

Signature-based and Behavior-based Attack Detection with Machine Learning for Home IoT Devices

Vasaka Visoottiviset, Pranpariya Sakarin, Jetnipat Thongwilai, Thanakrit Choobanjong
Faculty of Information and Communication Technology, Mahidol University
Nakhon Pathom, Thailand

vasaka.vis@mahidol.edu, {pranpariya.sak, jetnipat.tho, thanakrit.cho}@student.mahidol.ac.th

Abstract—Currently, Internet of Things (IoT) becomes pervasive and widely deployed. However, the lack of developer and user cyber security awareness leaves IoT devices become the new target of cyber attacks. Therefore, we design and develop "A System for Preventing IoT Device Attacks on Home Wi-Fi Router" (SPIDAR) in order to protect home Wi-Fi networks. This system consists of SPIDAR home Wi-Fi router, SPIDAR Raspberry Pi, and SPIDAR web application to prevent attacks and display the attack statistics to home users. It also helps saving costs from purchasing expensive intrusion prevention software and hardware to install at home. For the prevention method, we provide both the signature-based method using Snort software and the behavior-based method which learns and analyzes IoT devices' behavior by using either the baseline or the machine learning in order to increase the system performance. SPIDAR can prevent five major attack types specified in the OWASP IoT Top 10 vulnerabilities 2018.

Keywords—internet of things, cyber security, intrusion prevention system, behavior-based detection, smart home

I. INTRODUCTION

Nowadays Internet of Things (IoT) are integrated into many devices and play important roles in our daily life, such as smart home, smart city, smart factory, smart healthcare, etc. However, the security of IoT devices is not truly recognized by some manufacturers and users. Many IoT devices are shipped with many software vulnerabilities; and most users install them without the security awareness. Many users even setup them with a simple or well-known password, or never change the default password at all.

Since IoT devices are ubiquitous and widely spread, they are new targets for attackers. In 2017, Mirai botnet that targets IoT devices was discovered [1]. Security researchers found that it turned IoT devices, mostly IP cameras and home routers, into remotely controlled bots and used them as part of a botnet in large-scale network attacks. Mirai infected a large number of 600,000 vulnerable IoT devices at peak due to the use of default usernames and passwords.

Security threats in IoT devices occur because of the lack of security awareness from both the manufacturers and the user. For the IoT software or firmware, some developers did not follow the secure coding guidelines and did not perform the penetration testing before shipping the products. When someone realizes a software vulnerability, it takes sometimes for developers to fix the bugs and publish a patch or firmware for users to update. However, some users are also lack of the security awareness. Even the new security patch or the new firmware is published, they do not update the

firmware to their devices or do not know how to update it. For example, there are some home Wi-Fi routers that do not get updated for many years. Moreover, they are some devices that cannot be updated easily by users, such as smart plugs. Further, many users use IoT devices without changing the password or use an easy-to-hack or well-known popular password.

In the security area, they are offensive and defensive security. Offensive security focuses on finding and fixing system vulnerabilities, while defensive security is a proactive approve to protect systems from attackers. Offensive security requires the security awareness from both the developers and the users. Without the awareness of users, IoT devices will still be vulnerable even the new security patch is published. Therefore, in this paper we focus in the defensive security. Our objective is to protect IoT devices in the home Wi-Fi network from the external attackers.

For the enterprise network, intrusion prevention systems (IPS) are normally introduced in order to detect and prevent possible attacks from the external network. However, the hardware IPS is too expensive for a small home network. Therefore, the objective of this research is to design and implement a low-cost intrusion prevention system for protecting IoT devices inside the home network. We propose SPIDAR which stands for "A System for Preventing IoT Device Attacks on Home Wi-Fi Router" as an intrusion prevention system to impede popular IoT attacks as listed in the OWASP IoT Top 10 vulnerabilities 2018 [2]. Our contributions are as follows.

- We design and develop a low-cost Intrusion Prevention System (IPS) for home IoT devices. Our system can be installed on many existing home Wi-Fi routers.
- SPIDAR system can detect and prevent five types of IoT attack, which are (1) bruteforce password attack, (2) DoS/DDoS attack, (3) cross-site scripting (XSS) attack, (4) SQL injection attack, and (5) the evil twin attack.
- We also design and develop a web application for the home owner to view the attack statistics.
- SPIDAR can support both the signature-based and the behavior-based anomaly detection. The behavior-based detection can be achieved by using either the baseline or by using the machine learning.

II. BACKGROUNDS

A. Opensource Home Wi-Fi Router Firmware

Even most of normal cheap home Wi-Fi routers include the firewall function, they do not contain the intrusion detection/prevention module. For the firewall function, the users have to define the rules for dropping packets in advance. Users requires the security knowledge and specify the port numbers and/or the IP address of the source machines in advance by themselves. Moreover, attackers can dynamically change the source IP address and the port number, therefore the static firewall configuration cannot block all attacks.

In our design, we want to implement our solutions on the existing home Wi-Fi router, so that we can save costs of buying a new expensive hardware box. Based on our survey, we found that there are three popular opensource firmware for home Wi-Fi router, which are OpenWRT [3], DD-WRT, and Tomato. Table I compares the characteristics of these three firmware. We found that OpenWRT cloud support much more number of home Wi-Fi routers and could deliver a good network performance. Therefore, we select to implement the SPIDAR system by using OpenWRT.

B. Techniques for Intrusion Detection

There are two main methods to detect anomaly traffic: (1) rule-based or signature-based detection and (2) behavior-based or anomaly-based detection. The former one requires security experts to create pre-defined rules or signatures in advance from known attack types, while the latter one requires the system to learn the normal pattern or create a baseline in order to detect the abnormal activities. Behavior-based detection can be implemented by just comparing thresholds or by using the machine learning.

Table II compares the characteristics of these two detection methods. Rule/signature-based detection can work faster, but cannot identify new attack types. On the other hand, behavior/anomaly-based detection may work slower and can contain false positive, but it can detect new attack types. In our SPIDAR design, we provide both detection modes for users to select.

TABLE I. COMPARISON OF OPENSOURCE HOME WI-FI ROUTER'S FIRMWARE CHARACTERISTICS

Characteristics	Firmware		
	OpenWRT	DD-WRT	Tomato
Number of supported routers	1,367	815	60
QoS and VPN support	✓	✓	✓
Traffic Analysis	✓	✓	✓
Lightweight	✗	✗	✓
Interface	Command line or web-based interface, but not user-friendly	Web-based interface, but not user-friendly	User-friendly web-based interface
Stability and performance	Reduce latency and increase throughput	Unstable	Less bugs than DD-WRT

C. OWASP IoT Top 10 Vulnerabilities

Open Web Application Security Project (OWASP), which is a non-profit organization for security experts to exchange security knowledge to make web applications

secure. OWASP first published OWASP IoT Top 10 vulnerabilities in 2014, and updated this list in 2018. The list contains (1) I1 - Weak, Guessable, or Hardcoded Passwords, (2) I2 - Insecure Network Services, (3) I3 - Insecure Ecosystem Interfaces, (4) I4 - Lack of Secure Update Mechanism, (5) I5 - Use of Insecure or Outdated Components, (6) I6 - Insufficient Privacy Protection, (7) I7 - Insecure Data Transfer and Storage, (8) I8 - Lack of Device Management, (9) I9 - Insecure Default Settings, and (10) I10 - Lack of Physical Hardening.

However, the topics supported by SPIDAR are I1, I2 and I3. Table III shows the attack types mapping with each OWASP IoT Top 10 IoT vulnerabilities.

D. Related Works

Some existing research works for home IoT security also focus on either the intrusion detection system (IDS) or intrusion prevention system (IPS) for IoT devices.

Projects [4] and [6] are both IDS systems and use the OpenWRT firmware. They can just alert but cannot drop anomaly packets. Fu, et al. [4] proposed an intrusion detection mechanism by using finite state machine or finite automata to describe the traffic flows of wireless sensor network (WSN). This research work utilizes the OpenWRT as the IoT gateway and can automatically detect and report three types of IoT attacks which are jamming attack, malicious code attack, and replay attack. On the other hand, the Snort on OpenWRT project [6] provides a secure perimeter for SOHO (Small Office Home Office) LAN by using a router running OpenWRT firmware and Snort IDS [5].

For IPS, the research work [7] proposed a system to learn the behavior of malicious network activities in order to detect and prevent DoS attack, ARP cache poisoning, malformed packets and botnets. This work employed several types of machine learning algorithms.

TABLE II. CHARACTERISTICS OF DETECTION METHODS

Characteristics	Detection Method	
	Rule/Signature-Based Detection	Behavior/Anomaly-Based Detection
Mechanism	Use rules to identifies the packet	Analyze from the behavior
Advantages	Work faster	Can detect unknown attack types
Limitations	Unable to identify unknown and emerging threats	Work slower, False positive output

TABLE III. MAPPINGS BETWEEN OWASP TOP 10 IoT 2018 AND ATTACK TYPES

OWASP IoT Top 10 2018 Topic	Attack Types
I1- Weak, guessable, or hardcoded passwords	Bruteforce Password attack
I2- Insecure network services	DoS/ DDoS, Evil twin attack
I3- Insecure ecosystem interfaces	Cross-site scripting (XSS), SQL Injection

III. SYSTEM DESIGN

We design and develop the SPIDAR system to handle and prevent malicious packets that target home IoT devices.

This scenario is presented when an attacker tries to exploit IoT devices which are connected to home Wi-Fi router.

A. System Overview

Fig. 1 illustrates the system overview of SPIDAR system. It consists of three main components, which are (1) SPIDAR home Wi-Fi router, (2) SPIDAR Raspberry Pi, and (3) SPIDAR web server.

In order to block anomaly traffic coming into the home IoT network, we have to utilize the firewall function on the Wi-Fi router. In the initial design, we planned to install Snort IPS on the Wi-Fi router and drop anomaly packets directly. However, because of the small storage size and low processing power of the router, Snort cannot run smoothly. Therefore, we install the OpenWRT firmware on the Wi-Fi router and set the task to perform the port mirroring in order to copy all traffic to the SPIDAR Raspberry Pi which will detect anomaly traffic. Moreover, SPIDAR Raspberry Pi distinguishes the mirrored traffic from other traffic by selecting only packets that contains one of IP addresses of our IoT devices in either the source IP address or the destination IP address. For the connection from the home WiFi router to SPIDAR Raspberry Pi, we use the onboard 10/100 Mbps Ethernet interface. SPIDAR Raspberry Pi can connect with other home IoT devices via this LAN interface.

SPIDAR Raspberry Pi can handle either of two main detection modes. The first mode is signature-based mode which compares the incoming packets with a set of specified rules using the well-known Snort IDS. The second mode is behavior-based detection that analyzes and identifies the behavior of attacks by comparing with common network activities of IoT devices.

For the signature-based mechanism, SPIDAR Raspberry Pi will examine the Snort alert logs; and we will then use a shell script to extract the attacking flow information such as the IP address and port number. The anomaly flow information then is sent back to the SPIDAR home Wi-Fi router and will be used by the firewall function in order to block the anomaly packets.

On the other hand, the behavior-based attack detection is divided into two modes: the baseline and machine learning. The former mode has shell scripts to calculate captured packets by using some communication features, such as the data rate and packet size for both inbound and outbound directions. The latter mode is processed by using machine learning to determine whether the incoming packets are the attack or not from the labelled training dataset.

After that, the anomaly log is sent from SPIDAR Raspberry Pi via the home WiFi router to the SPIDAR web server which is located on the Internet. The home users can view the summary of anomaly traffic alert reports on this SPIDAR web server and learn how many attacks going on in their home network.

Fig. 2 shows the modules of each SPIDAR component in details. SPIDAR can be separated into three main components which are home Wi-Fi router, Raspberry Pi and web application. In home Wi-Fi router, there are two subparts, which are port mirroring and automatic firewall configuration. Moreover, modules on Raspberry Pi contain main three subparts, which are device configuration, attack detection, and transferring of anomaly flow information. Because the evil twin attack detection has to check the radio

frequency, this module will be used in both signature-based mode and behavior-based mode. For the web application, it contains authentication, information obtaining, information displaying and detection mode selection.

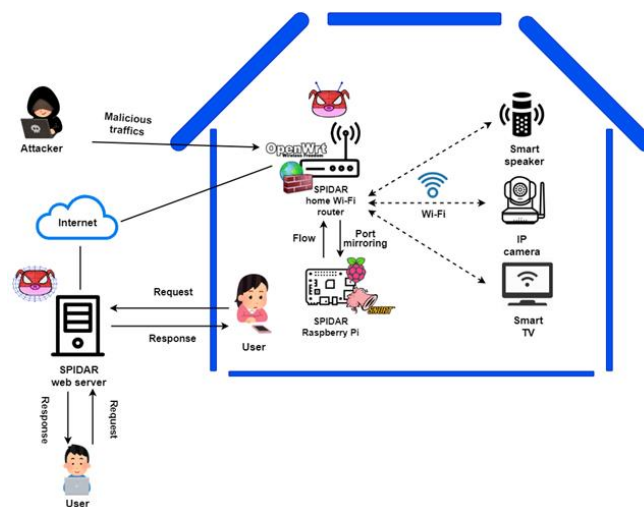


Fig. 1. SPIDAR system architecture

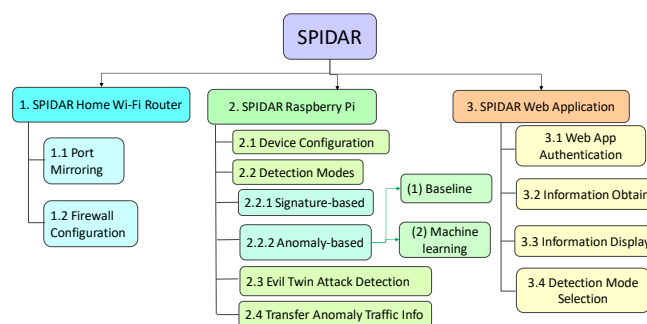


Fig. 2. SPIDAR system structure chart



Fig. 3. Transition diagram of SPIDAR Web application

Fig. 3 shows the transition diagram of SPIDAR web application. The home user can select the detection mode whether the signature-based or anomaly-based from the configuration page. For the behavior-based detection, the system will automatically learn IoT devices in the home network and display their information such as the name, the type, the vendor, the IP address, and the MAC address. The vendor name actually can be obtained from looking up from the beginning 24 bits of MAC address, which are the

Organizationally Unique Identifier (OUI). Moreover, the user can modify the name by themselves. After the system learns how many IoT devices are in the home network, the system will learn their behaviors such as the average bandwidth usage and the average packet size for both inbound and outbound directions. The SPIDAR web application also contains the learning center which is a page that displays the details of each supported attack type.

B. Anomaly Traffic Detection and Prevention

SPIDAR supports five attack types, which are the (1) brute force password attack, (2) DoS/DDoS attack, (3) cross-site scripting (XSS) attack, (4) SQL injection attack, and (5) the evil twin attack. However, the evil twin attack cannot be detected by observing the network traffic. We will mention about its solution later.

SPIDAR supports both the signature-based and behavior-based modes. The detection module is located on the SPIDAR Raspberry Pi so that the Wi-Fi router can focus only on forwarding packets.

1) Signature-based Detection

In this mode, SPIDAR will use the Snort IDS engine on the Raspberry Pi. Since the number of enabled Snort rules will affect the computing performance, we select only some Snort rules for our targeted attack types, except the evil twin attack. The detection results are output as the Snort alert log files.

2) *Behavior-based Detection*: There are two modes, which are detection with the baseline and detection by using the machine learning.

a) Baseline

The system will learn the behavior of each IoT device for a while in order to create a baseline or thresholds to compare with the current traffic pattern. In SPIDAR, the system will maintain the average packet size and the data rate of both incoming and outgoing direction. For example, for the IP camera that regularly transfers the video images to the server, the data rate of outgoing traffic from the IP camera is normally higher than the incoming traffic rate, because the incoming traffic contains just some packets that the user sends to control or configure the IP camera. If the system can observe a high data rate coming to the IP camera, the system will consider this data as a possible DoS attack.

b) Machine learning

Instead of using the baseline or threshold, the system can use the machine learning to classify and detect anomaly traffic. This mode requires the learning dataset of each attack types and the dataset of normal traffic. In our machine learning classification, there are four classes of attacks and multiple classes for normal traffic depending on how many IoT devices we have. For example, if there are three IoT devices in the home network, then there will three classes of normal traffic. To classify types of traffic, SPIDAR applies the Random Forest algorithm. Features for classification are the protocol, source and destination IP address, source and destination port number, the average data rate, and the average packet size.

The results of detection in each mode will be extracted as the anomaly flow information and sent to the SPIDAR Wi-Fi router to automatically block those traffics by using the firewall function. Fig. 4 summarizes the tasks in each

detection/prevention mode that are supported by the SPIDAR system.

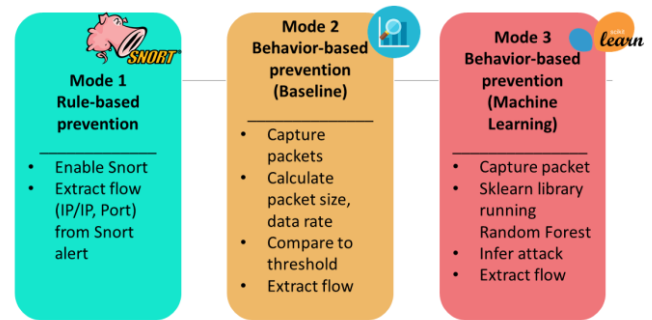


Fig. 4. Details of each detection/prevention mode supported in SPIDAR

C. Detection of Evil Twin attack

Evil Twin attack or rogue access point is a common attack type for the Wi-Fi communication. This attack happens in the radio layer and cannot be detected by observing the network traffics like other attack types. Even the evil twin attack is difficult to block, SPIDAR system can detect and alert to the home users. The SPIDAR Raspberry Pi will store the known MAC address of the legitimate access point and the received signal strength or signal-to-noise ratio (SNR). Because the MAC address can be spoofed, the SNR value is the most important threshold to determine whether that access point is the legitimate one or the rogue one; and a rogue access point is normally deployed outside the house and has the lower signal strength. Once SPIDAR found that there was another access point with the same SSID, but different MAC address or the value SNR is much different than the one in the database, for example different more than 30%, SPIDAR system will send an alert to the web server to inform the home user.

IV. IMPLEMENTATION AND EXPERIMENTS

A. Hardware and Software Specifications

Table IV, V, and VI show the hardware and software of specifications of SPIDAR home Wi-Fi router, SPIDAR Raspberry Pi and SPIDAR web server, respectively. Table VII shows the specifications of IoT devices used in our experiments.

TABLE IV. SPECIFICATIONS OF SPIDAR HOME WI-FI ROUTER

Components	Name	Descriptions
Hardware	D-Link DWR-921 Hardware version C3	Router model
	OpenWrt 18.06.5 r7897-9d401013fc / LuCI openwrt-18.06 branch	Router firmware
Software	Crontab	Command for task scheduling
	IPtables, ipset	Port mirroring and firewall
	Tcpdump	Detecting network packets
Programming language	Shell script	Automate process on the router




TABLE V. SPECIFICATIONS OF SPIDAR RASPBERRY PI

Components	Name	Descriptions
Hardware	Raspberry Pi 3 Model B+ with 64-GB SD card	Raspberry Pi board
Software	Snort	Intrusion detection engine
	Crontab	Command for task scheduling
	Tcpdump	For analyzing the traffic
	Tshark	For extracting the important flow information from the captured file.
	nmap	Tools for scanning IoT devices
	SSHpass	Running ssh with the password authentication
	Scikit-learn	Machine learning library for the Python
Programming language	Inotify-tools	Monitor and act upon the file system
	Shell script	Automate process on the router

TABLE VI. SPECIFICATIONS OF SPIDAR WEB SERVER

Components	Name	Descriptions
Hardware	Intel Core i7-8700K CPU 3.70 GHz RAM 8 GB	PC specifications
Software	Wordpress	Content management system for building a website
	Apache	Web server software
	phpMyAdmin	To manage the database
Programming language	HTML	Markup language for web
	CSS	Style sheet
	JavaScript	To build application functions
	PHP	To create server-side scripts

TABLE VII. SPECIFICATIONS OF IoT DEVICES USED IN EXPERIMENTS

Name	Image	Specifications
Smart Wi-Fi socket by Lampton		Wireless standard: Wi-Fi 2.4 GHz b/g/n Security protocol: WEP/WPA-PSK/WPA2-PSK
Smart Wi-Fi Plug HS110 by TP-Link		Wireless standard: Wi-Fi 2.4 GHz b/g/n
Smart Wifi Camera by Lampton		FPS: 30 fps Resolution: 720P/VGA/QVGA Input: 5V DC/1A

B. Tools for Generating Attacks

To create the learning dataset for machine learning and to perform the experiments, we have to generate the attacking traffic by using tools and techniques specified in Table VIII.

C. Experiment Results of Evil Twin Attack Detection

Because the evil twin attack detection has to check with the radio frequency's signal strength, we cannot detect this attack type by sniffing packets passing through the home router like other attack types. Therefore, we separate this module out from the Snort detection and behavior based detection.

To verify our detection mechanism, we placed SPIDAR Raspberry Pi and the home router for approximately 20 cm apart, and placed a rogue access point with the same SSID out of the experiment room. The distance between the rogue access point and the legitimate access point was around five meters. Based on this experimental setup, we observed the received signal strength on the Raspberry Pi on different time and different days for totally ten times and found that the range is between -7 and -9 dB. However, the signal strength of the rogue access point that we observed for ten times was between -44 and -51 dB. Thus, our SPIDAR system can completely detect the evil twin attack.

D. Experiment Results of Snort Detection

Table IX shows the results of Snort detection after we injected the bruteforce password attack, DoS/DDoS attack, SQL injection, and XSS by using tools specified in Table VIII. Note that false alerts of SQL injection happened because Snort interpreted those packets as XSS attacks.

E. Experiment Results of SPIDAR Processing Time with Snort

Table X shows the processing time of each task when SPIDAR is configured to use the Snort signature-based detection. The processing time of each attack was measured since the time we injected the attack traffic until the time the SPIDAR Wi-Fi router could block the traffic. For XSS attack, since the attacker can perform the attack with only one packet, the processing time cannot be measured. However, once SPIDAR can detect the XSS attack, SPIDAR will block the source IP address of the XSS attacker immediately.

TABLE VIII. MAPPINGS BETWEEN ATTACK TYPES AND TOOLS USED IN EXPERIMENTS

Attack Types	Attack Tools or Techniques
Password attack	Sqlmap
DoS/ DDoS	Hydra
SQL Injection	Metasploit
Cross-site scripting (XSS)	Metasploit
Evil twin attack	Access point with the same SSID and channel

TABLE IX. RESULTS OF SNORT DETECTION

Attack Type	Number of Alerts	Number of False Alerts	Accuracy
Bruteforce password	7	0	100
DoS/DDoS	291952	0	100
SQL injection	1164	9	99.23
XSS	89	0	100

TABLE X. RESULTS OF SPIDAR PROCESSING TIME WITH SNORT

Tasks	Processing Time (seconds)		
	Minimum	Maximum	Average
Bruteforce	27.98	29.51	28.57
DoS/DDoS	15.12	18.50	16.42
SQL injection	6.01	7.74	6.76

F. Dataset for Behavior-based Detection

Based on the attack tools listed in Table VIII, we generate the attack traffic and use them as the dataset for behavior-based detection. This dataset is used for both behavior-based detection with baseline and with the machine learning. Since we assume there are three IoT devices as shown in Table VII, there are seven types of traffic flows. Table XI shows the details of dataset.

G. Experiment Results of Behavior-based Detection with Baseline

For the behavior-based detection with baseline, we calculate the minimum data rate, maximum data rate, average data rate, minimum packet size, maximum packet size and the average packet size for both the inbound and outbound traffic of three IoT devices. These thresholds are used to compare with the new traffic to determine whether the traffics are anomaly or not. We test our approach with each attack type for ten times. Table XII shows the detection results of the behavior-based detection using the baseline. Unfortunately, the system sometimes incorrectly reported the bruteforce password attack as the normal traffic.

H. Experiment Results of Behavior-based Detection with Machine learning.

For the behavior-based detection with machine learning, we labelled the dataset specified in Table XI as seven classes of data and fed them to the Random Forest algorithm to generate a classification model. The results of 10-fold cross validation reveals that our model can achieve 98% of accuracy.

Later, we verify our model with another set of the real traffic. As shown in the results in Table XIII, there are some misclassified flows, especially for the SQL injection and TP-Link smart plug. Some attack flows are classified in the different attack class, but none of them are classified as the normal traffic. For example, some flows of SQL injection are classified as Bruteforce password attack. Moreover, some normal traffic flows are also misclassified, but none of them are classified as the attack flows. For example, the flows from TP-Link smart plug are classified as flows from Lamptan smart plug. Eventually, it means that SPIDAR can block all attack traffics.

I. CPU Utilization of Each Detection Mode

We found that all three modes can efficiently detect and block our selected types of attacks. However, we found that the CPU utilization of our Raspberry Pi is different for each situation. The average CPU utilization when running Snort, when using the baseline and when using the machine learning are 10.62%, 11.02%, and 25.85%, respectively.

V. CONCLUSIONS AND FUTURE WORK

To prevent IoT devices in the home network from the popular attacks, we proposed and implemented a low-cost SPIDAR system which consists of the SPIDAR home Wi-Fi router using OpenWRT firmware, the SPIDAR Raspberry Pi, and the SPIDAR web application. To detect attacks, we provided three detection mode: signature-based mode by using Snort, behavior-based mode from learned thresholds, and behavior-based mode by using machine learning. We found that both signature-based mode and the behavior-based mode with machine learning can block all attacks, but the signature-based mode used the less CPU utilization. For

the future works, we want to support more types of IoT attacks and employ VLAN separation to prevent attacks from compromised IoT devices to our SPIDAR Raspberry Pi.

TABLE XI. DATASET FOR BEHAVIOR-BASED DETECTION

Type	Total size (Bytes)	Number of Packets	Number of flows
Bruteforce password	2,707,534	18,520	653
DoS/DDoS	4,309,192	105,938	10
SQL injection	83,407,668	305,664	26,818
XSS	32,484	112	18
IP camera	2,656,086	10,821	589
Lamptan smart plug	2,242,631	19,090	27
TP-Link smart plug	1,559,284	12,048	103

TABLE XII. RESULTS OF BEHAVIOR-BASED DETECTION WITH THE BASELINE

Attack Type	Number of Detection	Number of Incorrect Detection	Accuracy (%)
Bruteforce password	9	1	90
DoS/DDoS	10	0	100
SQL injection	10	0	100
XSS	10	0	100

TABLE XIII. RESULTS OF BEHAVIOR-BASED DETECTION USING MACHINE LEARNING WITH SEVEN CLASSES

Type	Total Amount	Correctly classified	Incorrectly Classified	Accuracy (%)
Bruteforce password	189	185	4	0.978
DoS/DDoS	11225	11225	0	1.0
SQL injection	34606	188	34418	0.0054
XSS	89	89	0	1.000
IP camera	137	67	70	0.489
Lamptan smart plug	340	323	17	0.95
TP-Link smart plug	126	0	126	0.0

REFERENCES

- [1] The Cloudflare Blog, "Inside the infamous Mirai IoT Botnet: A Retrospective Analysis". <https://blog.cloudflare.com/inside-mirai-the-infamous-iot-botnet-a-retrospective-analysis/>
- [2] OWASP Internet of Things Project, "Owasp IoT Top 10 2018". https://www.owasp.org/index.php/OWASP_Internet_of_Things_Project#tab=OWASP_IoT_Top_10_2018_Mapping_Project
- [3] OpenWrt Project. <https://openwrt.org/>
- [4] Fu, Y., Yan, Z., Cao, J., Koné, O. and Cao, X., "An Automata Based Intrusion Detection Method for Internet of Things", Hindawi Mobile Information Systems, 2017, pp.1-13.
- [5] Snort - Network Intrusion Detection & Prevention System. <https://snort.org/>
- [6] Linux.com., "Snort on OpenWrt: Guarding the SOHO perimeter". <https://www.linux.com/news/snort-openwrt-guarding-soho-perimeter/>
- [7] I. Indre and C. Lemnaru, "Detection and prevention system against cyber attacks and botnet malware for information systems and Internet of Things," 2016 IEEE 12th International Conference on Intelligent Computer Communication and Processing (ICCP), Cluj-Napoca, 2016, pp. 175-182.