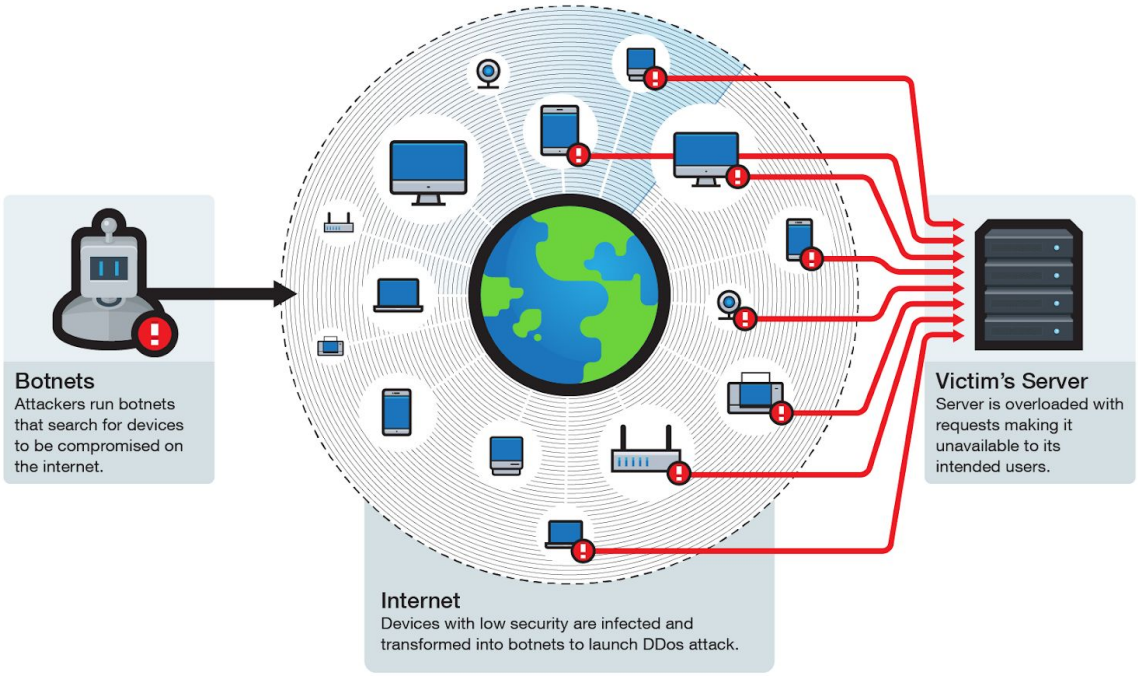
Botnet Detection on IoT Devices



Dineshkumar Sundaram

https://github.com/dineshh912

INTRODUCTION

Due to Increasing usage of digital communication in this digital era, cyber security is crucial to maintain a high level of safety. To prevent an increasing number of cyber attacks, traditional security system firewalls, encryption is not enough. There is a need for intrusion detection systems that can integrate with traditional systems and assure a high level of security of data.

Detecting strange anomalies is a good way to raise alerts for security administrators to investigate threats and new zero-day attacks that cannot be discovered by traditional security systems.

Internet of Things (IoT) devices are widely used in modern homes and became every part of our lives, because they are not that sophisticated, it becomes an easy target for Denial of service attack. IoT devices can be used as bots to launch a distributed DOS attack.

The rapid growth of IoT devices which can be more easily compromised than desktop computers has led to an increase in the occurrences of IoT based botnet attacks. Botnet attack is a type of DDOS attack, where the attacker uses a large number of IoT devices to participate in the DOS to overwhelm a specific target. This type of attack is hard to detect, since the device keeps functioning normally, and the user or the owner of the device will not notice if his device is a part of an attack, in some cases the device may suffer from delay of its functionality.

Botnets such as Mirai are typically constructed in several distinct operational steps.

- propagation
- infection
- C&C communication
- execution of attacks.

Dataset

Data gathered from nine commercial IoT devices infected by authentic botnets from two families.Mirai and BASHLITE, two of the most common IoTbased botnets, which have already demonstrated [1] their harmful capabilities. The dataset can be downloaded from https://archive.ics.uci.edu/ml/datasets/detection_of_IoT_botnet_attacks_N_BaIoT.

Data collection : Data capture the raw network traffic data (in pcap format) using port mirroring on the switch through which the organizational traffic typically flows. To ensure that the training data is clean of malicious behaviors, the normal traffic of an IoT is collected immediately following its installation in the network.

Feature extraction: Whenever a packet arrives, we take a behavioral snapshot of the hosts and protocols that communicated this packet. The snapshot obtains the packet's context by extracting 115 traffic statistics over several temporal windows to summarize all of the traffic that has

1. originated from the same IP in general.
2. originated from both the same source MAC and the same IP address.
3. been sent between the source and destination IPs (channel), and
4. been sent between the source to destination TCP/UDP sockets (socket).

The same set of features captured from five time windows of the most recent 100ms, 500ms, 1.5sec, 10sec, and 1min. These features can be computed very fast and incrementally and thus facilitate real time detection of malicious packets. Additionally, although generic these features can capture specific IEEE PERVASIVE COMPUTING, VOL. 13, NO. 9, JULY-SEPTEMBER 2018 4 behaviors like source IP spoofing , characteristic of Mirai's attacks. For instance, when a compromised IoT device spoofs an IP, the features aggregated by the Source MACIP, Source IP and Channel will immediately indicate a large anomaly due to the unseen behavior originating from the spoofed IP address.

The dataset contains the following nine device normal & attack traffic.

- Danmini - Doorbell
- Ennio - Doorbell
- Ecobee - Thermostat
- Philips B120N/10 - Baby Monitor
- Provision PT-737E - Security Camera
- Provision PT-838 - Security Camera
- Simple Home XCS7-1002-WHT - Security Camera
- Simple Home XCS7-1003-WHT - Security Camera
- Samsung SNH 1011 N - Web cam

The above 115 features are derived from a set of 23 features.the same set of 23 features extracted from five time windows of the most recent 100ms, 500ms, 1.5sec, 10sec, and 1min. These features can be computed very fast and incrementally and thus facilitate real time detection of malicious packets.

Value	Statistics	Aggregated by	Total Features
Packet size (outbound packet only)	Mean, Variance	Source IP, Source MAC-IP, Channel, Socket	8
Packet Count	Number	Source IP, Source MAC-IP, Channel, Socket	4
Packet jitter (the amount of time between packet arrivals)	Mean, Variance, Number	Channel	3
Packet size (of both inbound and outbound together)	Magnitude, Radius, Covariance, Correlation coefficient	Channel, Socket	8

1. Source IP - Used to track the host as a whole
2. The source MAC-IP adds the capability to distinguish between traffic originating from different gateways and spoofed IP addresses.
3. The sockets are determined by the source and destination TCP or UDP port numbers. For example, all of the traffic sent from 192.168.1.12:1234 to 192.168.1.50:80 (traffic flowing from one socket to another).

Exploratory Data Analysis

Device	Benign	Mirai	Bashlite
Philips B120N10 Baby Monitor	175240	610714	312723
Provision PT 838 Security Camera	98514	429337	309040
Provision PT 737E Security Camera	62154	436010	330096
Samsung SNH 1011 N Webcam	52150	-	323072
Danmini Doorbell	49548	652100	316650
Simple home XCS7 1002 WHT Security Camera	46585	513248	303223
Ennio Doorbell	39100	-	316400
Simple home XCS7 1003 WHT Security Camera	19528	514860	316438
Ecobee Thermostat	13113	512133	310630

Data doesn’t have any NaN values. (Detailed data information available on github reports folder)

Since 115 features is a bit difficult to plot on the surface, we used dimensionality reduction method to mathematically reduce the feature into 2 & 3 features and plotted on the scatter plot.

Fig -1 Danmini Doorbell - 2D Scatter Plot

Fig -2 Philips B120N10 Baby Monitor

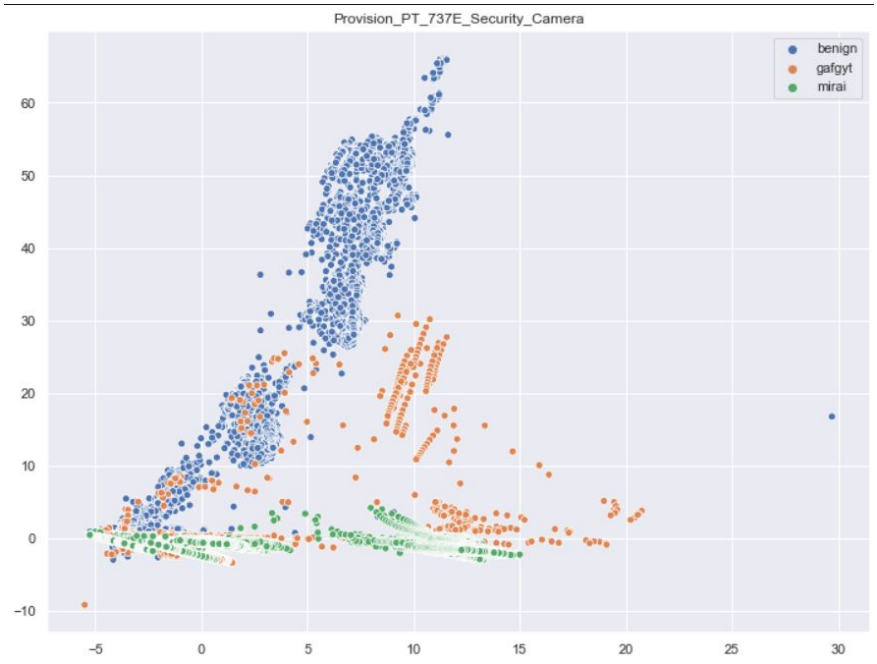


Fig -3 Provision Security camera

This 2d scatter plot gave some interesting trends between benign and mirai/bashlite data. Based on this plot the labels are clearly visible. At the same time baby monitors have more benign data than other IoT devices. And data spread across the plot looks like there is more activity happening on the baby monitors than other devices.

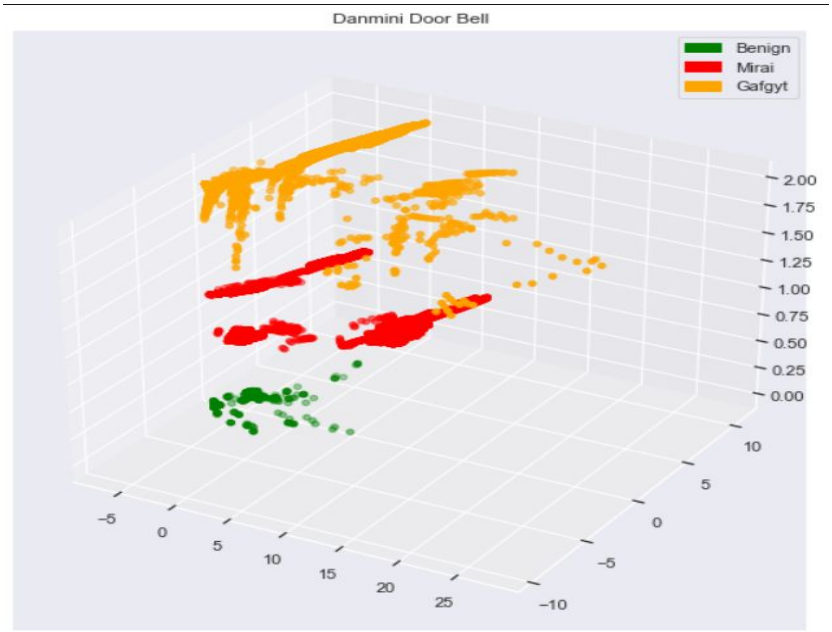


Fig-4 Danmini Doorbell - 3D Scatter Plot

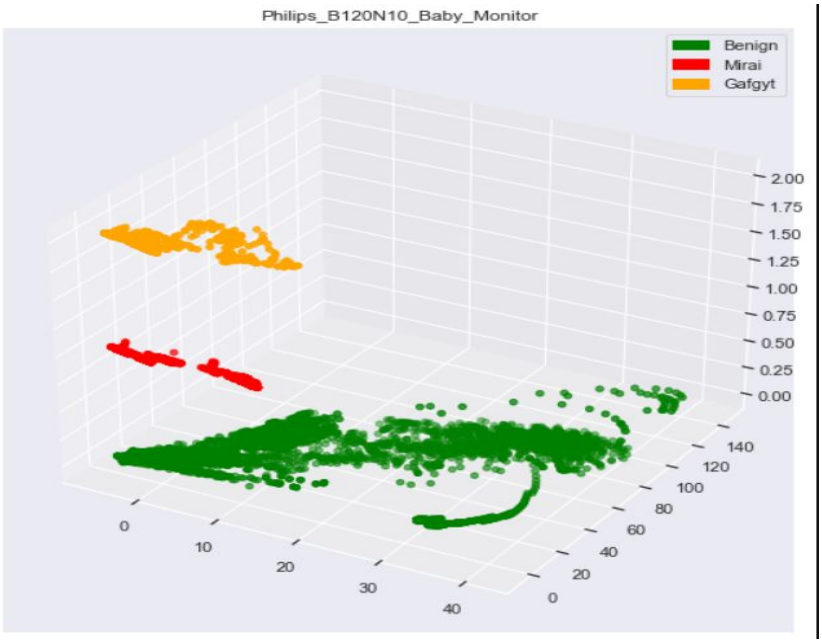


Fig -5 - Philips baby monitor - 3D Scatter Plot

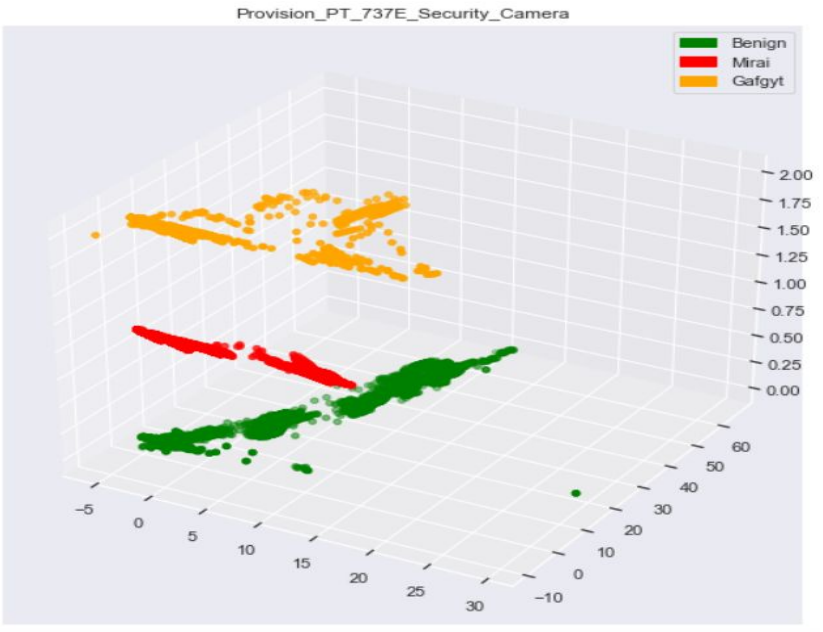


Fig-6 Provision PT-737E Camera - 3D Scatter Plot

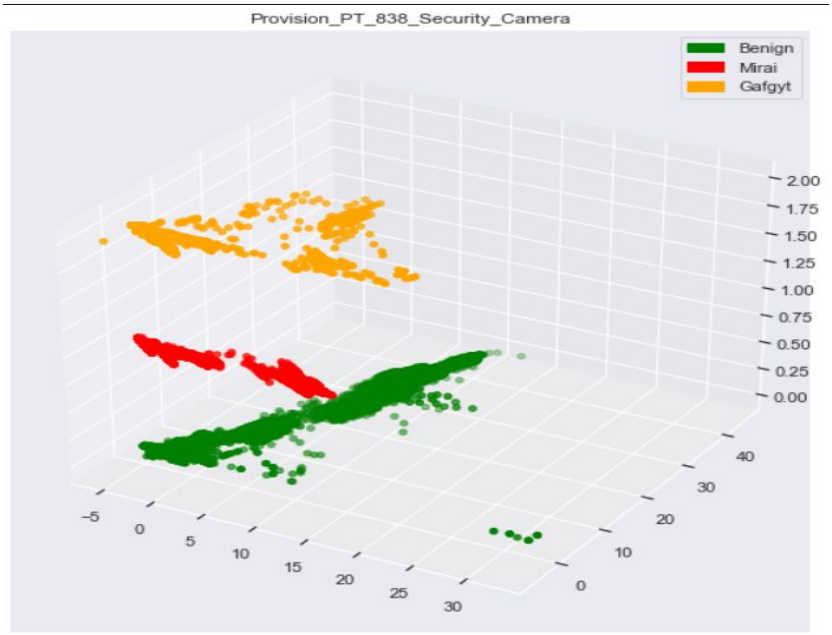


Fig -7 Provision PT 838 Camera - 3D Scatter plot

There is some interesting trend occurring when plotting 3d Scatter plots. Both provision PT-737 and PT 838 camera plots are similar to each other. Both mirai and bashlite malwares behave the same way for the same manufacturer device. Because of this we no need to build a model for each and every device.

Training and Pre processing

I used danmini doorbell data with the Pycaret python module to test it's baseline accuracy and F1 Score.

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
0	Random Forest Classifier	1.0000	0.0	0.9999	1.0000	1.0000	0.9999	0.9999	2.8086
1	Decision Tree Classifier	0.9998	0.0	0.9997	0.9998	0.9998	0.9997	0.9997	21.2064
2	K Neighbors Classifier	0.9980	0.0	0.9935	0.9980	0.9980	0.9960	0.9960	25.6996
3	Ridge Classifier	0.9969	0.0	0.9958	0.9969	0.9969	0.9936	0.9936	1.2116
4	Ada Boost Classifier	0.9245	0.0	0.9202	0.9340	0.9216	0.8392	0.8522	144.0179
5	Quadratic Discriminant Analysis	0.6834	0.0	0.8271	0.8491	0.6724	0.4799	0.5712	5.3659
6	Naive Bayes	0.6585	0.0	0.3543	0.7312	0.5410	0.0693	0.1829	0.8091
7	SVM - Linear Kernel	0.4204	0.0	0.3930	0.4682	0.3959	0.0762	0.1060	6.0382
8	Logistic Regression	0.0486	0.0	0.3333	0.0024	0.0045	0.0000	0.0000	4.2906

Since the data are non-linear, the model is performing well with a random forest and decision tree classifier. I approached the problem as a supervised learning. The above result came from 30% of the data.

There are three different approaches to this problem.

1. Build a generic model which combines all device benign and attacks data.
2. Build a category based model which has data related to the specific device categories like camera, doorbell.
3. Build a model for individual devices.

All above approaches have their merits and demerits. In order to train generic models, we need training data from all the devices that will increase the computing power to train a model.

I chose the third approach to build a model for each device, with this approach I tried 4 different ways. As we concluded from the EDA attacks data are more compared to benign data. This is mainly because attacks data further classified into 10 categories. With that in mind trained a model with 3 classes, 11 classes, 3 classes with under sampled data, 11 class with under sampled data.

Results

Random Forest Classifier - F1 Score

Device	All data with 3 Classes	Under sampled data with 3 classes	All data with 11 Classes	Under sampled data with 11 Classes
Danmini Doorbell	1.0	1.0	1.0	1.0
Ecobee Thermostat	1.0	1.0	0.998	0.988
Ennio Doorbell	1.0	1.0	0.992	0.983
Philips B120N10 Baby Monitor	1.0	1.0	0.997	0.989
Provision PT 737E Security Camera	1.0	1.0	0.993	0.981
Provision PT 838 Security Camera	1.0	1.0	1.0	1.0
Samsung SNH 1011 N Webcam	1.0	1.0	0.999	0.998
SimpleHome XCS7 1002 WHT Security Camera	1.0	1.0	1.0	1.0
Simple Home XCS7 1003 WHT Security Camera	1.0	1.0	0.993	0.970

Decision Tree Classifier - F1 Score

Device	All data with 3 Classes	Under sampled data with 3 classes	All data with 11 Classes	Under sampled data with 11 Classes
Danmini Doorbell	1.0	1.0	0.865	0.574
Ecobee Thermostat	0.997	1.0	0.925	0.770
Ennio Doorbell	0.999	1.0	0.945	0.983
Philips B120N10 Baby Monitor	1.0	1.0	0.857	0.878
Provision PT 737E Security Camera	1.0	1.0	0.781	0.859
Provision PT 838 Security Camera	1.0	1.0	0.799	0.877
Samsung SNH 1011 N Webcam	1.0	1.0	0.892	0.998
SimpleHome XCS7 1002 WHT Security Camera	0.996	1.0	0.913	0.648
Simple Home XCS7 1003 WHT Security Camera	0.997	1.0	0.915	0.847

Future Works

Due to computing power limitations I trained the model only for individual devices. This model can be further optimized and deployed onto the device firmware. This possesses a drawback where we need to train a model for each and every device. Which is not feasible in the long term. So we need to create a generic model which is able to identify the malware regardless of device category.

REFERENCES

1. C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, “DDoS in the IoT: Mirai and Other Botnets,” Computer, vol. 50, no. 7, pp. 80–84, 2017.
2. Y. Meidan, M. Bohadana, Y. Mathov, Y. Mirsky, D. Breitenbacher, A. Shabtai, and Y. Elovici 'N-BaIoT: Network-based Detection of IoT Botnet Attacks Using Deep Autoencoders', IEEE Pervasive Computing, Special Issue - Securing the IoT (July/Sep 2018)