A Secure Network Switch for IoT Devices using Whitelist Approach

Porapat Ongkanchana

03-170425

A thesis presented for the Bachelor Degree

Department of Information and Communication Engineering
The University of Tokyo, Japan
January 20, 2019

Abstract

Abstract Abstract

Contents

List of Figures

Chapter 1

Introduction

1.1 The Internet of Thing

IoT, the internet of thing, is a concept of implementing commutable daily objects. These objects are connected to the internet, allowing the transmission of gathered information and controller commands to be done anytime, anywhere. The problems that used to take time and effort to detecting or solving can now be done easier and faster because of IoT.

IoT technology is developing and growing at the exponential rate. The global economic impact of IoT is estimated to reach trillion dollars by 2025 and more than 50 billion devices are expected to be deployed in 2020. [1] Many of world leading technology companies are pouring their resource into developing the future of IoT. [2] It is natural to think that IoT will soon be integrated into a part of our life.

IoT applications can be used in almost every aspects of our life. For example; in healthcare system, some of patient's medical asset information can be collected through a smart wearable, reducing tasks of medical staff. In electricity power system, "Smart grid" implementation allows a better energy management and more durable against blackout. "Smart house", where your house can automatically turn off the light when nobody is at home. This is just a tip of an iceberg of what IoT can do for us.

With the number of IoT device and its application growing, our world has never been more comfortable, however it has also never been scarier. IoT device has access to our privacy information, if these information falls into the wrong hand, it can lead to privacy breaching, data forging or even worse a matter of life and death. The security of IoT is something we really need to put our attention to.

[http://forms1.ieee.org/rs/682-UPB-550/images/IEEE-IOT-White-Paper.pdf]

1.2 IoT Security

In 2018, it is reported that "Telnet attack" is the most frequent attack on IoT system, followed by "SSH attack". The telnet attack and SSH attack is an attack vector that exploit user's behavior of not changing device's default username and password. Attackers try to get device's root permission by brute-forcing all user, password combination. If adversary has acquired device's root permission, he can use that IoT device as their desire. This means accessing to all data in the device, installing and uninstalling any software, downloading and uploading, attacking target servers or even spreading the attack to other devices in LAN. It may seem like this kind of attack can be prevented by user's configuration, but the study show that more than 300 million devices are venerable by this attacking method. Moreover, in 2016, the infamous “MIRAI” virus was spread over six hundred thousand devices and caused one of the biggest DDOS attack ever in the history of mankind.

[https://securelist.com/new-trends-in-the-world-of-iot-threats/87991/]

1.3 purpose

In this research, we prioritize on improving the foundation of IoT system, the “Smart Sensor”. Smart sensor is used in almost every IoT system. We want to create a system, that can help user securing their devices and alleviating troublesome task of device’s configuration.

1.4 Paper Structure

In chapter 2, we described technologies related to this research along with previous attempt to secure IoT system. We presented our IoT security system in chapter 3. In chapter 4, we conducted various experiment to investigate system’s functions performance, along with how to improve it. Chapter 5 is this thesis summary and discussion.

Chapter 2

Related Technology

2.1 IoT system Architecture

There are many framework architectures used in IoT system. The simplest and most well-known model is the three-layer model: perception layer, network layer, application layer.

1. **Perception Layer** or recognition layer: This layer's main responsibility is to collect useful data from environment, translate those analog data into digital and send it to the server.

2. **Network Layer** : This layer in charge of transmitting the collected data to the application layer. Data can be sent through conventional LAN cable, 3G/4G, Zigbee etc.

3. **Application Layer** : This layer is where collected data is used to create services. The range of service is impressive: authenticate, real time data visualization and more.

2.2 Smart Sensor

Smart sensor is a device locate at the perception layer of IoT system, normally used for collecting environment data. Next, described some of sensor's characteristics.

1. **Sensor is deployed into the physical environment;** For example, electricity consumption meter placed behind the wall, near the electric plug to tracking how much electricity spent. Temperature meter embedded into the wall. Tsunami detection sensor located along sea course.

2. **Expected to operate without user's recognition;** After the initial setup, devices are often deployed sparsely into outdoor environment far away commanding center.

3. **Device's life span depends on the battery power;**

4. **Low computational power;** In order to reduce the manufacture cost, sensor usually can operate with a very limited set of functions: sensing and transmitting. All high-computation relied operations is calculated at the server.

5. **Communicate with a limited number of hosts;** After device sensed the environment and recorded all necessary data, it transmitted gathered data to its server. Normally this is the only function, sensor serves. So, the main server device needs to communicate to is its application server and there is no need for device to communicate with others.

2.3 Whitelist and Blacklist

Whitelisting and Blacklisting are network filtering techniques. Blacklist is a list containing IP or domain name of malicious hosts. When filter using blacklist-technique is applied to network traffic stream, all the packages related to suspicious hosts in blacklist is discarded. In the other hand, whitelist is a list of secured hosts. Only hosts in the whitelist can communicate with our network. Blacklist is used in various application, while whitelist is an un-popular technique. Our daily device, PC or smartphone, doesn't have any communication pattern. Listing all the host we would communicate to is impractical.

2.4 Attempt in securing IoT devices

Chapter 3

Proposed System

3.1 System requirement

This research aims to improve the security of smart sensor in IoT system. The main requirements of the system are mentioned below.

1. **Guarantee device safety;** The focus of the system is to secure the perception layer of IoT system, the sensor. System is implemented believed that security in network layer and application layer is integrity.

2. **Independent from device;** System should be able to use without any additional setups on IoT devices. This is designed to prevent adding an additional work when dealing with a large-scaled IoT system.

3. **Can implement on top of an existed system;** Users can easily install system without having to change their network topology.

4. **Do not affect system performance;** After system is added to the workspace, it should not impede on the performance: system latency, packet drop.

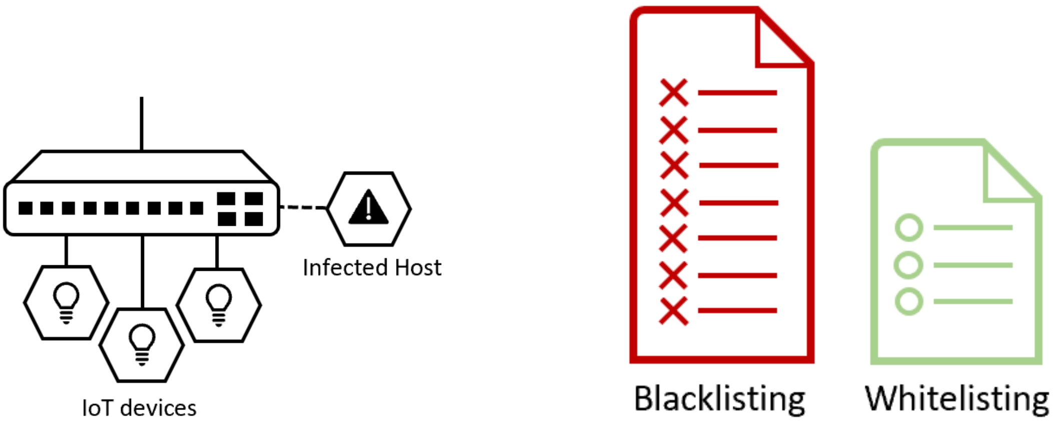
We want to create the system with all characteristics mention above. The approach we came up consists of two main idea: Switch level security and Whitelist.

3.1.1 Switch level security

Switch level security is securing IoT at the lower level of network protocol. The security measurement is done outside of the edge device, between the internet and device. (figure 2) The benefit of using this method is that there is no need for additional configuration at edge devices. Moreover, it is security at low level where transmission protocol is more restricted, therefore can be managed more easily. Another benefit of layer 2 security is even when malicious host or infected host has entered the local area network, it can prevent the infection from spreading.

3.1.2 Whitelist

Consider that the number of hosts, sensor normally connect to is limited: NTP (Network Time Protocol) server for clock synchronization, DNS (Domain Name System) Server for translating between host name and IP address, DHCP (Dynamic Host Configuration



(a) LAN Level Security

(b) Whitelist

Figure 3.1: System's key characteristics

Protocol) server for dynamic IP assigning, and HTTP (Hypertext Transfer Protocol) server for its application usage. Whitelist approaching is considerably more efficient in IoT sensor.

3.2 System Architecture

3.2.1 System Stage

System operation is divided into 3 stages: Preparation stage, Analyzation stage and Operation stage. Preparation stage and Analyzation stage would be performed at the system's initialization, while Operation stage is used after that for the rest of system operating period.

3.2.2 Preparation Stage

In this stage, all packet going through switch would be captured. This system is developed under the assumption that during Preparation Stage, the network, edge devices, servers, middleware is secure and no malicious software has entered into the system yet.

3.2.3 Analyzation Stage

In Analyzation stage, data collected by network plane is then further investigated to find.

1. All devices in the LAN, **Host discovery**.
2. All hosts, each device needs to operate, **Whitelist Extraction**.

Host Discovery

We believe that it is possible to find all devices in LAN by analyzing captured ARP (Address Resolution Protocol) packets.

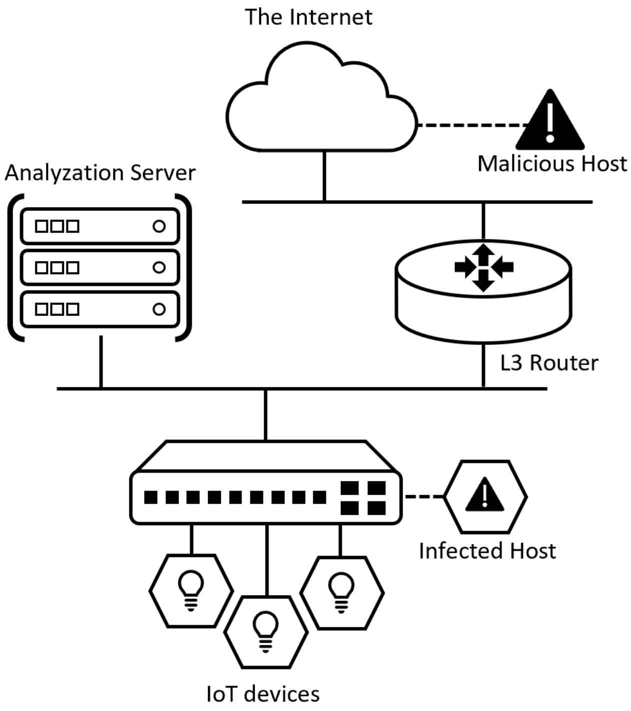


Figure 3.2: System Architecture

Whitelist Extraction

Whitelist Extraction program use packet data to create whitelist profile for each discovered host. In this research, we use rule-based algorithm to extract whitelist.

Definition of Secured host and Whitelist Rule In this research, we define secure host as "host that sensor needs in order to fully operate" and we assume that secure host can be extracted using the following rule.

> *"Only IP addresses to which the device initiates connection is considered secured host"*

3.2.4 Operation Stage

After Analyzation Stage, switch would start filtering network traffic, letting device to only communicate with hosts in its whitelist, guarantee the integrity of IoT system.

3.2.5 System Operation Flow

First, we start capture devices' packet under assumption that there is no on-going attacks or infected host in the LAN. After packet data is sufficiently captured, we pass gathered packet to analyzation server, where packet is inspected to find devices in the system. Then we create a whitelist for each device and start filtering traffic according to it.

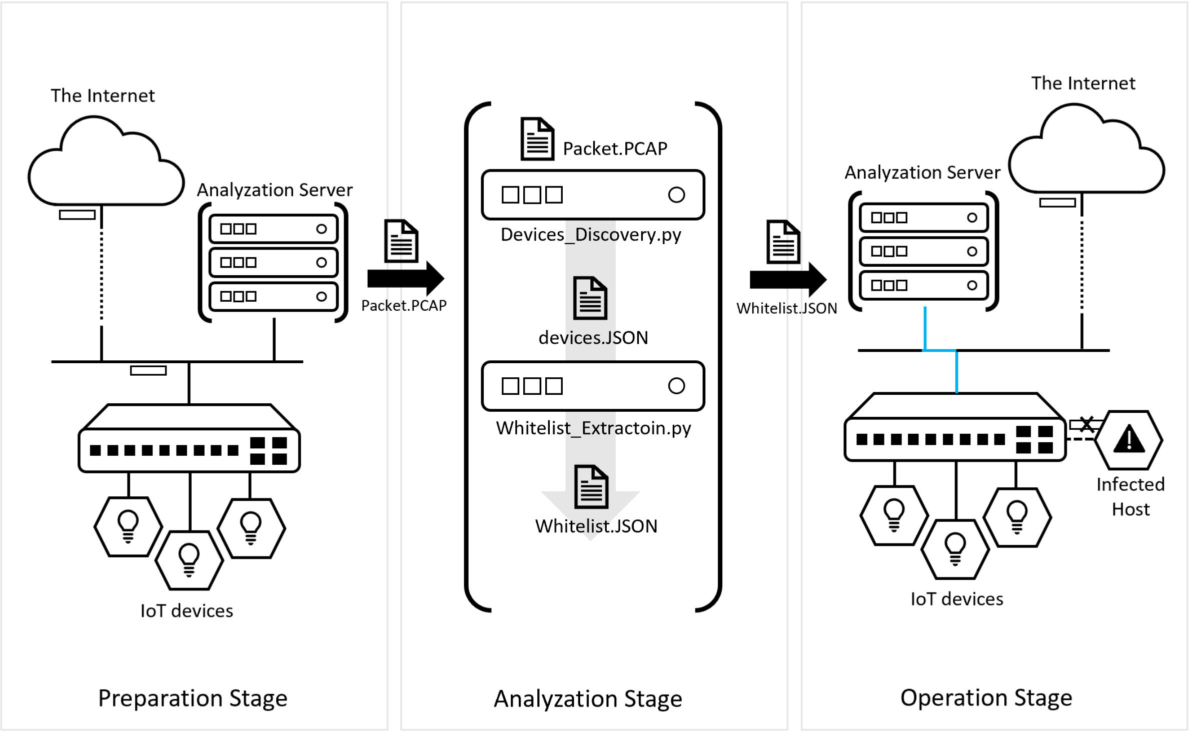


Figure 3.3: System Overview

Chapter 4

Experiment

Various experiments were conducted to find the way to implement functions mentioned in section 3: device discovery, whitelist extraction, network filtering. Next, we would like to explain IoT system used as the test subject in this experiment, also the elaborate detail for each experiment.

4.1 IoT system

4.1.1 eroom

4.1.2 10F

4.2 Packet Capturing

4.2.1 Purpose

Packet data is captured, then analyzed to investigate the pattern of IoT system communication. This packet data capturing method is used in both "Device discovery experiment", and "Whitelist extraction"

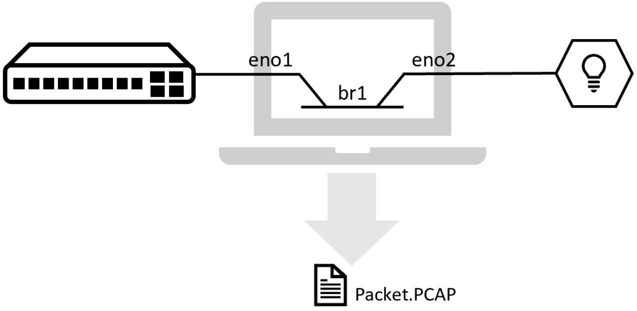


Figure 4.1: Packet Capturing

4.2.2 Method

First, we inserted a computer in between Switch and any IoT devices, we aim to capture its packet. As a requirement, computer used in this procedure must has at least 2 network interfaces. One network interface was connected to the switch, while another was connected to IoT device (Figure 4.3). After that, we created network bridge to connect two of inserted computer network interfaces. Any computers run with Linux kernel can initiate a bridge using "brctl" command. Then packet would be captured using tpcdump command. (tcpdump is a TCP/IP packet sniffing command) Collected data was saved into PCAP format.

4.2.3 Result

Packet Description

4.3 Device Discovery

4.3.1 Purpose

Device discovery is a process conducted to find all devices in LAN. Their IP address and MAC address is necessary when performing whitelist extraction.

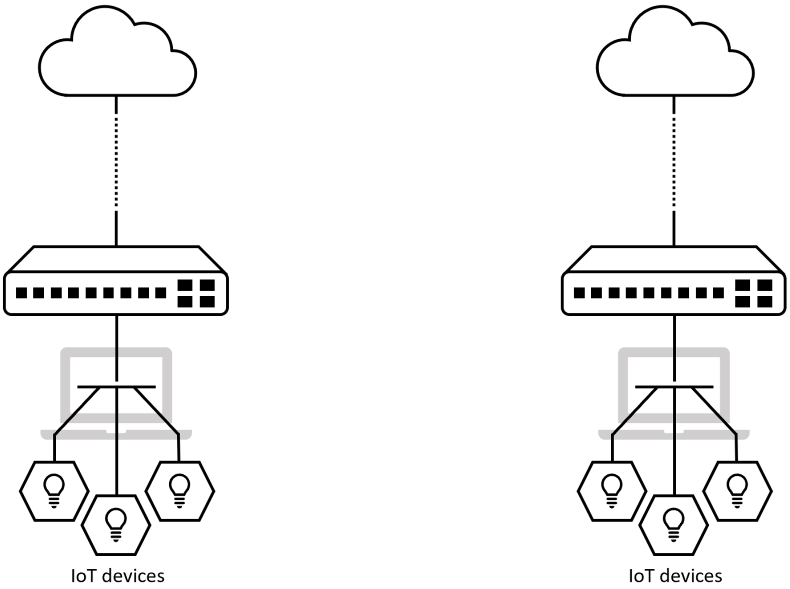
4.3.2 Method

In this experiment, we had tested capturing devices' packet in three patterns: pattern A and B. In pattern A (Figure 4.2a), we created a computer bridge between all devices and switch, while in pattern B (Figure 4.2b), we connected PC to one of IoT devices switch's ports, and capturing packet coming through. The gathered packet is then analyzed to find list of hosts in LAN.

In this experiment, we have tested host detection algorithm constructed with three approaches: Detecting hosts by ARP request, by ARP reply and by gratuitous ARP.

- **ARP request** is Arp packet with opcode equal to 1 and Target MAC address equal to "00:00:00:00:00:00". We considered Sender of this packet to be host in LAN and added tuple of its MAC address (SHA) and IP address (SPA) to host list.
- **ARP reply** is Arp packet with opcode equal to 2. We added both Target and Sender of this packet to host list.
- **Gratuitous ARP** is Arp packet which has is sender IP address equal to target IP address, its opcode equal to 1, and has target MAC address equal to "00:00:00:00:00:00". We added Sender of this packet it to host list.

We tested our approaches in the following combination.



(a) Pattern A : Capturing packet from bridge PC between all devices and switch

(b) Pattern B : Capturing packet from PC connected to one of switch port

Figure 4.2: Packet Capturing Patterns

	ARP request	ARP reply	Gratuitous ARP
Pattern A	○	-	○
Pattern B	-	○	○

Table 4.1: Tested Pattern

4.3.3 Result

4.3.4 Discussion

4.4 Whitelist Extraction

4.4.1 Purpose

In order to secure our IoT system, the suitable whitelist for each device is crucial. By implementing whitelist, not only can we guarantee that system can withstand the attacks from outside, but we can also prevent our infected device from attacking other servers. In this experiment, we want to find an algorithm to extract the secure hosts.

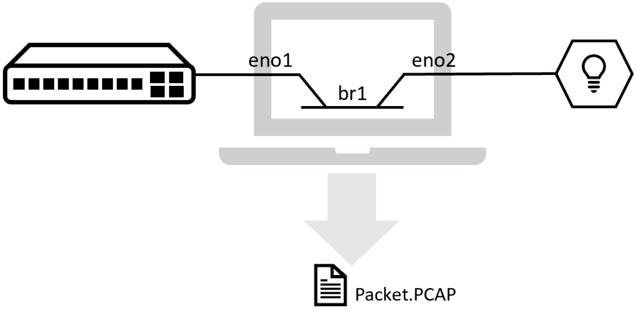


Figure 4.3: Packet Capturing

4.4.2 Method

Program and Algorithm We wrote program to extract whitelist for each device extracted in Device Discovery, using collected traffic packet as an input. Next, we would like to explain how program works.

1. Program take network traffic data as an input.
2. Traffic data was divided into a smaller group of time τ. Only IP packet with TCP/UDP as its transmission protocol was considered.
3. In each group, we looked at each host's first interaction (packet) with edge device.
    - If packet is originated from device, we decide that this host might be secure.
    - If packet is destined to device, we decide that this host might be insecure.
4. If in most group (90%), host that communicated with device was identified as "might be secure", we add it to device's whitelist.
5. The output of program is devices' whitelist saving in JSON format.

4.4.3 Result

4.4.4 Discussion

4.5 Packet Filtering

4.5.1 Purpose

After whitelist has been extracted, we can keep the network traffic of IoT system secure by filtering unwanted traffic.

4.5.2 Method

In this experiment, we created a filter by combining `iptables` and `NetfilterQueue-python` . `iptables` is a Linux command that allows user to filter network packets by configure tables of IP packet filter rules in the Linux kernel. We wrote a program that take the JSON output of whitelist extraction program as an input a create an `iptables` iptables script as its output. Next is the template of our script.

```
1 # iptables -F
2 # iptables -P FORWARD DROP
3 # iptables -A FORWARD -p [udp|tcp] --dport [port number] -d [
    device secure host IP] -s [device ip] -j ACCEPT
4 # iptables -A FORWARD -p [udp|tcp] --dport [port number] -d [
    device ip] -s [device secure host IP] -j ACCEPT
```

- 1st line, -F command is to clean the exist rule in iptables.
- 2nd line states that our default policy for any packet FORWARD through is computer is DROP
- 3rd and 4th line states that only device's hosts with specific protocol, port number can communicate with the device.

Then we ran this script on computer bridging between the internet and the switch, or the switch and devices (Figure 4.4). This allowed us to filter all unnecessary packets and kept our IoT system secure. Next step, we wanted to evaluate the performance of our whitelist. Therefore, instead of dropping and accepting at the iptables, we configured iptables to pass all dropped packets to our "NetfilterQueue" python program. NetfilterQueue is a module that provides access to packets matched by iptables rule, we can analyze those packets using "kamene" (also known as "scapy").

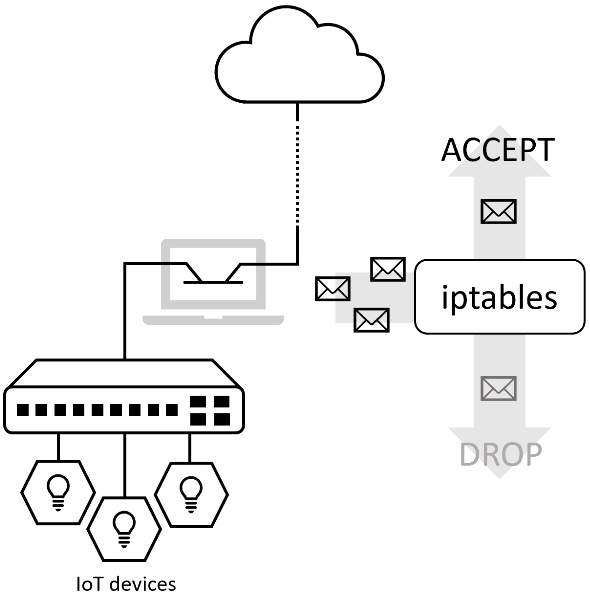


Figure 4.4: Packet Filtering using iptables

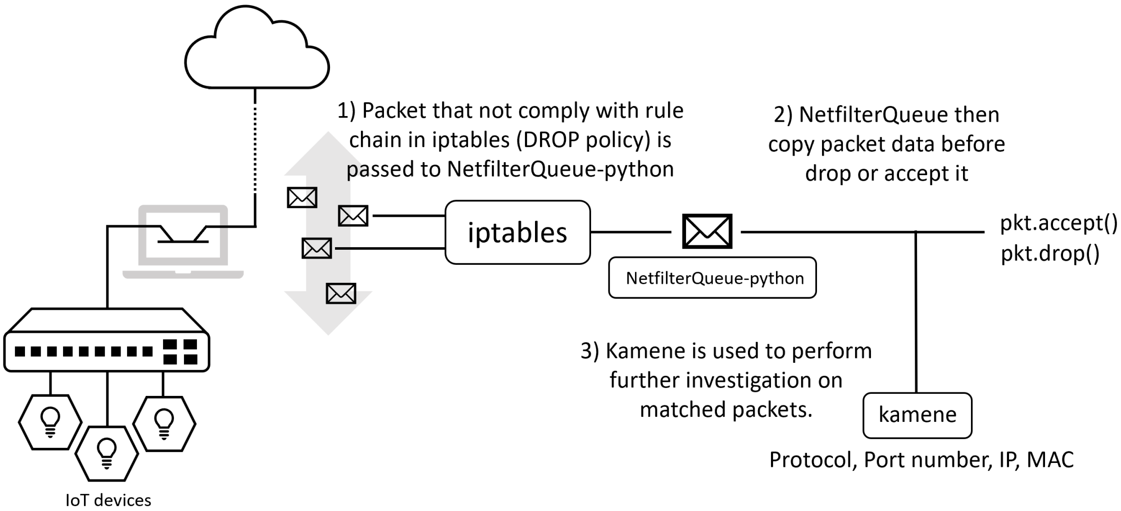

1) Packet that not comply with rule chain in iptables (DROP policy) is passed to NetfilterQueue-python
2) NetfilterQueue then copy packet data before drop or accept it
iptables
NetfilterQueue-python
pkt.accept()
pkt.drop()
3) Kamene is used to perform further investigation on matched packets.
kamene
Protocol, Port number, IP, MAC
IoT devices

Figure 4.5: Analyzing matched packet by iptables using NetfilterQueue-python module and kamene