAL OF SERVICE (DOS).

Received 31/August/2022; Accepted 09/November/2022

DOI: https://doi.org/10.33103/uot.ijccce.23.2.8



rule.hipaa	164.312.b
rule.id	5108
rule.level	12
rule.mail	true
rule.mitre.id	T1499
rule.mitre.tactic	Impact
rule.mitre.technique	Endpoint Denial of Service
rule.nist_800_53	AU.6
rule.pci_dss	10.6.1
rule.tsc	CC7.2, CC7.3

FIG. 15. DENIAL OF SERVICE ATTACKED.

2. *Detecting a Brute-force attack*

This test performs a brute-force attack on a device with the Wazuh agent installed. Its goal is to highlight the capabilities of a monitoring and logging tool to correctly detect and identify a cyberattack. After executing the previous command, it goes to Wazuh Alerts in the Security Events module of the Wazuh Kibana plug-in. There, it was detected that it was a response, as shown in *Fig. 16*, where the failed authentications are shown.

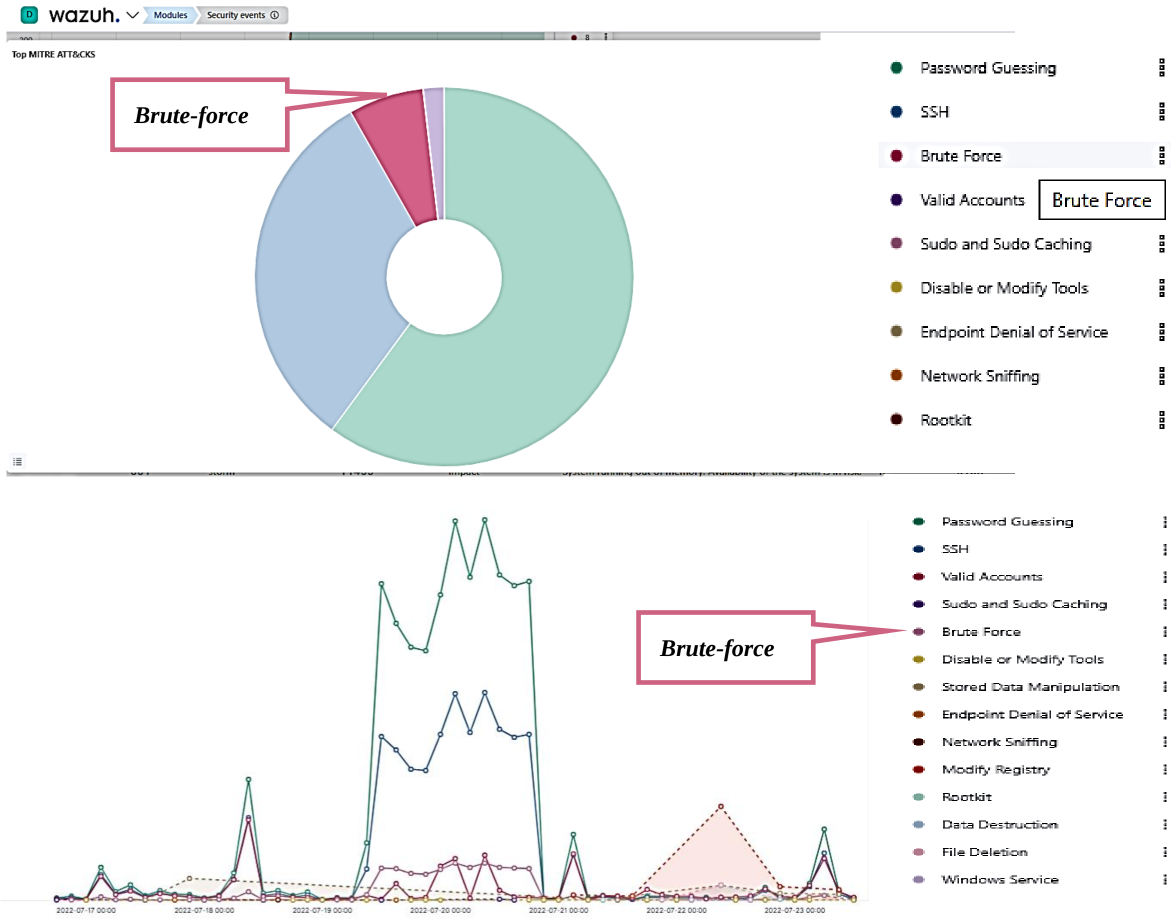


FIG. 16. THE BRUTE FORCE ATTACKED.

Received 31/August/2022; Accepted 09/November/2022

DOI: https://doi.org/10.33103/uot.ijccce.23.2.8

E. Comparison and Discussion

From the results of the previous work, it was found that the proposed system achieved a desirable performance in preventing and detecting various attacks such as DOS, DDOS, MITM, brute force, and dictionary attacks. The main contribution of this work is the implementation of the authentication model between IoT devices. And the application of the security portal compared to previous research Table I. shows a comparison between the proposed method and the latest relevant research.

TABLE I. COMPARISON BETWEEN THE PERFORMANCE OF THE PROPOSED SYSTEM AND THE PERFORMANCE OF THE IN PREVIOUS WORKS

Author (year)	Algorithm and protocol	IoT authentication	Experiment simulation	SIEM type	Cloud	Protection type
H. Sun, H. Yu, et al. [14]	ETCORA algorithm	No	fog-cloud and ETCORA algorithm	No	Yes	No
Tomas Zitta, et al. [15]	LLRP (Low Level Reader Protocol)	No	using Raspberry Pi 3 security to IDS/IPS	No	No	Detection System) and IPS (Intrusion Prevention
V. Teeraratchakarn and Y. Limpiyakorn [16]	No	No	Elastic Stack	ELK	No	IDS
F. Mulyadi, et al. [18]	No	No	Y	ELK + wazuh and docked	No	IDS
F. Balseca-Chávez, et al. [19]	JSON	No	(NIPS/HIDS)	Elasticsearch, Logstash and Kibana), Filebeat and Wazuh	No	IDS
proposed system (2022)	HTTP and JSON.	TFA by Token	Using ESP8266, (Security Settings on the Gateway for RaspberryPi	Elasticsearch, Logstash and Kibana), Filebeat and Wazuh	Azure	Authentication, IPS and IDS

VI. CONCLUSIONS

In order to make existing systems better, some researchers add small IoT devices without first figuring out how to keep them safe. If they use a small Internet of Things device as a microcontroller, the whole system will fail if the microcontroller is compromised. In light of this, our study uses a Security Settings on the Gateway for RaspberryPi to build a security gateway that can connect low-powered IoT devices back to the private network they came from. This keeps them from being connected to the public network.IoT devices that use little power can also have their firmware updated

Received 31/August/2022; Accepted 09/November/2022

DOI: https://doi.org/10.33103/uot.ijccce.23.2.8

remotely and safely over an isolated network.This article also looks at and compares how well different secure connections work for IoT devices to talk to each other.If they use token authentication instead of TFA, can avoid a man-in-the-middle attack because it adds another layer of security.Researchers must use the SSH port if they need to access a local computer from the outside world but that machine does not have a public IP address (i.e., the mapped port is on the remote server and the request comes from within the local network). Cloud services are a great choice for researchers who have to keep track of a large number of Internet of Things devices.From a safety point of view, this study could help make the Internet of Things safer for low-powered devices.

REFERENCES

[1] K. Routh and T. Pal, “A survey on technological, business and societal aspects of Internet of Things by Q3, 2017,” in Proceedings - 2018 3rd International Conference On Internet of Things: Smart Innovation and Usages, IoT-SIU 2018, 2018, doi: 10.1109/IoTSIU.2018.8519898.

[2] S. Narang, T. Nalwa, T. Choudhury, and N. Kashyap, “An efficient method for security measurement in internet of things,” in Proceedings of the 2018 International Conference On Communication, Computing and Internet of Things, IC3IoT 2018, 2019, pp. 319–323, doi: 10.1109/IC3IoT.2018.8668159.

[3] A. J. Cui, C. Li, and X. M. Wang, “Real-time early warning of network security threats based on improved ant colony algorithm,” in Proceedings - 2019 12th International Conference on Intelligent Computation Technology and Automation, ICICTA 2019, 2019, pp. 309–316, doi: 10.1109/ICICTA49267.2019.00072.

[4] H. H. Hassan and M. A. A. Khodher, “Data Hiding by Unsupervised Machine Learning Using Clustering K-mean Technique,” IRAQI J. Comput. Commun. Control Syst. Eng., vol. 21, no. 4, pp. 37–49, 2021.

[5] S. Chaudhary, R. Johari, R. Bhatia, K. Gupta, and A. Bhatnagar, “CRAIoT: Concept, Review and Application(s) of IoT,” in Proceedings - 2019 4th International Conference on Internet of Things: Smart Innovation and Usages, IoT-SIU 2019, 2019, doi: 10.1109/IoT-SIU.2019.8777467.

[6] N. S. Yamanoor and S. Yamanoor, “High quality, low cost education with the Raspberry Pi,” in GHTC 2017 - IEEE Global Humanitarian Technology Conference, Proceedings, 2017, vol. 2017-January, pp. 1–5, doi: 10.1109/GHTC.2017.8239274.

[7] H. Hassan, A. Kamal Taqi, H. J. Hassan, and K. Hadi, “Implementation of Wireless Body Area Network Based Patient Monitoring System,” J. Inf. Eng. Appl. , vol. 8, no. 4, pp. 51–64, 2018, [Online]. Available: www.iiste.org.

[8] N. A-hussein and A. D. Salman, “IoT Monitoring System Based on MQTT Publisher/Subscriber Protocol,” IRAQI J. Comput. Commun. Control Syst. Eng., vol. 20, no. 3, pp. 75–83, Jul. 2020, [Online]. Available: https://ijccce.uotechnology.edu.iq/article_168073.html.

[9] A. Assiri and H. Almagwashi, “IoT Security and Privacy Issues,” in 1st International Conference on Computer Applications and Information Security, ICCAIS 2018, 2018, doi: 10.1109/CAIS.2018.8442002.

[10] S. Cherrared, S. Imadali, E. Fabre, and G. Gössler, “Sakura a model based root cause analysis framework for VIMS,” in MobiSys 2019 - Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services, 2019, pp. 594–595, doi: 10.1145/3307334.3328642.

[11] J. Upadhyaya and N. J. Ahuja, “Quality of service in cloud computing in higher education: A critical survey and innovative model,” in Proceedings of the International Conference on IoT in Social, Mobile, Analytics and Cloud, I-SMAC 2017, 2017, pp. 137–140, doi: 10.1109/I-SMAC.2017.8058324.

[12] A. Sun, G. Gao, T. Ji, and X. Tu, “One Quantifiable Security Evaluation Model for Cloud Computing Platform,” in Proceedings - 2018 6th International Conference on Advanced Cloud and Big Data, CBD 2018, 2018, pp. 197–201, doi: 10.1109/CBD.2018.00043.

[13] S. Hu, H. Suzuki, Y. Kitaguchi, H. Ohno, and S. Sampalli, “Design, implementation and performance measurement of raspberry gate in the IoT field,” in ACM International Conference Proceeding Series, 2019, pp. 82–89, doi: 10.1145/3361821.3361827.

[14] H. Sun, H. Yu, G. Fan, and L. Chen, ‘‘Energy and time efficient task offloading and resource allocation on the generic IoT-fog-cloud architec?ture,’’ Peer Peer Netw. Appl., vol. 13, no. 2, pp. 548–563, Mar. 2020.

[15] T. Zitta, M. Neruda, and L. Vojtech, “The security of RFID readers with IDS/IPS solution using Raspberry Pi,” 2017 18th International Carpathian Control Conference, ICCC 2017.

[16] V. Teeraratchakarn and Y. Limpiyakorn, “Exploring Network Vulnerabilities for Corporate Security Operations,” in Information Science and Applications, Springer, 2020, pp. 341–351.

[17] N. Neshenko, E. Bou-Harb, J. Crichigno, G. Kaddoum, and N. Ghani, “Demystifying IoT Security: An Exhaustive Survey on IoT Vulnerabilities and a First Empirical Look on Internet-Scale IoT Exploitations,” IEEE Commun. Surv. Tutorials, vol. 21, no. 3, pp. 2702–2733, 2019, doi: 10.1109/COMST.2019.2910750.

Received 31/August/2022; Accepted 09/November/2022

DOI: https://doi.org/10.33103/uot.ijccce.23.2.8

[18] F. Mulyadi, L. A. Annam, R. Promya, and C. Charnsripinyo, “Implementing Dockerized Elastic Stack for Security Information and Event Management,” in 2020-5th International Conference on Information Technology (InCIT), 2020, pp. 243–248.

[19] F. Balseca-Chávez, A. M. Colina-Vargas, and M. A. Espinoza-Mina, “Identificación de amenazas informáticas aplicando arquitecturas de Big Data,” INNOVA Res. J., vol. 6, no. 3.2, pp. 141–167, 2021.

[20] B. K. Oleiwi, “Scouting and Controlling for Mobile Robot Based Raspberry Pi 3,” J. Comput. Theor. Nanosci., vol. 16, no. 1, pp. 79–83, 2019.

[21] P. Shen, Y. Qi, W. Yu, J. Fan, and F. Li, “OTA Measurement for IoT Wireless Device Performance Evaluation: Challenges and Solutions,” IEEE Internet Things J., vol. 6, no. 1, pp. 1223–1237, Feb. 2019, doi: 10.1109/JIOT.2018.2868787.

[22] S. Yoon and J. Kim, “Remote security management server for IoT devices,” in International Conference on Information and Communication Technology Convergence: ICT Convergence Technologies Leading the Fourth Industrial Revolution, ICTC 2017, 2017, vol. 2017-December, pp. 1162–1164, doi: 10.1109/ICTC.2017.8190885.

[23] C. C. Teng, J. W. Gong, Y. S. Wang, C. P. Chuang, and M. C. Chen, “Firmware over the air for home cybersecurity in the Internet of Things,” in 19th Asia-Pacific Network Operations and Management Symposium: Managing a World of Things, APNOMS 2017, 2017, pp. 123–128, doi: 10.1109/APNOMS.2017.8094190.

[24] H. A. Odat, A. Nsour, and S. Ganesan, “Firmware over the air adhoc network, FOTANET,” in IEEE International Conference on Electro Information Technology, 2015, vol. 2015-June, pp. 101– 106, doi: 10.1109/EIT.2015.7293326.

[25] D. D. Khudhur and M. S. Croock, “Application of Self-Managing System in Greenhouse with Wireless Sensor Network,” Iraqi J. Comput. Commun. Control Syst. Eng., vol. 21, no. 1, pp. 53–61, 2021, doi: 10.33103/uot.ijccce.21.1.5.

[26] M. D. Pratama, F. Nova, and D. Prayama, “Wazuh sebagai Log Event Management dan Deteksi Celah Keamanan pada Server dari Serangan Dos,” vol. 3, no. 1. pp. 1–7, 2022.

[27] S. L. Jurj, R. Rotar, F. Opritoiu, and M. Vladutiu, “White-box testing strategy for a solar tracking device using nodemcu lua esp8266 wi-fi network development board module,” in 2018 IEEE 24th International Symposium for Design and Technology in Electronic Packaging(SIITME), 2018, pp. 53–60.

[28] Szalachowski, S. Matsumoto, and A. Perrig, “PoliCert: Secure and flexible TLS certificate management,” in Proceedings of the ACM Conference on Computer and Communications Security, 2014, pp. 406–417, doi: 10.1145/2660267.2660355.

[29] Al-Rubaye, Saba, et al. "Industrial internet of things driven by SDN platform for smart grid resiliency." IEEE Internet of Things Journal 6.1 (2017): 267-277.

[30] M. P. Eve, “How to block distributed brute-force attacks against Wordpress using fail2ban,” martineve. com, 2015.

[31] Verma, Nidhi, and Apurva Jha. "Extending port forwarding concept to IoT." 2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN). IEEE, 2018.

Received 31/August/2022; Accepted 09/November/2022