Ethereum Smart Home for Denial of Service and Single Point of Failure

Bernawan Ikhsan Syahputra, Desi Marlena, Dimas Febriyan Priambodo and Arizal

***Abstract*— Blockchain is a digital transaction technology adopting the peer-to-peer concept. The implementation of blockchain on Internet of Things (IoT) aims to secure the possibility of potential attacks against devices or transactions taking place on the IoT system. At practical levels, blockchain uses smart contracts to automate programs according to predetermined terms and conditions. This research is aimed at implementing an ethereum-based smart home Smart Contract by modifying the device components, dashboards, and consensus used in Xu et al.'s research. The consensus modification was performed by using Proof of Authority (PoA) aiming to improve block verification performance on the system. The Denial of Service (DoS) attacks and Single Point of Failure (SpoF) vulnerability were performed to evaluate the proposed system. The evaluation was performed with TCP Flood Attack, with request packets of 81,519 packets on port 8545 and ICMP Floods by sending 11,481,703 PING packets. The attack caused some application services running on the Ethereum Node 3 to stop, but did not stop the geth application. As for the Single Point of Failure (SPoF) vulnerability, the Ethereum network is still running and there were no obstacles in the mining process or block verification.**

***Index Terms*—Blockchain, Denial of Service, Ethereum, Single Point of Failure, Smart Contracts.**

Original Research Paper
DOI: 10.53314/ELS2327050S

I. Introduction

Blockchain is a technology for recording digital transactions using the internet network based on a peer-to-peer concept [1]. Blockchain consists of a distributed database containing an organized and growing list of transactions and multiple server storages [2]. Blockchain distributed databases are governed by an agreement called consensus and have append-only characteristics so that transaction data can only be added but cannot be deleted or changed. [3]. Basically, the nature of blockchain is decentralized, distributed and immutable by using asymmetric algorithms and hash functions to provide strong cryptographic strength for authentication and data integrity. [4]. Blockchain implementation can be carried out using smart contracts which have characteristics, namely anti-counterfeiting and non-tampering so that it is a new technology that can bring about social change [5]. Integrating blockchain and IoT for security gave promising result [6].

Smart contracts are programming codes that run on the blockchain and run automatically according to predetermined or fulfilled terms and conditions [7]. Smart contracts were created with the aim of recording contracts in computer code form so as to eliminate the need for a third party [8]–[9]. As the development of smart contract technology is realized by creating Ethereum which utilizes blockchain technology so that it can solve various problems in real life, not only cryptocurrencies.

One of the uses of the Ethereum Smart Contract is on the Internet of Things (IoT). The IoT network is data-centric, that is, all data on the network is sent by the microcontroller device to the server so that it allows for potential attacks on devices or transaction data [10]. Some of the applications of blockchain in IoT are health systems, smart grids, supply chain management, smart cities and smart homes [11]–[12]. Research related to smart home systems based on Ethereum Smart Contract has been carried out by [13] to monitor the temperature of two different rooms and provide a comparison table of system performance against the block forming process. Subsequent research namely [14] implement the Ethereum Smart Contract on a smart home system to monitor room temperature and control the air conditioner automatically based on the temperature threshold value that has been input on the monitor dashboard by the user. On research [14] also provided a statement that a smart home system based on Ethereum Smart Contract could be a solution to the Distributed Denial of Service (DDoS) problem without providing a technical description of how the attack was tested. Therefore, Proof of Denial of Services (DoS) attacks needs to be done technically to provide appropriate evidence based on facts and data that occur in the field. Research by Anggun Mugi Mabruroh et al [31] provide another implementation regarding the ethereum Smart Contract on smart meters.

DDoS and DoS are one of the many dangerous threats on the internet network. This attack aims to consume the resources or bandwidth of legitimate users. The difference between the two is that DDoS attacks are carried out by several machines in a distributed and coordinated manner, while DoS attacks are carried out by only one machine [15]. DDoS and DoS attacks

Manuscript received on March 3rd, 2023. Received in revised form on June 27th, 2023 and July 24th 2023. Accepted for publication on August 24th, 2023.

B. I. Syahputra is with the Crypto Hardware Engineering, Cyber and Crypto Polytechnic, Bogor, Indonesia (e-mail: bernawan.ikhsan@bssn.go.id).

D. Marlena is with Crypto Hardware Engineering, Cyber and Crypto Polytechnic, Bogor, Indonesia (e-mail: desi.marlena@poltekssn.ac.id).

D. F. Priambodo is with the Cybersecurity Engineering, Cyber and Crypto Polytechnic, Bogor, Indonesia (phone: +6281226400058; e-mail: dimas.febriyan@poltekssn.ac.id).

Arizal is with Cybersecurity Engineering, Cyber and Crypto Polytechnic, Bogor, Indonesia (e-mail: arizal@poltekssn.ac.id).

are one of the causes of Single Point of Failure (SPoF) on a network. SPoF is damage or system failure caused by the system itself or attacks from other parties, causing the entire system to not work properly [16].

II. Related Works

The related works for this research are [11], which implemented smart contracts on the Ethereum network in a smart home system. Additionally, study [10] presents a prototype of an Ethereum-based smart home system that performs real-time temperature and humidity monitoring, and provides automated reactions such as LED lighting based on predefined situations. Furthermore, the subsequent study [21] discusses the conceptual design of implementing an Ethereum private blockchain in a Smart Home System (SHS) to regulate transactions and enable system monitoring by authorized parties.

This research implements Decentralized Smart Home System based on Ethereum Smart Contract according to research [13] with modifications to the components of the device, dashboard, and consensus used in the study. Study [13] uses a Proof of Work (PoW) consensus which has the characteristics of block verification speed depending on the strength or performance of the device used as a node miner. Proof of Authority (PoA) is a type of consensus that makes it possible to whitelist parties who serve as block verifiers on the network [17]. In contrast to PoW which implements a mathematical calculation settlement system (nonce) on blocks in a competitive manner between nodes. Consensus replacement is carried out using PoA consensus which aims to improve block verification performance in the system built as shown in the research [17]. The use of PoA consensus also has a few attack models because it can minimize forks on the Ethereum network that is being built [18]. This research is expected to be a solution to Denial of Service attacks and Single Point of Failure vulnerabilities by utilizing the Go-Ethereum (Geth) environment using the Clique client which implements PoA consensus.

III. Background

A. Blockchain

First discovered by Satoshi Nakamoto in 2008 as the basis for using Bitcoin, blockchain is a digital technology that combines cryptography, data management, and networks to support checking, executing or recording transactions between parties [19]. The blockchain structure consists of a group of blocks that are connected to each other using the hash method so as to form a chain of blocks [20]. The nature of the blockchain is append-only, immutable, and a chronological ledger that strings each block together using a cryptographic hash function [21]. Each block carries data or transactions that cannot be changed or modified because they are secured using cryptographic methods [22]. Parties that are members of the blockchain or nodes function as storage of all transaction data and forming blocks on the blockchain network [23].

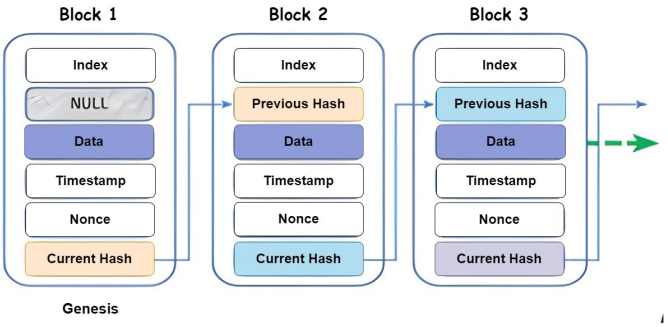


Fig. 1. Blockchain Structure [37]

Fig. 1 shows the structure of each block which consists of index, previous block hash, time stamp, nonce, and current hash. Genesis is the first block formed by a blockchain network. Every node that is connected in the blockchain network must first form the same genesis block as a condition of being part of the blockchain network. The timestamp is a description of the time and date when the block was formed, while the nonce is a 32-bit (4 byte) integer whose value controls the hash result calculated from the block [24]. One of the factors that causes a secure blockchain is its decentralized system or does not require a third party as an authorization center. On a blockchain, consensus is the rules that develop agreements between parties or nodes that do not trust each other. The concept of how blockchain forms an agreement or consensus is shown in Fig. 2. Proof of Work (PoW) is the consensus used in Bitcoin and Ethereum. The core of the consensus work is to perform mathematical calculations and hash power competition among other miner nodes to add blocks to the blockchain network. Miners who successfully complete block verification will receive rewards in the form of tokens. In this consensus, it is more profitable for nodes that have powerful computing devices because they affect mathematical calculations or block mining.

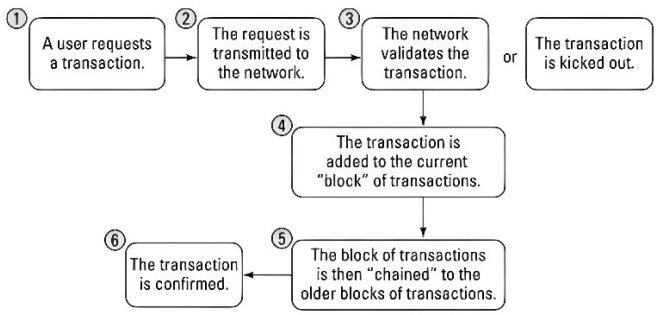


Fig. 2. Blockchain Work Process [36]

B. Internet of Things Architecture

The IoT architecture in general is shown in Fig. 3 which contains the perception layer, network layer, middleware layer, application layer, and business layer. The perception layer is a layer that contains IoT devices and sensors. These devices will send data to the middleware layer via the network layer. Data can be in the form of 2D, barcodes, RFID or sensor data such as infrared, temperature and humidity. At the network layer, you

can generally use Zigbee, Bluetooth, 3G, and Wifi as data transmission media to the middleware layer. The middleware layer uses a database to store data sent by sensors. The data will be sent to the server for processing. The application layer retrieves data from the middleware layer and integrates it with the smart app. The business layer is in charge of displaying data from the IoT system in the form of flowcharts or graphics received from the application layer [25].

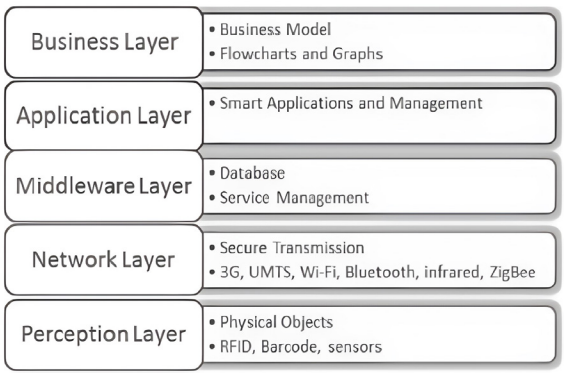


Fig. 3. IoT based architecture [38]

In conventional IoT, sensors function as receivers or collectors of information about what will be monitored, such as temperature data, images, smoke, sound, and so on. Sensor data that has been recorded will then be processed by a microcontroller or microprocessor device. After processing, the data is then sent by the IoT device to the central server using a communication line in the form of Bluetooth, wifi, or the internet. Furthermore, the central server will function as a basis for making decisions on the next step to determine the output in the form of moving actuators or monitoring data in real time [26].

The smart home system in this research will integrate blockchain and IoT based on modifications to the research of Xu et al. Blockchain systems that are distributed ledgers can change conventional IoT operations by using new network concepts, namely peer-to-peer and sustainable infrastructure to form social innovation [27]. In addition, the blockchain network also supports the use of devices or parties in the network of up to billions of devices without additional management or intermediary resources (brokers) [27]. In Fig. 4, an example of sending data on conventional IoT and blockchain-based IoT is given in the research that will be built.

In conventional IoT systems (a), sensor data in the form of digital or analog data is sent and processed on a microcontroller or microprocessor device to produce sensor data in the form of strings or characters. Furthermore, the gateway (router) in the system will send sensor data to the central server for decision making such as forwarding to the actuator, storing in the database or displaying on the monitor [27].

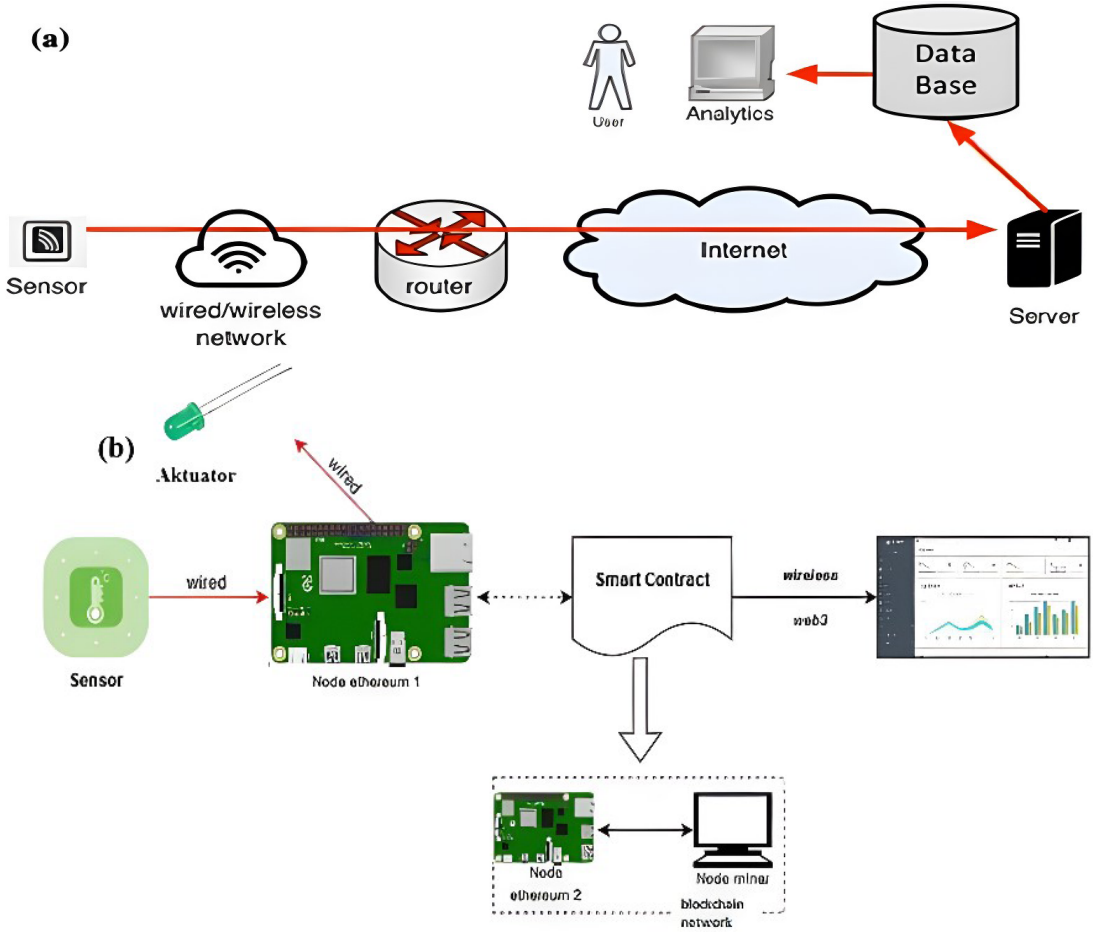


Fig. 4. Data transmission on the IoT system [27]

In blockchain-based IoT systems (b), sensors are sent to the microcontroller (node) to be processed into data in the form of strings or characters. Processed data is sent by calling the setData function on the smart contract using web3. All transactions or data sent in the smart contract are then sent to the blockchain network. When transaction data is displayed on a monitor or used to move actuators, the getData function on the smart contract will be called using web3. Blockchain implementation in IoT systems shows that data storage is not located on a central server but on all databases or parties involved in the network (nodes). So the use of blockchain in IoT systems provides added value [27].

TABLE I
COMPARISON OF CONVENTIONAL IOT AND BLOCKCHAIN BASED IOT [27]

Parameter	Conventional IoT	IoT + BC
Trust model	Centralized	Decentralized
Security/immutability	Low	High
Openness	Low	High
Privacy	High	High
Identities	Non-transferable	Transferable

Based on Table 1 shows the advantages of IoT integration with blockchain based on several parameters. Blockchain adheres to a decentralized trust model, which consists of a distributed database that has many storage servers and the concept of peer-to-peer or does not have a centralized authority [2].

The blockchain system uses asymmetric algorithms and hash functions to provide strong cryptographic strength for authentication and data integrity so as to ensure security and immutability of the system [4].

Transaction records on the blockchain are distributed in every database on the network so that all transactions that occur are open to all parties involved in the blockchain network to maintain data integrity [18]. The use of asymmetric algorithms and hash functions on the blockchain functions to encrypt and hash data so that the integrity of the data stored in each block can be maintained and remains valid [18].

All transactions that occur will be encrypted using the private key so that only parties who have the public key can identify the transaction as being from a trusted party.

C. Single Point of Failure

Single Point of Failure is damage or system failure caused by the system itself or attacks from other parties, causing the entire system to not work properly [16].

The current IoT infrastructure is still dominated by centralized systems, so that there are parties who control or store the system's central data. This makes IoT infrastructure such as smart buildings, smart cities, smart homes and so on to be vulnerable to SPoF. The principle of SPoF is that if there is one party that maintains a system, then if that party cannot carry out its duties, there will be changes or damage to the system [28].

D. Ethereum

The development of blockchain technology introduced by Satoshi Nakamoto through Bitcoin has attracted the attention of many people so that in 2013 a programmer named Vitalik Buterin introduced a new type of blockchain called Ethereum [29].

Ethereum is an open-source blockchain platform that enables anyone to build and deploy decentralized applications (Dapps) that run on the blockchain network [14].

Vitalik believes that blockchain technology can not only be used as a means of payment (bitcoin), but can be utilized in various decentralized systems. Decentralized applications are applications that can carry out transactions using smart contracts that are executed automatically according to predetermined conditions [14].

Ethereum is a platform that can create and run smart contracts on the blockchain network. Smart contracts use the blockchain as a database to collect and validate all transaction data including timestamped data and distribute it to all parties on the blockchain network [4].

The use of smart contracts is able to ensure that the actions taken are appropriate to the current conditions and the conditions set in the smart contract [30].

Fig. 5 provides an overview of how smart contracts work on a blockchain network. Before the smart contract is deployed to the network by the node miner, the contract will be compiled first by the Ethereum Compiler on the miner node so that Ethereum understands the contract. After compilation is complete, the smart contract will be sent to every node on the blockchain network so that all nodes on the network can send transactions on the blockchain network using the smart contract. Smart contract compilation data on the Ethereum Compiler generates bytecode in blocks. When a client (node) makes a transaction, the client will make a code invocation of the bytecode stored in the block. After the contract requirements are met, the smart contract bytecode will be executed in the EVM then the results of the contract will be updated in the next block.

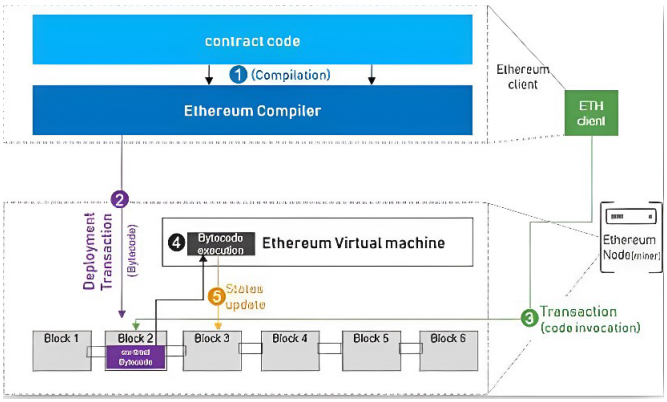


Fig. 5. How Smart Contracts work [36]

IV. METHOD

This research using System Development Life Cycle (SDLC) waterfall approach seen in Fig. 6. In the waterfall approach, each stage that will be carried out runs systematically,

and it must be ensured that everything needed must be fulfilled so that there is no possibility of repeating the previous stage. The advantage of the waterfall approach is that research becomes structured and system identification long before programming begins so as to minimize changes as the project progresses [31].

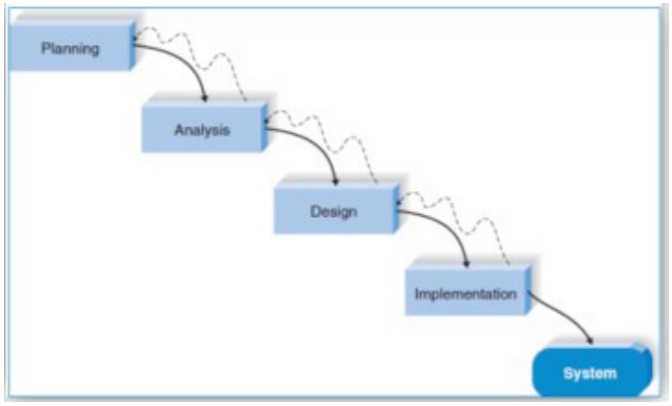


Fig. 6. SDLC Waterfall approach Model [28]

A. Planning

At the planning stage, a literature study was carried out based on reference papers, namely Xu et al., identifying projects to be developed based on reference papers, as well as analyzing devices and an overview of the system. Conventional smart home systems are vulnerable to various attacks, one of which is a Denial of Service attack. Based on subsequent literature studies it was found that one solution to the various problems of smart home is to implement blockchain in the system. Thus, it was continued with a literature study on blockchain solutions for smart home. It was found that research [13] and [14] is one of the studies implementing blockchain on smart home systems as a system security method and solution for Denial of Service (DoS) and Single Point of Failure attacks.

B. Design

Fig. 7 shows the system process flow in detail with the first step Admin using the Ethereum 1 Node to deploy smart contracts on the Ethereum network.

The process of deploying smart contracts is carried out after all nodes on the Ethereum network have mined. Raspberry Pi Node Ethereum 1 processes sensor data sent from the DHT11 sensor in room 1 then converts it into string data. The sensor data is then sent to the Ethereum network by calling the transaction function on the smart contract using web3. Sensor data that has been sent to the Ethereum network is then displayed on the monitoring system dashboard as room 1 sensor data by calling the call function on the smart contract using web3. Raspberry Pi Node Ethereum 2 processes sensor data sent from the Flame Sensor in room 2 and then converts it into string data in the form of flame sensor status. The sensor status is then sent to the blockchain network by calling the transaction function on the smart contract using web3. Sensor data that has been sent to the blockchain network is then displayed on the monitoring system dashboard as room 2 sensor data by calling the call function on the smart contract using web3. When the sensor data indicates a fire in the room, the Raspberry Pi will automatically sound

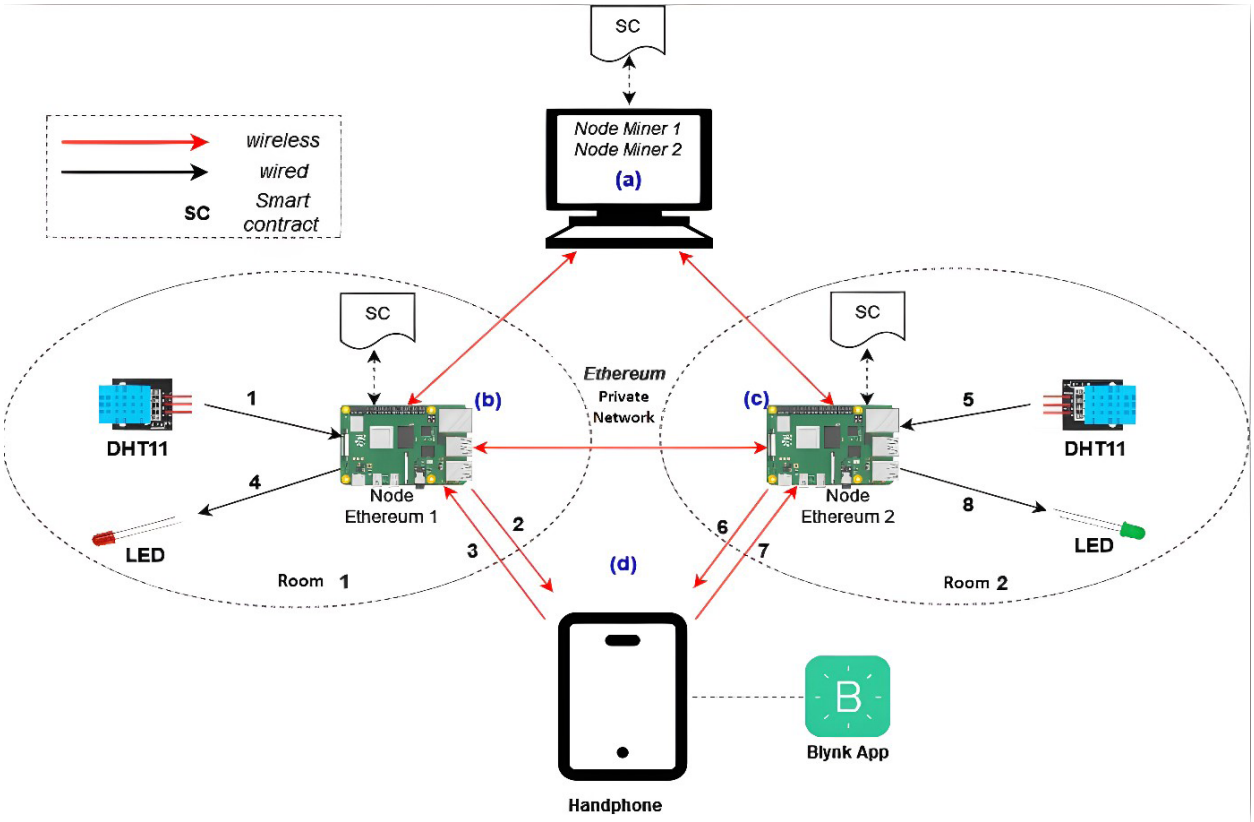


Fig. 7. System Overview

a buzzer. The dashboard/interface receives flame sensor data from the Raspberry Pi and displays it on the dashboard that has been created. When the admin will make changes to the threshold value, then fill in the threshold field on the dashboard/interface. The threshold value and current sensor value are then compared, when the current sensor value exceeds a predetermined threshold value it will automatically display the temperature status on the dashboard

DHT11 collects current temperature sensor data in room 1 and sends it to the Raspberry Pi to be processed and converted into string data. The string data is then processed on the Raspberry Pi to be sent to the smart home system. Changes to the threshold value on the dashboard are then processed and sent to the blockchain network by calling the setThreshold function on the smart contract using web3. String data that shows the current temperature value after being called on the smart home system is then displayed on the dashboard. The temperature threshold value changed by the user through the system dashboard is sent to the Raspberry Pi to be entered on the blockchain network using a smart contract. The threshold value is also compared with the current temperature value on the Raspberry Pi to turn on the actuator under certain conditions. When the current temperature value is higher than the predetermined threshold value, the Raspberry Pi will turn on the LED. In room 2, the Flame Sensor detects a fire in room 2 then the results are sent by the Raspberry Pi to be

processed and converted into string data. The string data is then processed on the Raspberry Pi to be sent to the smart home system and then displayed on the smart home system. String data that shows the current status of the room after being sent to the smart home system is then displayed on the dashboard. When the sensor status indicates there is a fire in the room, the Raspberry Pi will trigger the buzzer to sound.

C. Analysis

Analysis is carried out so that data is obtained about what needs are needed in making the system both functional and non-functional requirements as well as hardware and software requirements for the system to be built. Functional requirements are requirements that can be carried out by users of the system to be built besides functional requirements also show how the system must work, while non-functional requirements are requirements related to the characteristics, behavior and specifications that must be possessed by the system[31]

TABLE II
FUNCTIONAL REQUIREMENTS

No	Functional requirement
1	Admin can access the smart home dashboard/interface
2	Admin can change the threshold value on the dashboard/system interface
3	The system can display temperature sensor data in chart form
4	The system can show the status of the temperature sensor
5	The system can show the status of the flame sensor
6	The system can overcome Denial of Service attacks
7	The system can overcome the Single Point of Failure vulnerability

TABLE III
NONFUNCTIONAL REQUIREMENTS

No	NonFunctional Requirement
1	All nodes in the smart home system built are connected to each other in the Ethereum network
2	The system can record temperature data using the DHT11 sensor and send it to the smart home system.
3	The system can send the status of fire detection in the room using a flame sensor and send it to the smart home system.
4	Sistem can show the value of temperature in dashboard or interface.
5	The system can process temperature threshold values that are changed by the admin
6	The system can turn on the LED actuator when the temperature sensor data exceeds a predetermined threshold value.
7	The system can turn on the buzzer actuator when the flame sensor detects a fire in the room.
8	All processes on the system work in real time
9	The system can carry out the mining process to verify blocks on the Ethereum network being built

D. Implementation

At this stage an overview of the system is applied to hardware and software. The general description of the system was designed based on the reference paper research scheme and added several additional components at the design stage.

The result of the implementation phase is a smart home system that is in accordance with what was planned. the system was built using Proof of Authority (PoA) because it has better optimization in terms of CPU usage than using Proof of Work (PoA) consensus [17].

In addition, the use of PoW on Raspberry Pi is not possible due to limited hardware specifications while using PoA can still be done by using Raspberry Pi to be involved in the block verification process. Sensor and actuator programming languages are developed using Python and dashboard programming languages with a combination of HTML, CSS and Python no longer using the Blynk App as research by Xu et al. In addition, testing of the system that was built will also be carried out. The system testing will be carried out as follows:

a. Unit Testing
   Unit Testing aims to ensure that every function contained in the observable system goes according to plan [32]. This test is carried out on every component of the system that is built such as sensors, Raspberry Pi, actuators, laptops, and HTML web as a system monitor.

b. Integration Testing
   Integration Testing aims to determine the integration of the combination between the interface and the system and ensure the functions of the system can run properly [33].

c. System Testing
   System Testing focuses on testing the system against the programs that are built. System testing is also carried out to determine whether functional and non-functional requirements are met [34].

d. Performance Testing
Performance Testing aims to determine the performance of the system that has been built. In this study, performance testing was carried out to test the delivery time of temperature data and flame sensor status on the smart home system that has been built.

e. Security Testing
Security Testing aims to test the security of a system that is built based on mapped vulnerabilities [35]. In this study, testing will be carried out by carrying out attacks on Single Point of Failure and Denial of Service (DoS) vulnerabilities in the form of TCP Floods and ICMP Floods attacks on the Ethereum network being built.

V. Result and Discussion

A. Unit Testing

For software testing, the Go Ethereum, Thonny IDE, and Remix Ethereum IDE applications were used, compiled in Table 4. For hardware testing, it was carried out on DHT 11 Sensors, Flame Sensors, LEDs, and buzzers shown in Table 5.

TABLE IV
Software Testing

No	Perangkat Lunak	Keterangan
1	Go Ethereum	Succeed
2	Thonny IDE	Succeed
3	Remix Ethereum IDE	Succeed

TABLE V
Hardware testing

No	Perangkat Keras	Keterangan
1	DHT11 Sensor	Succeed
2	Fflame Sensor	Succeed
3	LED	Succeed
4	Buzzer	Succeed

B. Integration Testing

Integration testing consists of four discussions, namely integration of the DHT11 sensor in the smart home system, integration of the flame sensor in the smart home system, integration between the DHT11 sensor and LED and integration between the flame sensor and buzzer. Fig. 8 (a) shows the appearance of the DHT11 program that runs on the Ethereum 3 Node device and can detect temperature in real time accompanied by sending recorded temperature data on the smart home system.

The success of sending temperature data on the smart home system can be seen by looking at the display of the running mining process. Fig. 8 (b) shows several times the "Submitted Transaction" status appears. This is an indicator that the temperature data transaction was successfully sent. In line with this, the flame sensor program running on the Ethereum 4 Node device immediately sends the fire detection status to the smart home system. And shows "Submitted Transaction" which means that the status of the fire detection on the flame sensor has been successfully sent.

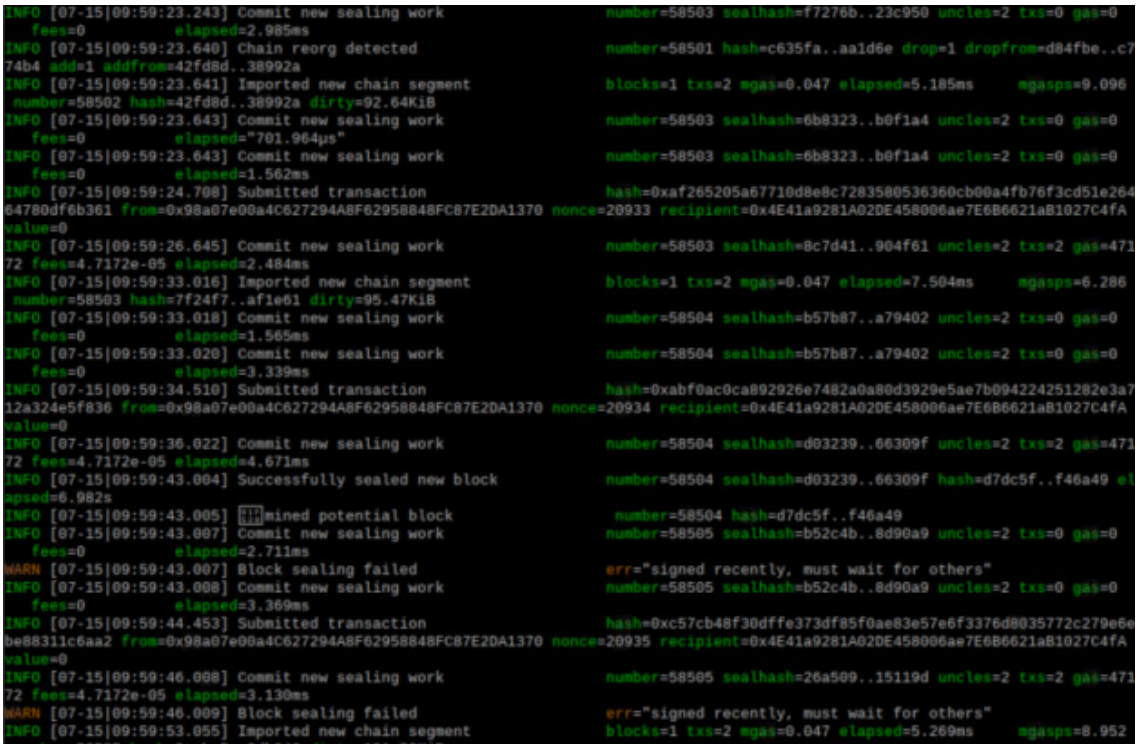


Fig. 8. DHT output and mining program on Ethereum 3 node (b)

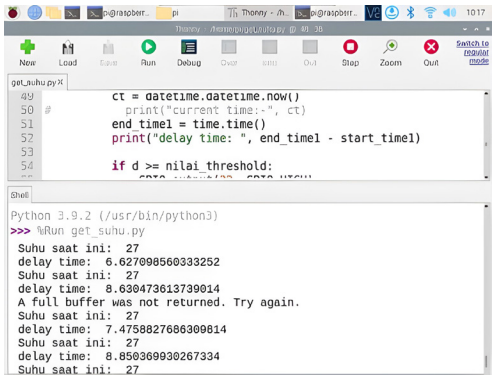


Fig. 8. DHT output and mining program on Ethereum 3 node (a)

C. System Testing

TABLE VI
CONFORMITY SET THRESHOLD ON THE DASHBOARD

Pengujian	Keterangan	Delay
1	Succeed	8,66
2	Succeed	11,41
3	Succeed	9,42
4	Succeed	10,10
5	Succeed	8,67
6	Succeed	7,86
7	Succeed	9,19
8	Succeed	12,30
9	Succeed	11,25
10	Succeed	5,51
	Average	9,437

This test is a large scope of integration testing to see conformity to the functional requirements that have been made before. testing begins with running the web server script on the Ethereum Node 1. After the script is run, it will show the dashboard ip address on the terminal. The next system test is to do a scenario of changing the temperature threshold on the dashboard. The test was carried out 10 times with additional information given when the time changed after the new threshold value was added as shown in Table 6.

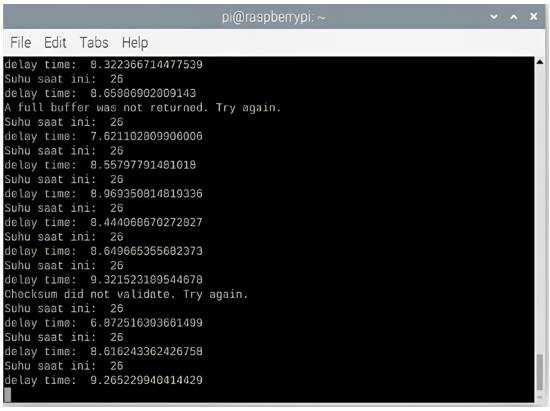
Testing the display of the chart on the dashboard is also carried out by observing and comparing the temperature values that come out when the program is run on the Thonny IDE with the temperature data chart displayed on the dashboard. The test was carried out based on 10 samples of recorded temperature values. Fig. 9 shows a comparison display of temperature values recorded on the terminal with the chart displayed on the dashboard. Based on the results of the system testing carried out, the functional and non-functional requirements in Table 2 and Table 3 were successfully fulfilled as a whole.

D. Performance Testing

Performance testing is carried out by analyzing the time of sending temperature data and the status of the flame sensor on the smart home system that has been built. Tests were carried out using the time.time() syntax in the DHT11 sensor program.

TABLE VII
TIME FOR SENDING DATA

Testing	Temperature Sensors	Flame Sensor
1	6,627098560333252	8,450021982192993
2	8,418138265609741	9,124223709106445
3	9,376110553741455	8,585147619247437
4	8,56318187713623	9,656146764755249
5	9,200273275375366	8,708126783370972
6	8,616291999816895	8,805291175842285
7	8,546558380126953	9,412282228469849
8	7,340145826339722	9,543951749801636
9	8,481131076812744	8,225429773330688
10	8,75612473487854	8,426900148391724



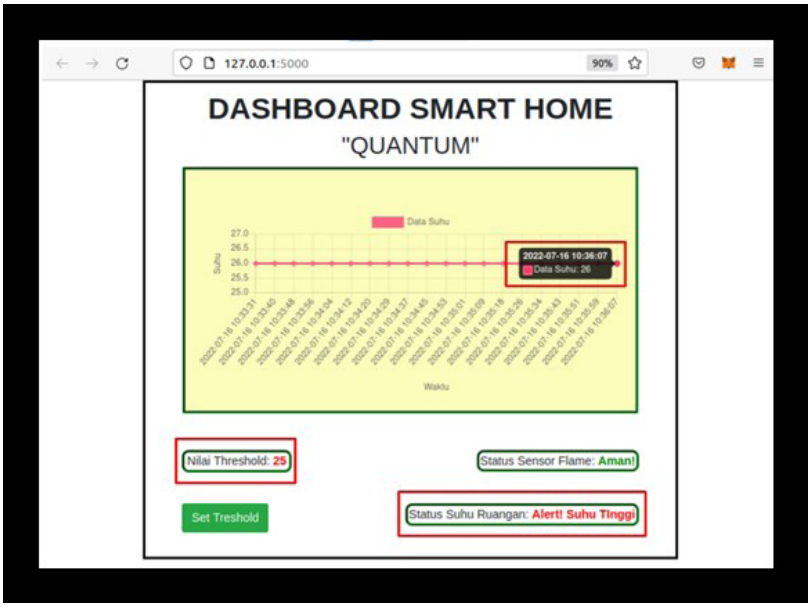


Fig. 9. DHT output and temperature on dashboard

Table 4 shows the results of testing the average time needed to send DHT11 sensor data and flame sensors to smart home systems. Data is taken based on 10 samples of the time of sending recorded temperature data shown in Table 7.

E. Security Testing

1) Denial of Service (DoS) Attack

DoS testing is carried out using two types of attacks, namely TCP Flood Attack and ICMP Flood Attack. The DoS attack scenario is aimed at the Ethereum 3 Node device and then an analysis of the results of the test is carried out. Furthermore, in the SPoF test, a scenario of turning off one of the nodes in the smart home system alternately is carried out, then observations are made of mining activities on each node to find out whether the built system can still run or not and observations are made on the dashboard to find out whether when one of the sensors off, the other sensors will also die.

```
┌──(berna1㉿kali)-[~]
└─$ nmap -p 8545 192.168.158.91
Starting Nmap 7.91 ( https://nmap.org ) at 2022-07-28 10:00 WIB
Nmap scan report for 192.168.158.91
Host is up (0.041s latency).

PORT     STATE SERVICE
8545/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 0.09 seconds
```

Fig. 10. Results of Scanning port 8545 status open

Fig. 10 shows a sample of devices that show open status when scanning on port 8545. The results of scanning all hosts on the network show that devices that have open status on port 8545 are devices with IP addresses 192.168.158.91, 192.168.158.121, 192.168.158.125, and 192.168.158.7. Denial of Service attack scenarios that are carried out using TCP Floods and ICMP Floods Attack types with the attack object being an Ethereum 3 Node device with IP 192.168.158.91.

The TCP Floods attack uses the hammer.py script obtained from https://github.com/cyweb/hammer. When executed, this script will send TCP packets to the Ethereum 3 Node with port 8545. Before carrying out the attack, monitoring of packets sent and received by the geth application is carried out for 1 minute. Fig. 11 is the result of calculating the size of packets sent and received by the geth application on port 8545 using the nethogs application.

```
┌──(root💀kali)-[/home/berna1/senjata/hammer]
└─# python3 hammer.py -s 192.168.158.91 -p 8545
192.168.158.91  port:  8545  turbo:  135
Please wait...
Thu Jul 28 15:22:36 2022   <--packet sent! hammering-->
Thu Jul 28 15:22:36 2022   <--packet sent! hammering-->
Thu Jul 28 15:22:36 2022   <--packet sent! hammering-->
```

Fig. 12. TCP Floods Attack on Target

Fig. 12 shows the display of the attack using the hammer.py script which targets the IP address of the Ethereum 3 Node with destination port 8545 for 1 minute as many as 81,519 packets. Fig. 13 shows the display of packet traffic using wireshark tools when the attack is carried out.

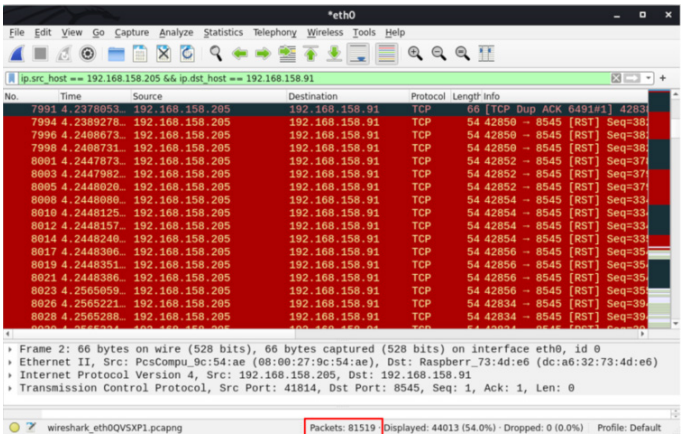


Fig. 13. TCP Floods Attack on the Wireshark Application

Fig. 14. Network Traffic After a TCP Floods Attack

TCP Floods attack caused an increase in the number of packets sent and received by the geth application on the Ethereum 3 Node device. Based on Fig. 11 and Fig. 14 it can be seen that the difference in packets received by the application before and after the attack was 2,887.99 MB while the difference in packets received was 4,074 .989 MB. This is because the TCP

Fig. 11. Traffic of TCP Floods attack

Floods attack floods port 8545 which causes network traffic to be fuller than before the attack. However, the results of the TCP Floods attack based on the scenario carried out did not affect the block verification or mining process as evidenced by observing the time of sending temperature data to the smart home system. During the TCP Floods attack, it succeeded in sending temperature data to the smart home system 6 times in one minute with an average of 8.29 seconds. This does not differ too much based on the performance test that was carried out previously, namely for 8.39 seconds. Thus, it can be concluded that the geth application on the Ethereum 3 Node has no impact on the TCP Floods attack scenario that is carried out.

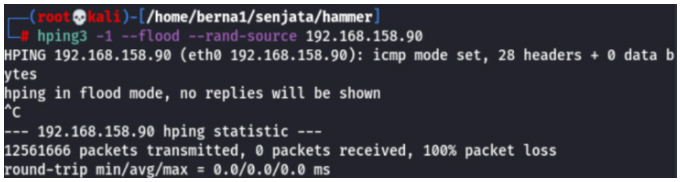


Fig. 15. ICMP Floods attack on the Ethereum 3 Node

The ICMP Floods Attack testing technique is sending a large number of ICMP echo request (PING) packets using the hping3 tool. Fig. 15 shows the appearance of the ICMP Floods Node Ethereum 3 attack by sending an ICMP echo request (PING) packet. Fig. 13 also shows the number of packets transmitted by the attacker device against the Ethereum 3 Node device for 1, which is 12,561,666 packets. The result of the attack is an increase in the sending of PING packets in large numbers to the Ethereum 3 Node as shown in Fig. 16.

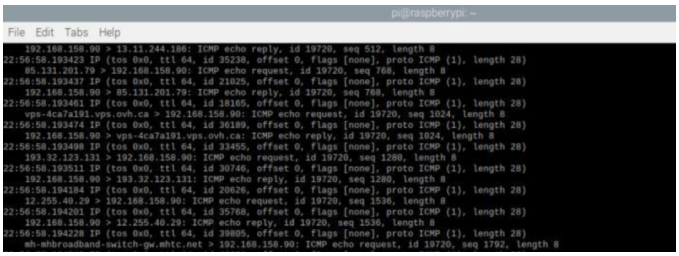


Fig. 16. Network Traffic Scanning on Ethereum 3 Node

The impact of the ICMP Floods can be seen that VNC cannot be accessed during the ICMP Floods attack, thus showing that the ICMP Floods attack affects network traffic on VNC applications that use port 5900. Furthermore, observations are made on the effect of geth applications when the ICMP Floods attack is carried out. Fig. 17 shows the appearance of a node that is still connected to the Ethereum 3 network (192.168.158.90) still connected to the smart home system when the ICMP Floods attack is carried out. So the results of testing the ICMP Floods attack show that the geth application service is still running even though an attack scenario is carried out on the Ethereum 3 Node device.

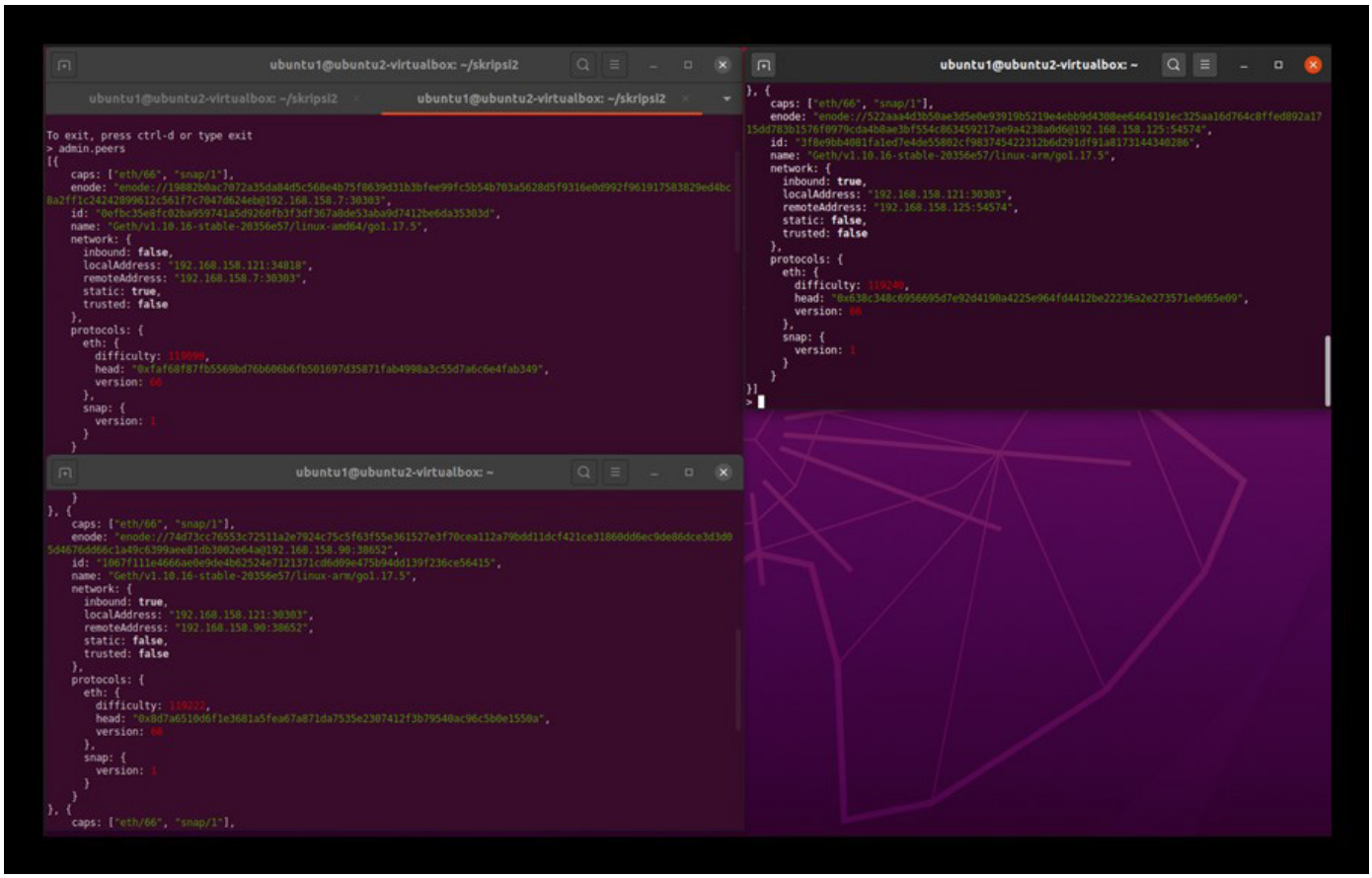


Fig. 17. DHT output and temperature on dashboard

2) Single Point of Failure (SPoF) Test

The scenario that is carried out is by turning off one of the nodes to prove that the system is still running even though one of the nodes is turned off. The process of turning off the node is done by gradually shutting down the VM Node Ethereum 1 to 4 devices. The SpoF Node Ethereum 1 scenario proves that the system is still running even though the Ethereum Node 1 is turned off as shown in Fig. 18. The SPoF test on the Ethereum 1 Node is not proven by observing the dashboard, this is because the flask server is running on that Node. Observations on the sensor show that the LED can still light up when the temperature exceeds the threshold and the buzzer can sound when it detects a fire around the sensor. This is because the threshold can only be changed via the dashboard so that the threshold that is used as a reference is the last threshold inputted.

Testing by turning off the Ethereum 2 Node device can be seen in Fig. 19 which shows that the system built can still run and continue to carry out the mining process even though one of the nodes is turned off. Fig. 19 also shows the system dashboard which still displays all sensor data in real time even though the Ethereum 2 Node device is turned off.

In the test of turning off the Node Ethereum 3 dashboard in Fig. 20 after the Node Ethereum 3 device is turned off it shows that the temperature chart is not running, this is because there is no input of temperature data by the Node Ethereum 3 device which has been turned off. However, the dashboard can still be accessed and the flame sensor can still detect fire in the room and trigger the buzzer to sound.

The last scenario in the SPoF test is to turn off the Raspberry Pi 2 as an Ethereum 4 Node. The results show that the system

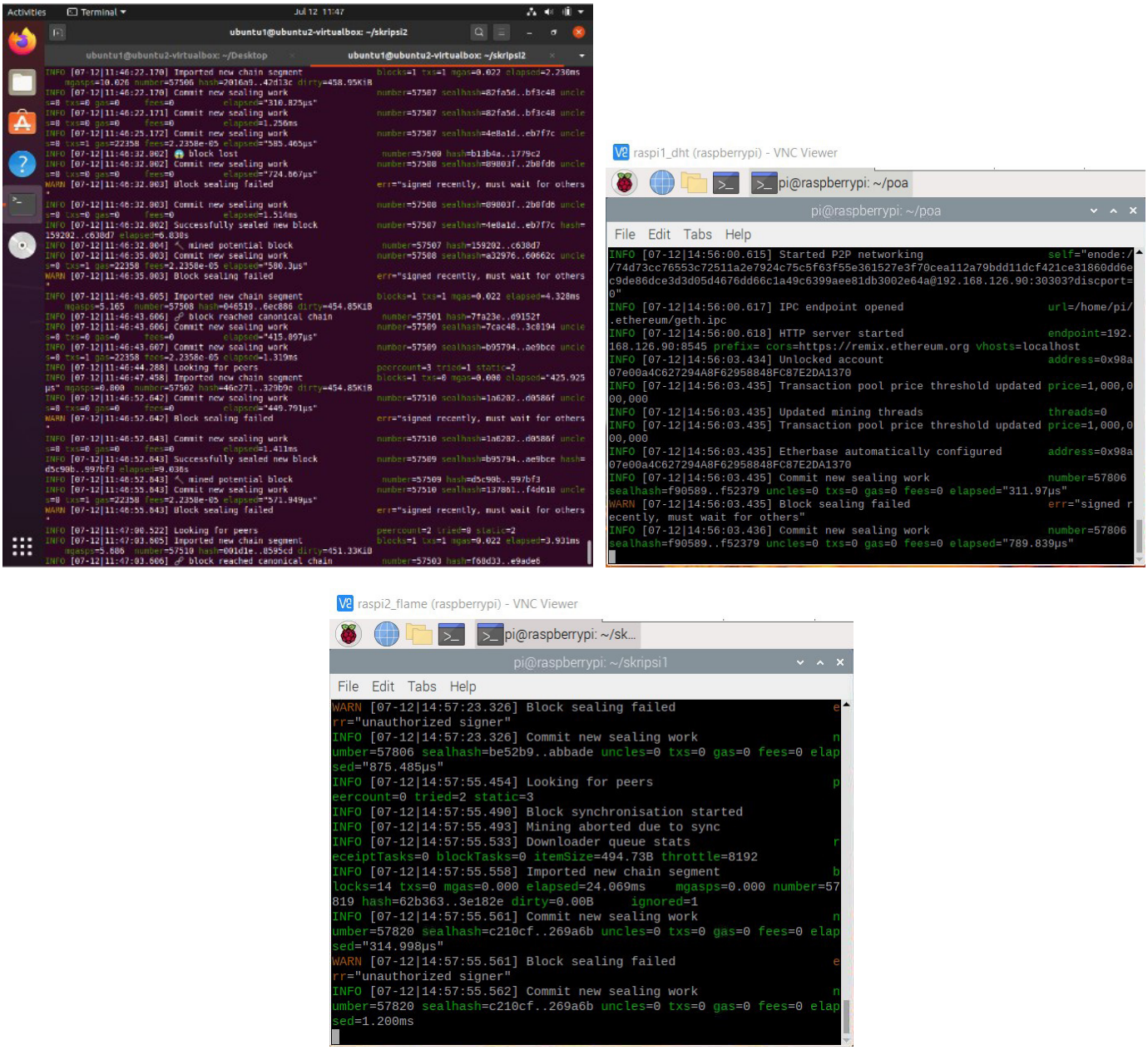


Fig. 18. Mining Process Continues to Run After Ethereum Node 1 Turns Off

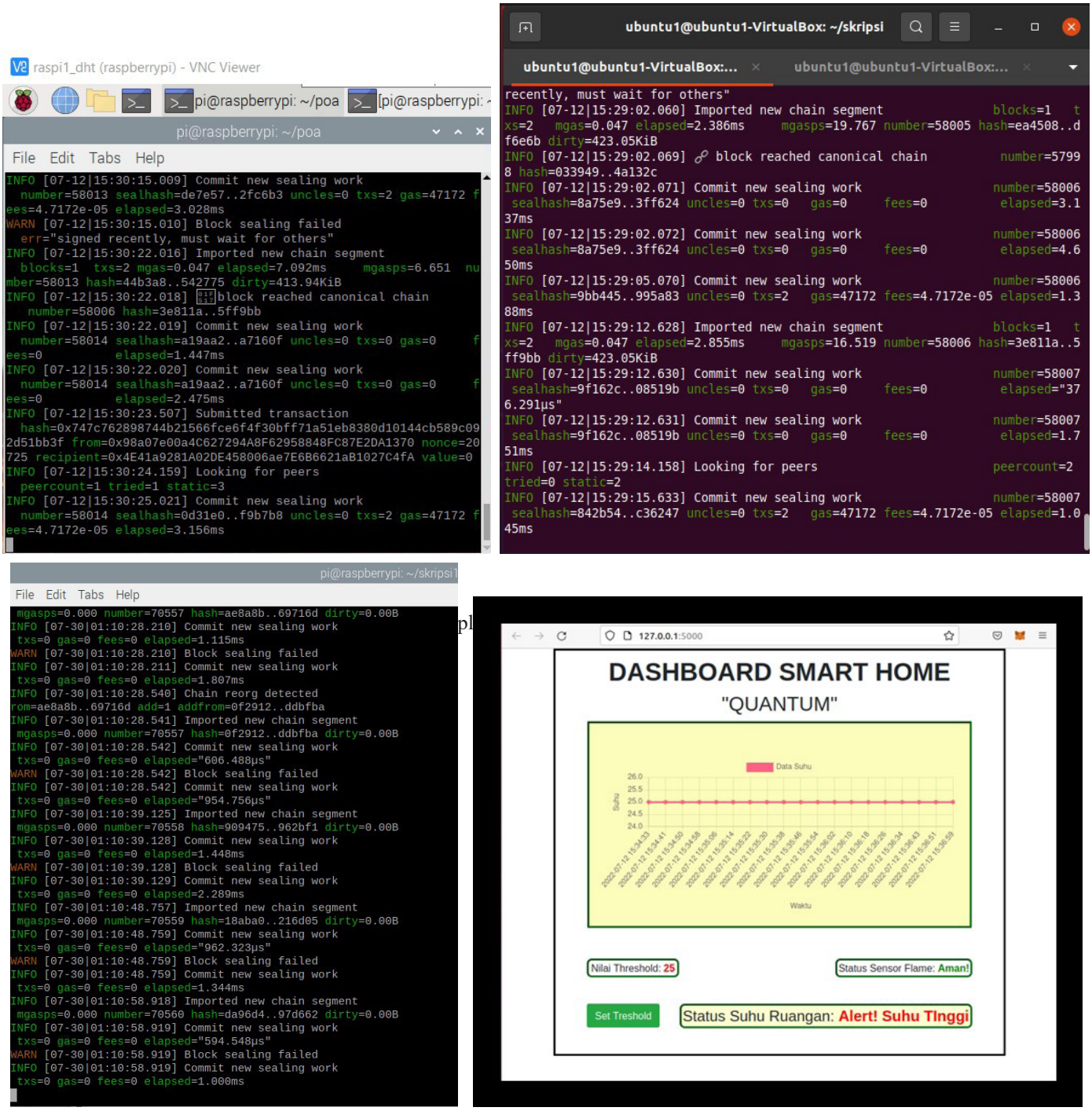


Fig. 19. Mining Process Display and dashboard when testing is carried out (b)

built can still carry out the block verification process (mining) even though one of the nodes in the system is turned off. Fig. 21 shows the dashboard view when the Ethereum 4 Node device is turned off. flame sensor status shows "?" which means there is no flame sensor status data displayed on the dashboard. This is because the Ethereum 4 Node device has been shut down. Furthermore, it was observed that other systems such as charts and the set threshold function on the dashboard were still running as usual. In addition, the LED still lights up when the temperature exceeds the threshold value.

VI. Conclusion

Denial of Service (DoS) attack with the type of TCP Floods attack using the hammer.py script based on the scenario that has been carried out targeting Raspberry Pi devices (Node Ethereum 3) with port 8545 (Ethereum port) by sending 81,519 request packets within one minute indicating the existence increased traffic as shown in the nethogs and wireshark applications but did not stop the geth program running on the Ethereum 3 Node. Furthermore, DoS attacks with the ICMP Floods attack

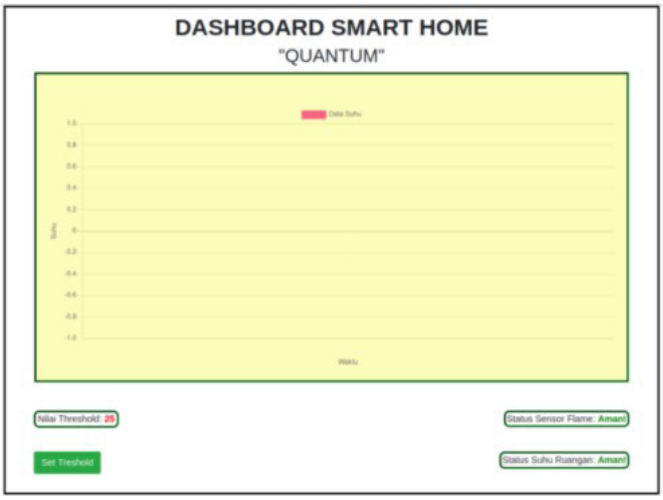


Fig. 20. Dashboard after Testing Ethereum 3 Node

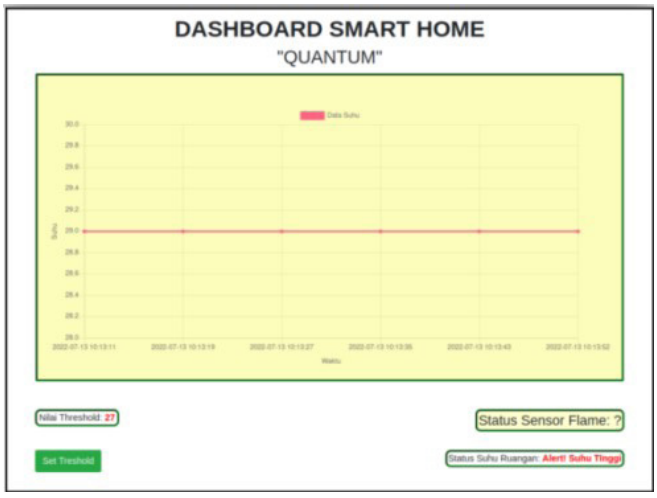


Fig. 21. Dashboard after Testing Ethereum 4 Node Turned off

type are carried out by sending a large number of ICMP echo request or PING packets on as many as 12,561,666 packets that cause some services with open ports to stop such as the VNC application. However, the attack does not affect the Geth application so the application continues to run on the Ethereum 3 Node while the ICMP Floods attack is running. Security testing with Single Point of Failure (SPoF) proof shows that the block verification process is not affected when one of all nodes is turned off. So that the smart home system continues to run. Performance testing on a smart home system based on the Ethereum Smart Contract that was built has a long time for sending temperature data of 8.39 seconds and a long time for sending flame sensor status of 8.89 seconds.

VII. Future Works

The potential future works include the development of a smart home system using smart contracts, specifically focusing on the creation of more sophisticated smart contracts to regulate various aspects within a smart home. Enhancing the functionality of Ethereum-based smart home systems can be achieved by introducing new features and integrating additional sensors to enable richer automated responses. The implementation of Ethereum private blockchain in smart home systems can be explored further, emphasizing the need for enhanced security measures. Performance analysis of Ethereum-based smart home systems can be conducted to assess efficiency and responsiveness in managing tasks and transactions. Moreover, investigating the integration of other technologies such as the Internet of Things (IoT) and Artificial Intelligence (AI) presents an intriguing research area to enhance connectivity, interoperability, and capabilities of smart home systems.

References

[1] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends," *Proc. - 2017 IEEE 6th Int. Congr. Big Data, BigData Congr. 2017*, no. October, pp. 557–564, 2017,

[2] A. B. Goldstein, N. A. Sokolov, V. S. Elagin, A. V Onufrienko, and I. A. Belozertsev, "Network characteristics of blockchain technology of on board communication," in *2019 Systems of Signals Generating and Processing in the Field of on Board Communications*, 2019, pp. 1–5.

[3] B. Warburg, B. Wagner, and T. Serres, *Basics of Blockchain: A Guide for Building Literacy in the Economics, Technology, and Business of*. Animal Venturs LLC, 2019.

[4] M. Shurman, A. A.-R. Obeidat, and S. A.-D. Al-Shurman, "Blockchain and Smart Contract for IoT," in *2020 11th International Conference on Information and Communication Systems (ICICS)*, 2020, pp. 361–366.

[5] X. Yang, J. Liu, and X. Li, "Research and Analysis of Blockchain Data," *J. Phys. Conf. Ser.*, vol. 1237, no. 2, 2019,

[6] M. S. Mahmood and N. B. Al Dabagh, "Blockchain technology and internet of things: review, challenge and security concern," *Int. J. Electr. Comput. Eng.*, vol. 13, no. 1, pp. 718–735, 2023,

[7] H. Sheikh, R. M. Azmathullah, and F. Rizwan, "Smart Contract Development, Adoption and Challenges: The Powered Blockchain," *Int. Res. J. Adv. Eng. Sci.*, vol. 4, no. 2, pp. 321–324, 2019.

[8] S. Wang, Y. Yuan, X. Wang, J. Li, R. Qin, and F.-Y. Wang, "An Overview of Smart Contract: Architecture, Applications, and Future Trends," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, 2018, pp. 108–113.

[9] IBM, "What are smart contracts on blockchain?" https://www.ibm.com/topics/smart-contracts (accessed Feb. 10, 2023).

[10] P. Sajana, M. Sindhu, and M. Sethumadhavan, "On blockchain applications: hyperledger fabric and ethereum," *Int. J. Pure Appl. Math.*, vol. 118, no. 18, pp. 2965–2970, 2018.

[11] X. Wang *et al.*, "Survey on blockchain for Internet of Things," *Comput. Commun.*, vol. 136, pp. 10–29, 2019.

[12] S. Ganesh Kumar, B. Sriman, A. Murugan, and B. Muruganantham, "IoT-smart contracts in data trusted exchange supplied chain based on block chain," *Int. J. Electr. Comput. Eng.*, vol. 10, no. 1, pp. 438–446, 2020,

[13] Q. Xu, Z. He, Z. Li, and M. Xiao, "Building an Ethereum-Based Decentralized Smart Home System," *Proc. Int. Conf. Parallel Distrib. Syst. - ICPADS*, vol. 2018-Decem, pp. 1004–1009, 2019,

[14] A. Qashlan, P. Nanda, and X. He, "Automated ethereum smart contract for block chain based smart home security," in *Smart Systems and IoT: Innovations in Computing*, Springer, 2020, pp. 313–326.

[15] E. Džaferović, A. Sokol, A. A. Almisreb, and S. Mohd Norzeli, "DoS and DDoS vulnerability of IoT: A review," *Sustain. Eng. Innov.*, vol. 1, no. 1, pp. 43–48, 2019,

[16] G. Lynch, *Single Point of Failure: THE TEN ESSENTIAL LAWS OF SUPPLY CHAIN RISK MANAGEMENT*. 2009.

[17] D. A. R. Pérez, "Study on Blockchain Technologies for Modern Enterprise Applications," *Comput. Eng.*, 2018, [Online]. Available: http://prcrepository.org:8080/xmlui/handle/20.500.12475/217

[18] X. Liu *et al.*, "MDP-Based Quantitative Analysis Framework for Proof of Authority," in *2019 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, 2019, pp. 227–236.

[19] X. Xu, I. Weber, and M. Staples, *Architecture for blockchain applications*. Springer, 2019.
[20] B. Shrimali and H. B. Patel, "Blockchain state-of-the-art: architecture, use cases, consensus, challenges and opportunities," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 9, pp. 6793–6807, 2022,
[21] I. M. Al-Joboury and E. H. Al-Hemiary, "Automated decentralized IoT based blockchain using ethereum smart contract for healthcare," in *Enhanced Telemedicine and e-Health*, Springer, 2021, pp. 179–198.
[22] Z. Qi, Y. Zhang, Y. Wang, J. Wang, and Y. Wu, "A Cascade Structure for Blockchain," *Proc. 2018 1st IEEE Int. Conf. Hot Information-Centric Networking, HotICN 2018*, no. HotICN, pp. 252–253, 2019,
[23] B. W. P. Bettina Warburg and Tom Serres, *THE BASICS OF BLOCKCHAIN*.
[24] P. Xiao, *Practical Java Programming for IoT, AI, and Blockchain*. 2012.
[25] P. Xiao, *Practical Java Programming for IoT, AI, and Blockchain*. John Wiley & Sons, 2019.
[26] B. A. C. Insight, S. Wani, M. Imthiyas, H. Almohamedh, K. M. Alhamed, and S. Almotairi, "Distributed Denial of Service (DDoS) Mitigation Using Blockchain—A Comprehensive Insight," *symmetry-MDPI*, vol. 13, no. 2, pp. 1–21, 2021.
[27] V. Rakovic, J. Karamachoski, V. Atanasovski, and L. Gavrilovska, "Blockchain Paradigm and Internet of Things," *Wirel. Pers. Commun.*, vol. 106, no. 1, pp. 219–235, 2019,
[28] T. Laurence, "How Decentralization Solves Connectivity's Single Point Of Failure," *www.rtinsights.com*, 2018.
[29] V. Butterin, "Ethereum Whitepaper," *Ethereum.org*, 2013.
[30] "How Blockchain and Smart Contracts Can Impact IoT | by Smartz | Smartz Platform Blog | Medium." https://medium.com/smartz-blog/how-blockchain-and-smart-contracts-can-impact-iot-f9e77ebe02ab (accessed Jan. 05, 2023).
[31] R. M. R. Alan Dennis, Barbara Wixom, *Systems Analysis and Design*, 8th ed. John Wiley & Sons, Inc., 2021.
[32] M. Papadakis, M. Kintis, J. Zhang, Y. Jia, Y. Le Traon, and M. Harman, "Chapter Six - Mutation Testing Advances: An Analysis and Survey," vol. 112, A. M. Memon, Ed. Elsevier, 2019, pp. 275–378.
[33] M. Bures, "Framework for Integration Testing of IoT Solutions," in *2017 International Conference on Computational Science and Computational Intelligence (CSCI)*, 2017, pp. 1838–1839.
[34] N. Anwar and S. Kar, "Review Paper on Various Software Testing Techniques & Strategies," *Glob. J. Comput. Sci. Technol.*, vol. 19, no. 2, pp. 43–49, 2019,
[35] S. Siboni *et al.*, "Security Testbed for Internet-of-Things Devices," *IEEE Trans. Reliab.*, vol. 68, no. 1, pp. 23–44, 2019,
[36] M. Singh, A. Singh, and S. Kim, "Blockchain: A game changer for securing IoT data," in *2018 IEEE 4th World Forum on Internet of Things (WF-IoT)*, 2018, pp. 51–55.
[37] Y. N. Aung and T. Tantidham, "Review of Ethereum: Smart home case study," in *2017 2nd International Conference on Information Technology (INCIT)*, 2017, pp. 1–4.
[38] D. Pavithran, K. Shaalan, J. N. Al-Karaki, and A. Gawanmeh, "Towards building a blockchain framework for IoT," *Cluster Comput.*, vol. 23, no. 3, pp. 2089–2103, 2020,