JCBI
Journal of Computing & Biomedical Informatics
ISSN: 2710 - 1606
Volume 06 Issue 01
2023

Review Article
https://doi.org/10.56979/601/2023

Guardian Hive: Safeguarding IoT Devices with Honeypot Security

Danial Jamil[1], Muzammil Ahmad Khan[2], Zaib Un Nisa Khosa[1], Mubeen Ashraf[3], Fahad Atta[4], and Yousaf Haroon[5]

[1]Department of Computer Science & IT, Ghazi University, Dera Ghazi Khan, Pakistan.
[2]Computer Engineering Departments, Sir Syed University of Engineering and Technology, Karachi , Pakistan.
[3]School of Cyber Engineering, Xidian University, Xi'an China
[4]School of Computer Science and Technology, Xidian University, China
[5]Capital University of Science and Technology Islamabad, Pakistan.
*Corresponding Author: Danial Jamil. Email: danialjamil05@gmail.com

Received: March 29, 2023 **Accepted:** August 30, 2023 **Published:** December 05, 2023

Abstract: The word "Internet of Things (IoT)" denotes a modern concept that attempts to connect our current technology. The Internet evolved into the foundation for a wide range of objects uniquely linked inside the widely used Internet set-up known as the "Internet of Things." As IoT is widely used, devices are more susceptible to attacks from hackers. The IoT uses the honeypot concept to thwart these kinds of attacks. The security source of inspiration, a honeypot, is set up to entice criminals by acting as a false alarm. However, this process is limited to a unique machine, while the Honeynet is a network of Honeypots with high levels of interaction with them. As a result, we will apply the honeynet concept to the Internet of Things. This research focuses on how to defend against man-in-the-middle attacks on Internet of Things technology while enabling fraudulent assessment. This study integrates the OAuth authentication technique into the honeynet, enabling us to tackle the Internet of Things (IoT) issue, which is Man-in-the-Middle (MITM) attacks.

Keywords: IOT; Honeynet; Open Authorization; MITM cyber-attack.

1. Introduction

Internet of Things (IoT) belongs to any device linked to the Internet. It all comes down to how devices interact with one another to communicate. In addition, computers can communicate with one another, but any IP-enabled machinery, like air conditioners, automobiles, and city dumpsters, can do the same thing. IoT is only defined as "a network of data-gathering and data-exchanging Internet-connected things." The two primary elements of the motto "Internet of things" are elements and the Internet. Things are objects—items or devices—and the Internet is a basis for connection.

You will wake up in a very different world in a few years when almost all technology that people come into contact with has some form of "logic." However, allowing devices to communicate with one another comes with a cost, which is better described as severe weaknesses in security. When a device joins a network of other devices, even if it is safe on its own, it opens itself up to many threats [1].

1.1 Developments of IoT

IoT may be supposed to be an extensive system made up of many devices that are working by many different innovations, like barcodes, RFID, radio frequency connections, Near Field Communication (N and so on, in addition to computers. Its connection is made possible by wired as well as wireless technologies. (Figure 1) clearly illustrates the IoT's foundational structure starting from the internet, datacenter and data acquisition from sensors.

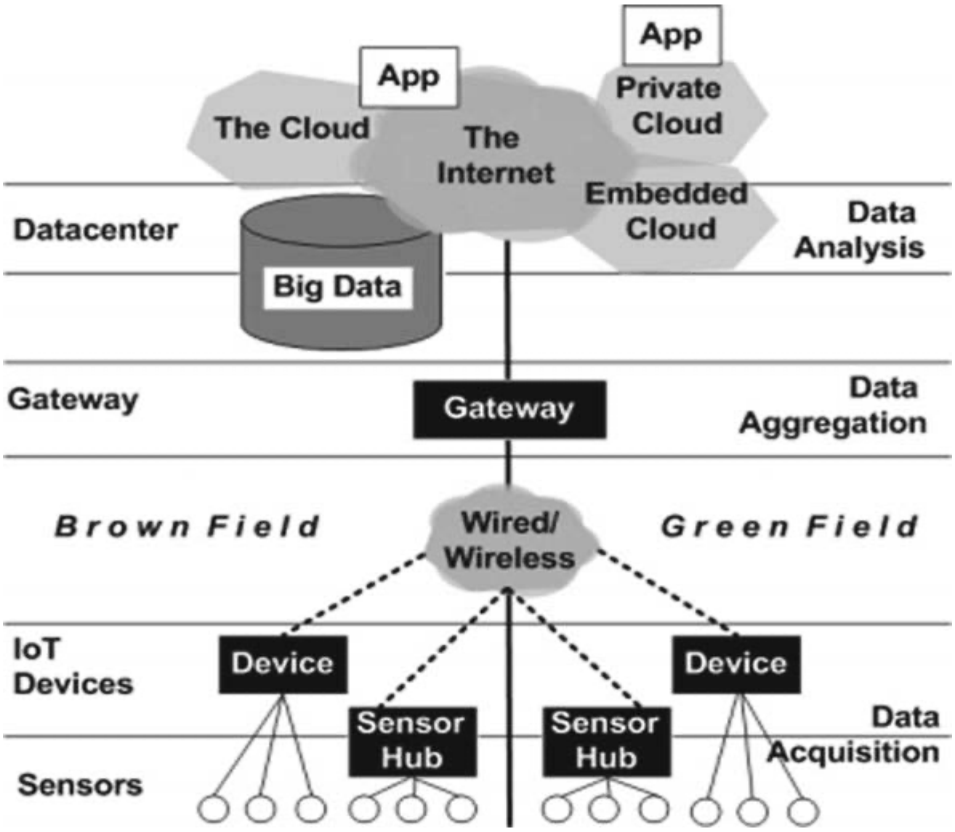


Figure 1. IoT structure

IoT devices are primarily made up of controllers and devices. Information detection is the responsibility of detectors. It must take some action, which controllers can carry out, based on the sensor data [2]. A gateway is its point of management. After gathering, data is referred to the portal. The online platform is connected to the router. Edge computing is done first, and then the acquired data is directed to the cloud. Edge computing is the preprocessing of collected data before it enters the cloud. Remote control and monitoring of data is possible through programs that an agent controls. Whereas "brownfield" refers to connecting pre-existing infrastructures, systems, and devices, "greenfield" is directed at developing new goods entirely from the ground up [1].

Some of the most recent innovations aimed at boosting the safety of the Internet of Things (IoT) have dealt with methods for controlling access to and data stored by smart devices; fostering an environment where users and devices can trust one another; and enforcing existing privacy and security regulations. However, despite these precautions, the (IoT) field is still susceptible to a plethora of assaults designed to interrupt networks [3]. Because of this, it's important to have an additional layer of protection to identify and counterattack attacks. There are numerous methods to avoid such attacks, but an active strategy that can turn the tables on the attacker is essential. Honeypot and Honeynet are the greatest tools for the job [1].

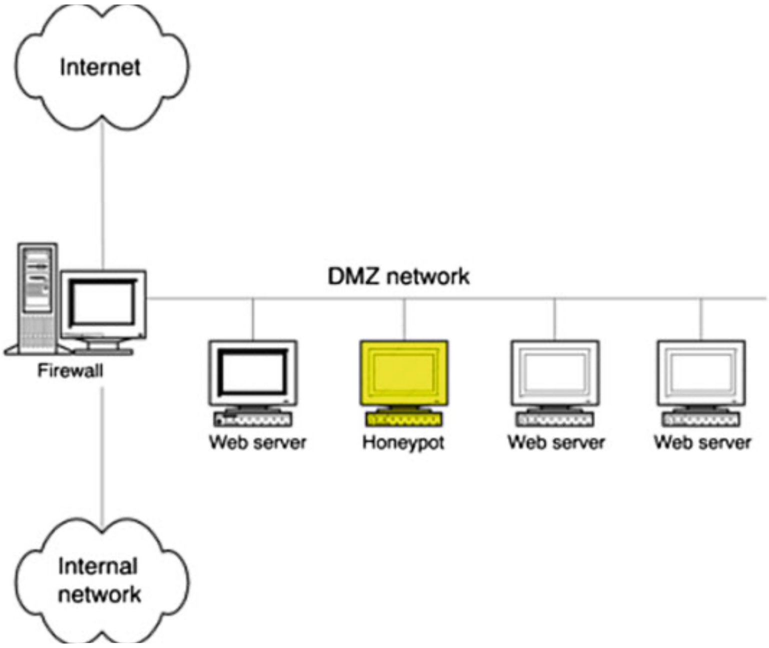


Figure 2. Low level honeypot

1.2 Honeynet

Honeypots are security resources framed up as tricks to be examined, criticized, or hacked [4]. It includes no information that is confidential but aims to be a valuable network component. Production honeypots safeguard an organization, whereas research honeypots learn. Honeynets are networks of Honeypots as shown in Fig. 2. Honeynet is a smart, interactive honeypot. Data capture, collection, and control are essential. The data control function controls data flow, ensuring attackers are unaware of monitoring and allowing them to act freely. Securely gathering data from nodes is data collection. Data capture grabs Honeynet data. Honeynet collects little yet valuable information as shown in Figure. 3 [1] for Honeynet architecture.

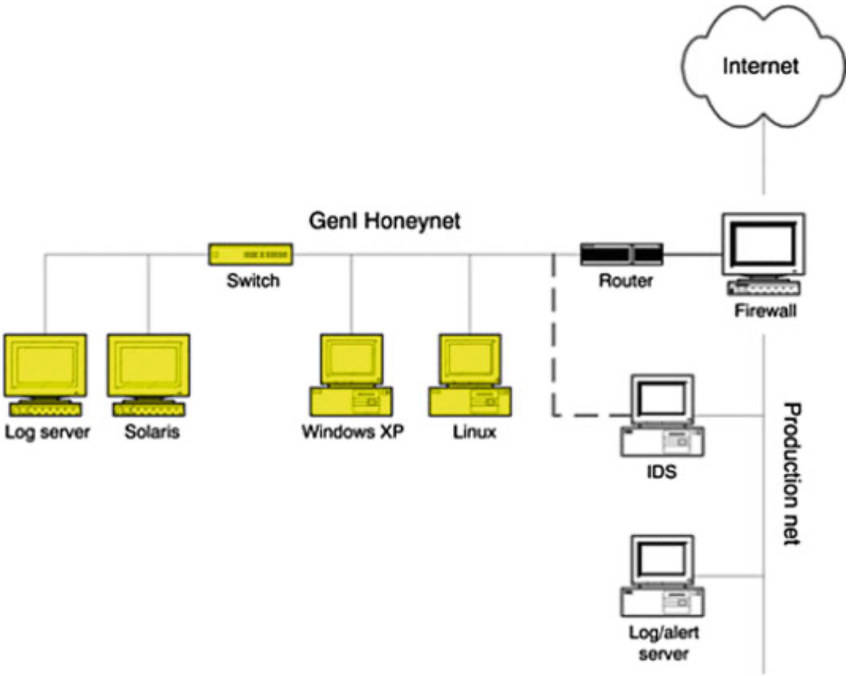


Figure 3. Honeypot structural view

2. Problem Identification

With the rapid growth of IoT, the threat of MITM attacks in IoT is becoming an important concern. Man-in-the-middle (MITM) strikes frequently go undetected and unreported in statistical data. Conventional security techniques, such as traffic blocking, prevent attackers from gaining network access, but they also hinder the investigation of attacks. We require a robust mechanism to effectively counteract Man-in-the-Middle (MITM) attacks in IoT networks while retaining the ability to evaluate them.

While we're on the topic of authentication, common methods used by Honeypot and Honeynet systems compare incoming requests to known patterns of viruses or worms. When hackers can easily eavesdrop on network traffic and steal passwords, traditional authentication mechanisms become ineffective. It is critical to utilize robust authentication methods that keep passwords secret. To determine if the traffic is harmful or not, traditional honeynet methods employ intrusion detection systems. On the other hand, intrusion detection systems have a drawback: they begin to produce false positives when the traffic load increases. As a result, Honeynet will squander its resources analyzing traffic that was never harmful, leading to incorrect findings.

3. Implemented Framework

The main concept of this architecture is to reduce the risk of Man-in-the-Middle (MITM) strikes in the Internet of Things (IoT) network via a Honeynet. However, it is important to note that prevention in this context does not entail blocking communications. To develop a Honeynet, we must possess the capability to recognize and examine intrusions for subsequent utilization. This system offers both deception and authentication mechanisms to circumvent Man-in-the-Middle (MITM) attacks as seeen in Fig. 4, where the structure of an IoT Honeynet that is supported by an authentication mechanism.

The user initiates a message request to the Internet of Things gadget in order to remotely execute specific events by signing in. The request will undergo confirmation in the system. If the request is new, its details are not saved in the electronic signature system, and it will be directly sent to the authentication system. The Authenticator interface sends a message to a valid user, using the user's mail ID and MAC address stored in the database, together with the sender's message request. We are employing OAuth2 for authentication and permission. The Python-OAuth2 package is utilized for the implementation of OAuth. The request will only be implemented if the user allows it, and it will be sent to the actual IoT devices. If the request message has been tampered with by an attacker, the user will be able to detect it through this authentication technique and then reject the request. If the user rejects the request, the traffic will be forwarded to the Honeynet to determine whether the deleted request is indeed malicious or not. Occasionally, users may unintentionally dismiss the request, therefore it should not be seen as malevolent. Subsequently, the request will be routed to the Intrusion Detection System (IDS) to verify the port and protocol. If the traffic is malicious, the specifics of the harmful activity and its signature will be stored in the Signature Verification Database. In the event of a recurring attack, the system will be capable of directly redirecting requests to the Honeynet.

If a request matches the Signature Validation Database's records, the request will not be forwarded to the authentication system but rather sent directly to the Honeynet for further examination. The user will receive a notification stating that "your original request is altered." This will lead to a decrease in processing time. With this framework, we can protect IoT environments against man-in-the-middle assaults while still analyzing them with the help of deception. With the framework in place, false-positive production is less of an issue, and Honeynet can no longer produce misleading results as in Figure 5.

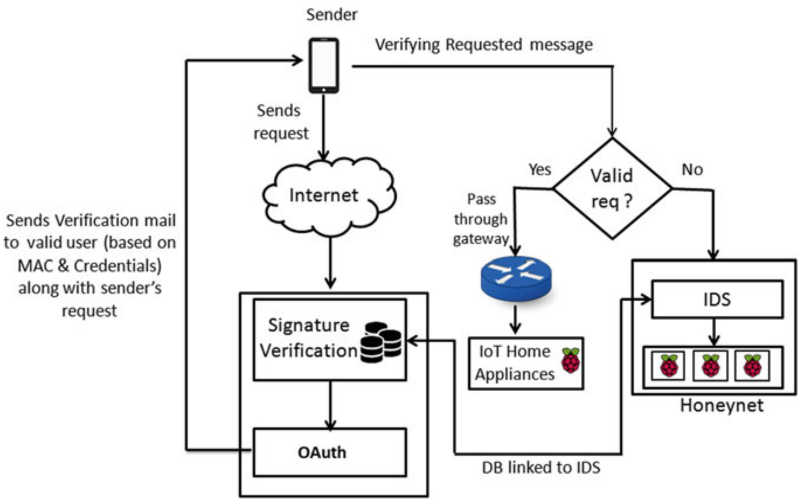


Figure 4. Implemented structure view

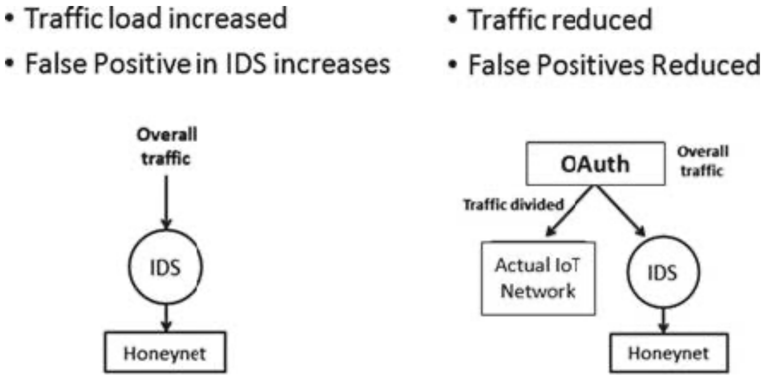


Figure 5. Controlled network traffic by applying OAuth

3.1 System Implementation Process Diagram

The below figure 6 shows the flow fiagram of the implemented system used in this study.

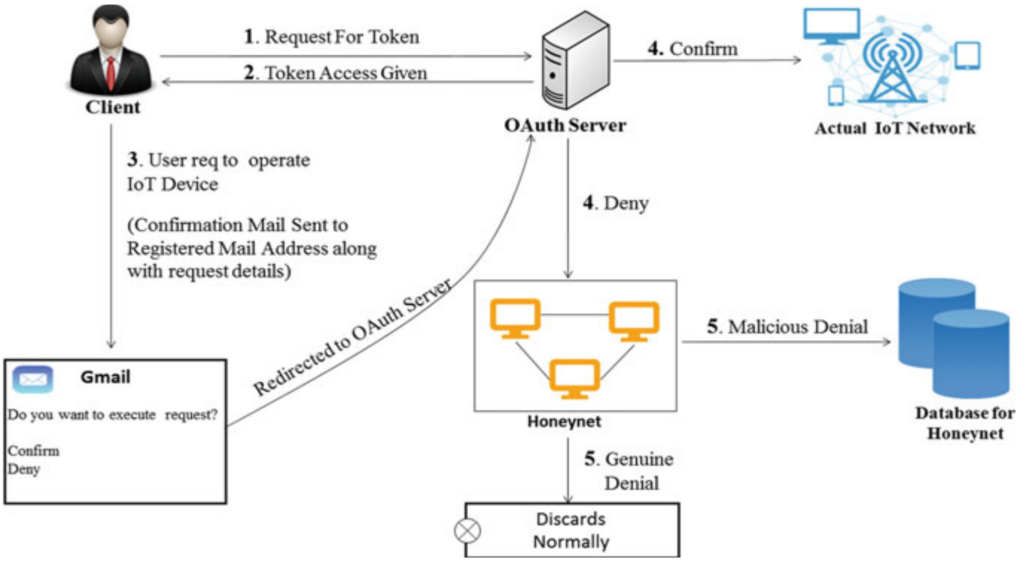


Figure 6. Flow diagram of implemented system

3.2 Virtual Honeynet Architecture

The below figure 7, shows the structural view of the virtual honeypot system.

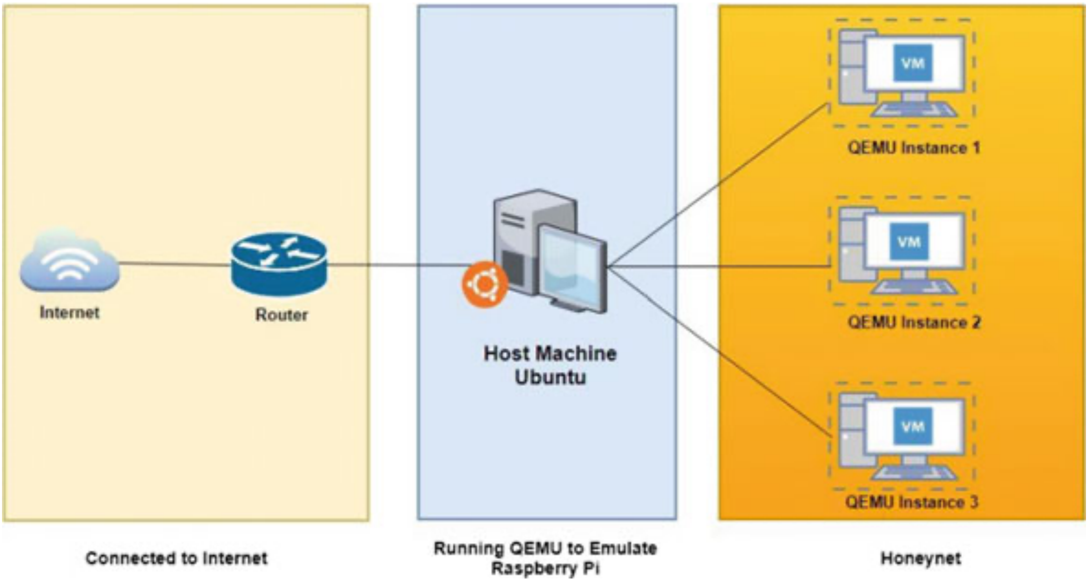


Figure 7. Structural view of virtual honeypot

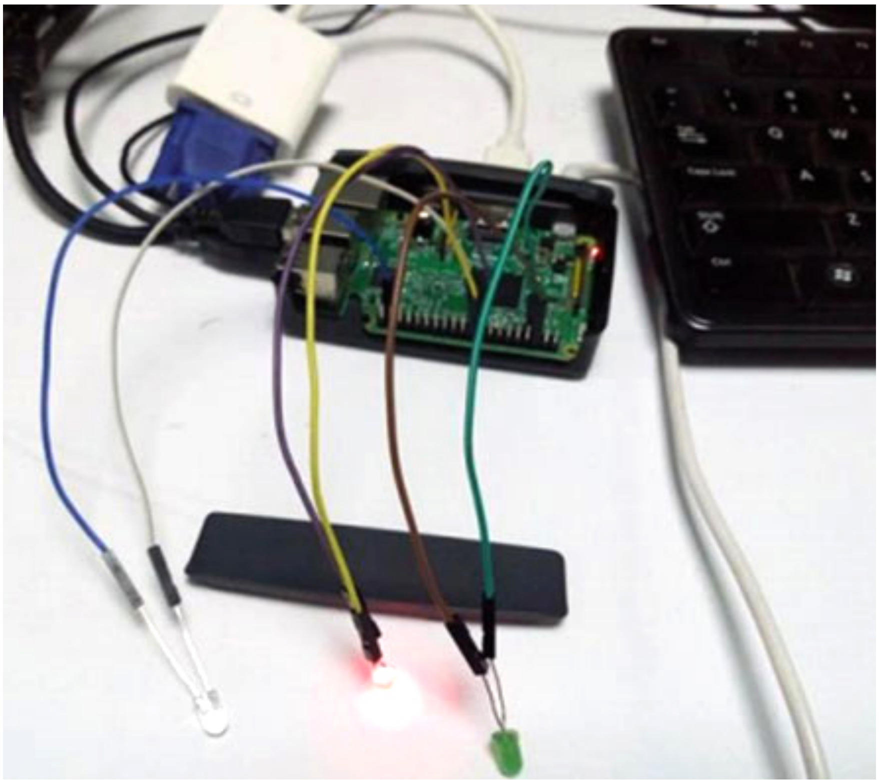
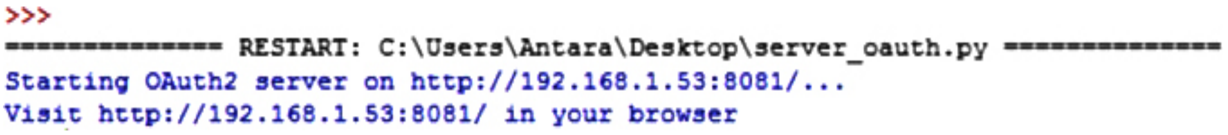


Figure 8. IOT network devices

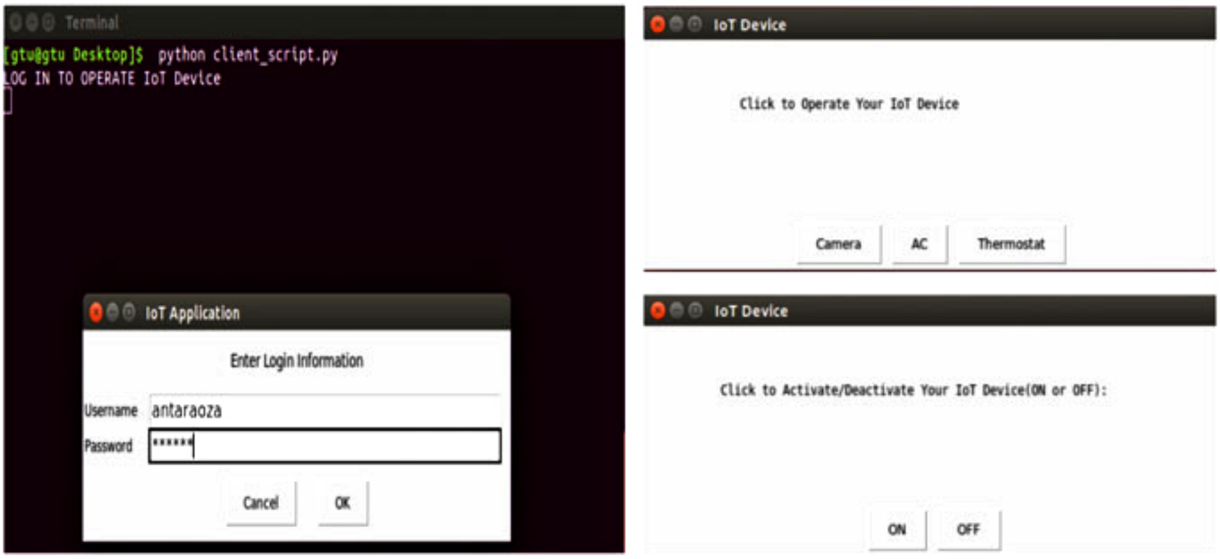
4. Honeynet and OAuth Installation

Begin the main program process by executing the programming code which is written in Python. The Linux server has been configured to receive requests.



```
>>>
============== RESTART: C:\Users\Antara\Desktop\server_oauth.py ==============
Starting OAuth2 server on http://192.168.1.53:8081/...
Visit http://192.168.1.53:8081/ in your browser
```

Launch the client function to manage the IoT device from a different machine as shown below:



```
[gtu@gtu Desktop]$ python client_script.py
LOG IN TO OPERATE IoT Device
```

IoT Application

Enter Login Information

Username: antaraoza

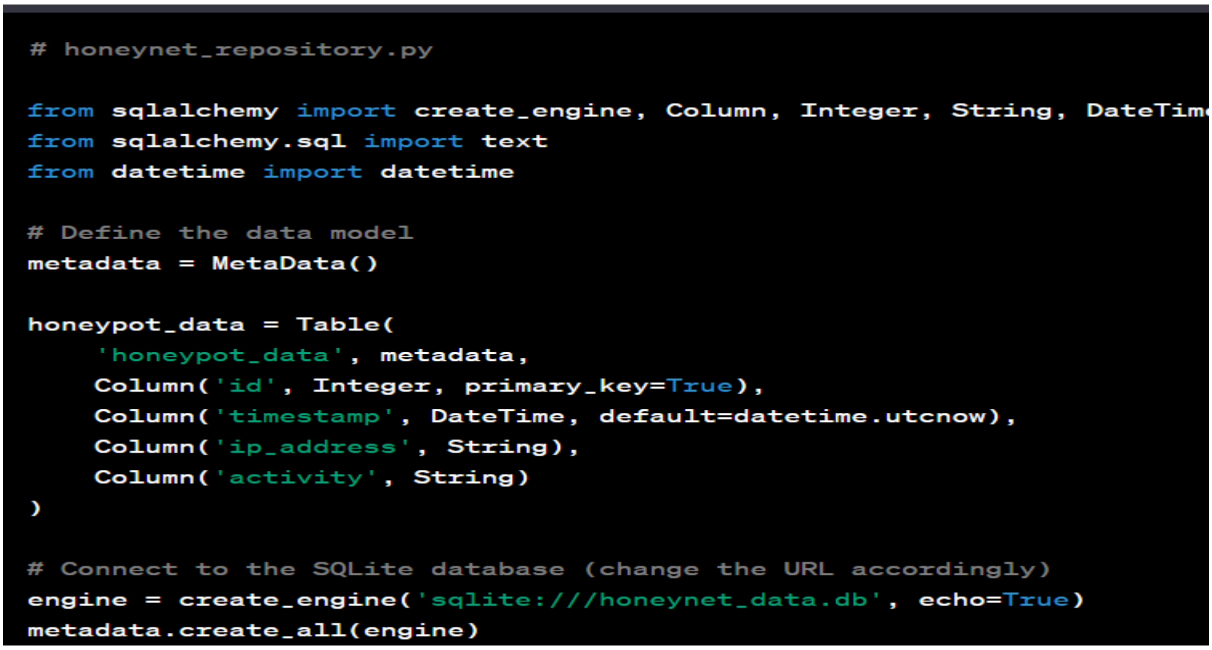
Password: ******

Terminal
Cancel
OK
IoT Device
Click to Operate Your IoT Device
Camera
AC
Thermostat
IoT Device
Click to Activate/Deactivate Your IoT Device(ON or OFF):
ON
OFF

Figure 9. Function Manager

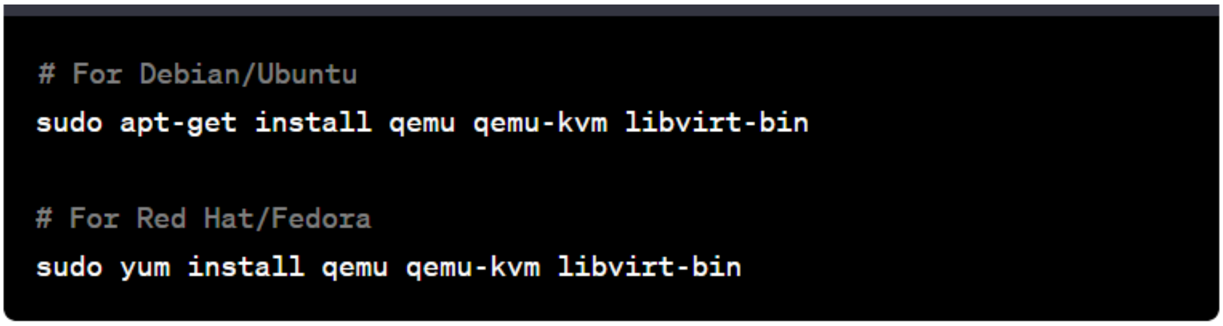
Only authorized users are allowed to get the verification email. "Access control is paramount in ensuring the security of user accounts. Our system reinforces this principle by allowing only authorized users to receive verification emails. This added layer of authentication ensures that sensitive information is delivered exclusively to intended recipients, thwarting unauthorized access. By implementing stringent verification processes, we enhance account security and protect user data from potential breaches. This feature not only bolsters privacy but also contributes to a more robust and reliable user authentication system. With a focus on user trust and data integrity, our platform prioritizes the safeguarding of sensitive communications, reinforcing the foundation of a secure and user-centric experience.

Honeynet stores the information obtained in a central repository for future usage in examinations.



```python
# honeynet_repository.py

from sqlalchemy import create_engine, Column, Integer, String, DateTim
from sqlalchemy.sql import text
from datetime import datetime

# Define the data model
metadata = MetaData()

honeypot_data = Table(
    'honeypot_data', metadata,
    Column('id', Integer, primary_key=True),
    Column('timestamp', DateTime, default=datetime.utcnow),
    Column('ip_address', String),
    Column('activity', String)
)

# Connect to the SQLite database (change the URL accordingly)
engine = create_engine('sqlite:///honeynet_data.db', echo=True)
metadata.create_all(engine)
```

Figure 10a. Honeynet instances running on QEMU



```
# For Debian/Ubuntu
sudo apt-get install qemu qemu-kvm libvirt-bin

# For Red Hat/Fedora
sudo yum install qemu qemu-kvm libvirt-bin
```

Figure 10b. Honeynet instances running on QEMU

The reality of the IoT network, in this case, Three LEDs for the AC, for the photographic camera, AC, and regulator or also Raspberry PI is used to construct simulated Honeypot scenarios. Figure 9 depicts three Honeynet occurrences. QEMU is used to imitate Raspberry Pi in this case. Because we require numerous Raspberry Pi to build the system, virtualization is the best and most cost-effective method in this scenario. Since no additional administrative tools are required by QEMU, it is advantageous that it is compact as shown in Figure. 10a and 10b.

5. Analyzing the Outcomes' Parameters

The ouctomes parameters analyzing shows in the network congestion in the figure 10 below that give the clear representation of normal system and improved system process.

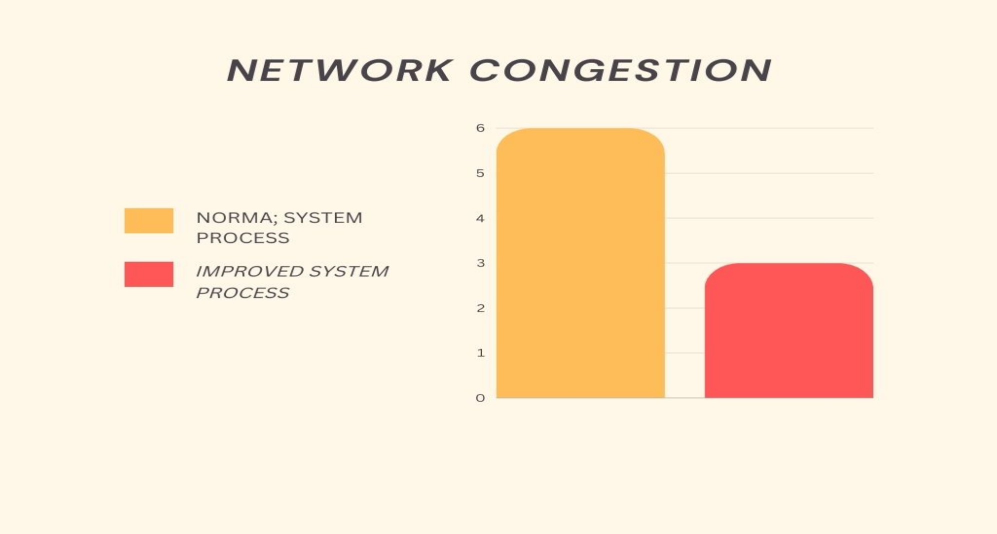


Figure 11. Measurement of contention in networks

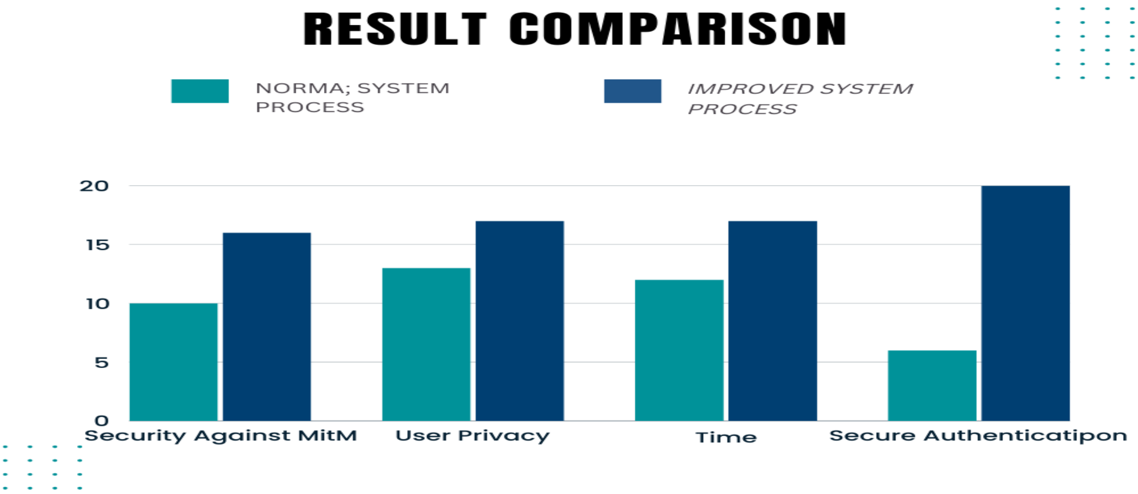


Figure 12. Ratio of variables

6. Conclusion

In recognizing the inherent imperfections of our world, one must acknowledge that perfection is not the goal. Rather, our objective is to adeptly assemble the fragmented components using every available resource at our disposal. This truth holds particularly poignant in the realm of cyber defense. Securing the Internet of Things (IoT) is akin to navigating treacherous waters, a perilous journey amidst predatory sharks. The conflict is perpetual, a constant dance between defenders fortifying their positions and assailants evolving their tactics in response. Yet, in the face of this unyielding challenge, we persist. The primary aim of this study was to address the pressing issue of man-in-the-middle attacks on IoT devices while allowing for their thorough examination. While conventional security measures, such as blocking, effectively deter infiltrations, they simultaneously impede the crucial analysis of these attacks. A novel and effective resolution lies in redirecting detrimental traffic to a Honeynet for comprehensive investigation. This framework seamlessly incorporates the OAuth (Open Authorization) method, a robust system for user authentication written in Python scripts. By integrating OAuth into IoT, not only is the authentication process fortified, thereby enhancing overall security, but it also alleviates network congestion, resulting in a reduction of false positives. Consequently, this augmentation substantially bolsters the accuracy of Honeynet analysis. One of our future research objectives is to delve into other vulnerable areas susceptible to IoT attacks. The quest for cybersecurity is unending, an evolving narrative in the digital age. In the imperfection of our cyber landscape, our pursuit remains steadfast. Our commitment is unwavering, navigating the uncharted waters of cybersecurity, adapting, and persisting in our mission to fortify the ever-expanding realm of the Internet of Things.

References

1. Oza AD, M.E. Student, Gujarat Technological University, Kumar GN, Project Engineer, CDAC-ACTS INDIA, Khorajiya M, Project Engineer, CDAC-ACTS INDIA, Survey of Snaring Cyber Attacks on IoT devices with Honeypots and Honeynet. In: 3rd International conference for convergence in technology (I2CT), IEEE, (2018).
2. Drrajivdesaimd.com, INTERNET OF THINGS (IoT)—Dr Rajiv Desai, (2017). [online] Avail-able at: http://drrajivdesaimd.com/2016/07/19/internet-of-things-iot
3. Sherasiya T, Upadhyay H, Intrusion detection system for internet of things. IJARIIE-ISSN(O)-2395-4396, (2016).
4. Dial.uclouvain.be, Measurements of compromised IoT devices from blackhole and hon-eypot|Mémoire UCL, (2017). [online] Available at: https://dial.uclouvain.be/memoire/ucl/en/object/thesis%3A10671
5. Honeynet.org, Know your enemy: Honeynets|The Honeynet project, (2017). [online] Available at: https://www.honeynet.org/papers/honeynet
6. Anas Abd Almonim Nour Albashir, Detecting unknown vulnerabilities using Honeynet, IEEE, (2015).
7. La QD, Member, IEEE, Quek TQS, Senior Member, IEEE, Lee J, Member, IEEE, Jin S, Mem-ber, IEEE, Zhu H, Deceptive attack and defense game in Honeypot-enabled networks for the internet of things, IEEE, (2015).
8. Zhang W, He H, Kim T, Xen-based virtual honeypot system for smart device. Springer Science+Business Media, New York, (2013).
9. IEEE Spectrum: Technology, Engineering, and Science News, Internet-exposed energy control systems abound, (2017). [online] Available at: https://spectrum.ieee.org/energywise/energy/thesmartergrid/thousands-of-control-systems-connected-to-the-internet
10. Anirudh M, Thileeban SA, Nallathambi DJ, Use of Honeypots for mitigating DoS attacks targeted on IoT networks. In: IEEE international conference on computer, communication, and signal processing, IEEE, (2017).
11. Sans.org, (2017). [online] Available at: https://www.sans.org/readingroom/whitepapers/detection/designing-implementing-honeypotscada-network-35252
12. Miloslavskaya N, Tolstoy A, Ensuring information security for internet of things. In: 2017 IEEE 5th international conference on future internet of things and cloud, IEEE, (2017).
13. Sans.org. Cite a Website - Cite This For Me, (2017). [online] Available at: https://www.sans.org/reading-room/whitepapers/detection/designing-implementing-honeypot-scada-network-35252 [Accessed 30 Nov. 2017]
14. Natalia Miloslavskaya , Alexander Tolstoy "Ensuring Information Security for Internet of Things" 2017 IEEE 5th International Conference on Future Internet of Things and Cloud,IEEE, (2017).
15. Dial.uclouvain.be. Measurements of compromised IoT devices from blackhole and honeypot Mémoire UCL, (2017). [online] Available at: https://dial.uclouvain.be/memoire/ucl/en/object/thesis%3A10671
16. I. Andrea, C. Chrysostomou and G. Hadjichristofi, "Internet of Things: Security vulnerabilities and challenges" 2015 IEEE Symposium on Computers and Communication (ISCC), Larnaca, (2015), pp.180-187.
17. Tariqahmad Sherasiya, Hardik Upadhyay "Intrusion Detection System for Internet of Things", IJARIIE-ISSN(O)-2395-4396, (2016).
18. Jain, Pragya and Sardana, Anjali, "Defending against Internet Worms Using Honeyfarm", Proceedings of the CUBE International Information Technology Conference, (2012), pp.795-800.
19. Fabien Pouget, Marc Dacier ,Hervé Debar "Honeypot, Honeynet, Honeytoken: Terminological issues",Research Report RR-03-081, (2003).
20. Owasp.org. Top 10 IoT Vulnerabilities (2014) – OWASP, (2017). [online] Available at: https://www.owasp.org/index.php/Top_10_IoT_Vulnerabilities_(2014).
21. Getting-started-with-iot-security-mapping-the-attack-surface http://resources.infosecinstitute.com/getting-started-with-iot-security-mapping-the-attack-surface/

22. Wallen, J. Five nightmarish attacks that show the risks of IoT security | ZDNet, (2017). [online] ZDNet. Available at: http://www.zdnet.com/article/5-nightmarish-attacks-that-show-the-risks-of-iot-security/ [Accessed 30 Nov. 2017].

23. Drrajivdesaimd.com. INTERNET OF THINGS (IoT) – Dr Rajiv Desai, (2017). [online] Available at: http://drrajivdesaimd.com/2016.