IoT Security: A Cost-Effective Approach

Preventing IoT security attacks and loss of valuable information through the internet and data analysis multiple methods ranging from AI algorithms to AWS/RPI services

John Plaras[†]
ECE
Rutgers University
Piscataway, NJ, USA

Andrew Rezk
ECE
Rutgers University
Piscataway, NJ, USA

Shivani Sunil
ECE
Rutgers University
Piscataway, NJ,
USA

ABSTRACT

With an increasing use of IoT devices, comes a greater risk of vulnerability. In modern times, we have seen a great surge in the use of IoT devices as "According to a 2019 Gartner report, enterprise IoT adoption grew 21.5% from 2018 to the end of 2019, totaling an estimated 4.8 billion devices.*[1]. Although the increasing use of IoT will lead to a smarter tomorrow, this technology does not exist without its downsides. One of the main concerns being the issue of cybersecurity. To back this claim we use the research conducted by an organization who "To assess the current state of the IoT threat landscape….analyzed security incidents throughout 2018 and 2019 across 1.2 million IoT devices in the U.S."[1]. According to the same organization, it was deduced that 98% of all IoT device traffic is unencrypted, i.e., sensitive data is exposed on the network. In case of a cyberattack, an attacker who has broken through the first layer of security, mainly through phishing scams, has access to private information which can further be used for malicious purposes, including but not limited to identity theft. Surprisingly, security cameras account for only 5% of IoT devices but their issues are at a 33% mark. As of recent, there has been a surge in decentralized peer-to-peer C2 communications through which the hackers can communicate through a local network and block out outside connections and can even operate without an internet connection. Cryptojacking malware has been on the rise, as well [1]. The dataset that we will be using is the IEEE published IoT network intrusion dataset. The dataset consists of 42 raw network packet files and "All attacks except Mirai Botnet category are the packets captured while simulating attacks using tools such as Nmap. The case of Mirai Botnet category, the attack packets were generated on a laptop and then manipulated to make it appear as if it originated from the IoT device"[2].

There has been a lot of research done in this area, and our goal is to use the previous research and bring about a combined solution to prevent the above-mentioned problems. As of now, our main focus is to use active monitoring, RPI firewalls, AWS services, as well as patching devices.

Article Title Footnote needs to be captured as Title Note
[†]Author Footnote to be captured as Author Note

WOODSTOCK'18, June, 2018, El Paso, Texas USA
© 2018 Copyright held by the owner/author(s). 978-1-4503-0000-0/18/06...$15.00
Journals and Proceedings).

KEYWORDS
Cybersecurity, IP and TCP protocol, DDoS attacks, Hacking Hash, Firewall, RaspberryPi Firewall, Network Security, Mirai Botnet, Wireframe, IDS, IPS, IP spoofing

1. Introduction: Device and Network Security

Every device we own has a unique ID associated with it, which serves as its identifier in the world wide web, and this is known as the internet protocol (IP). A protocol is a means through which devices communicate effectively; basically it is a set of rules for communication through the internet. The TCP protocol, which stands for Transmission Control Protocol is a standard medium used for exchanging data. The basic unit of transmission through TCP is via the means of a packet. And "Each connection must always be identified by two clearly defined endpoints (client and server). It doesn’t matter which side assumes the client role and which assumes the server role. All that matters is that the TCP software is provided with a unique, **ordered pair** consisting of IP address and port (also referred to as "2-tuple" or "socket") for each endpoint” [3]. Figure 1 explains the client and server connection when it comes to the TCP which is also known as a reliable protocol.

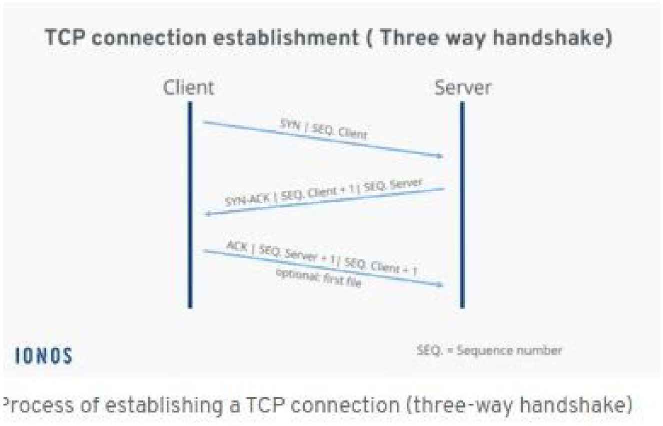

TCP connection establishment (Three way handshake)
Client
Server
SYN | SEQ. Client
SYN-ACK | SEQ. Client + 1 | SEQ. Server
ACK | SEQ. Server + 1 | SEQ. Client + 1
optional: first file
IONOS
SEQ. = Sequence number
ᵓrocess of establishing a TCP connection (three-way handshake)

Figure 1.1: TCP connection

Expanding on this topic we move on to firewalls. A firewall, in simple terminology, serves the purpose of a door in a house; i.e. it only allows people you know and trust into your house. The same thing applies to network and device firewalls. Here we will be focusing on a network firewall, which serves to prevent unauthorized access and also monitors the network traffic data (incoming as well as outgoing). It blocks access based on a defined set of rules. Another term for firewall is that it is known as an intrusion detection mechanism. However, this technology exists with its downsides, as well. Firewall cannot stop attacks even though it can stop traffic from getting through. This brings into picture the IDS (Intrusion detection system) which tracks all the inbound and outbound traffic in the system. IDS would notify the computer/host of an attack or intrusion in the network. The IDS “also watches for attacks that originate from within a system. This is traditionally achieved by examining network communications, identifying heuristics and patterns (often known as signatures) of common computer attacks and taking action to alert operators” [4]. An IPS (Intrusion Prevention System) is a detective as well as preventative system, whereas an IDS is a database of such systems, i.e, not allowing the traffic to access the target network for attack. With IoT devices being connected to this network there are many types of attacks that can take place, as detected in the IoT IDS data. Using this data and making use of Regression we will determine the trend in attacks vs number of devices. According to our initial hypothesis- based on the assumption that as more IoT devices are connected to a network the more vulnerable it is to different types of network attacks described in the next section and hence putting user data at risk. Our solution makes use of a cost-effective approach of having a Raspberry-Pi Firewall for each IoT device in the network. This will provide defense against an IDS attack as it will configure the firewall to single out the IP address of an intruder.

1. NETWORK ATTACK AND THEIR TYPES

As mentioned earlier, a network is susceptible to many forms of attacks, on this paper we will discuss the two most known attacks that serve as the motivation to come up with an RPI firewall solution. The primary goal of a network attack is to gain unauthorized access for either malicious activity or stealing data.

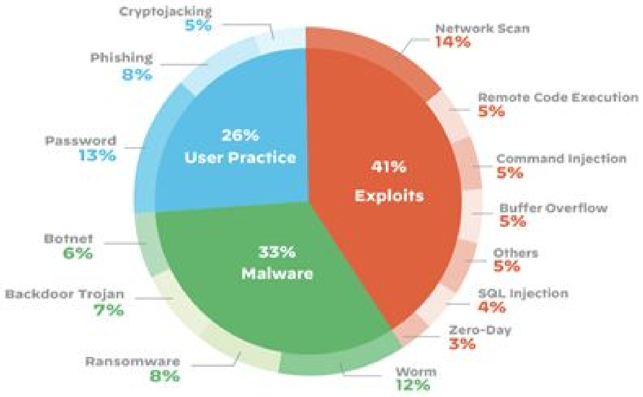
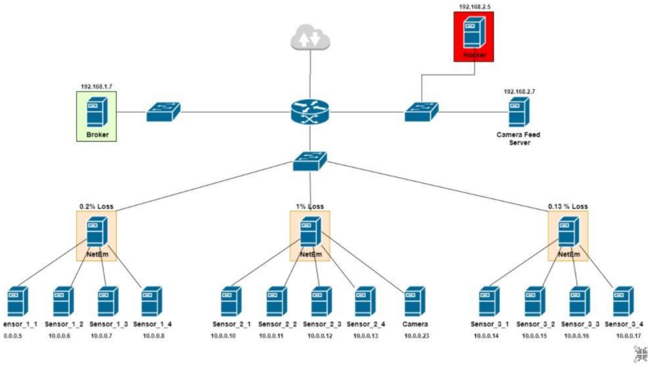


Figure 1.2: Breakdown of top IOT threats

As seen from the graph above three of the most known threats are User Practice, exploits, and malware. Malware gains entry over the internet and takes control of all the systems connected to that network. IP spoofing uses a fake IP address to create the illusion of a legitimate user. [5]. DoS which stands for Distributed Denial of Service, is an attack that fully destroys the network and makes it unavailable for the users. A DDoS which adds to the prefix distributed to DoS is a more complex version of an attack. There are many types of attacks possible, and some are harder to detect than the others. The DDoS attack was used in 2012, when almost all of the internet was inaccessible on the U.S. east coast. Known as the Mirai Botnet attack, it turned out to be ever more powerful than the creators intended. Botnet, in this case, refers to a group of computers connected over the internet [6]. And are under control of a third party. and “Because there are many bots, the controllers basically have access to a sort of a hacked-together supercomputer that they can use for nefarious purposes”[6]. Home PCs serve as the primary spot for the botnet creation, once the PC is composed of the bot herder (third party) issues commands. In 2017, 8.4 billion IoT devices were in use. Mirai “Rather than attempting to use complex wizardry to track down IoT gadgets, it scanned big blocks of the internet for open Telnet ports, then attempted to log in using 61 username/password combos that are frequently used as the default for these devices and never changed. In this way, it was able to amass an army of compromised closed-circuit TV cameras and routers, ready to do its bidding” [6]. Started by an undergraduate student at Rutgers, who tested his DDoS attacks on the university’s students, it was named Mirai which means “the future” in Japanese. Even though the people in the making were caught, the Mirai Botnet code was made public, and people were building on it to make it stronger to detect - which implies that they can use their luck to infect IoT devices which are still unprotected. The Mirai attack was so significant in the sense that “There are certain IP address ranges that Mirai is hard-wired to avoid, including those owned by GE, Hewlett-Packard, and the U.S. Department of Defense” [6].

3. APPROACH: DATA AND AI-BASED SOLUTION

In order to show weaker security among IoT devices we have made use of publically available datasets and made use of the MAC addresses and the attack information to show how common DDoS attacks are in the case of IoT. Figure 1.1 reveals how the systems are compromised in the case of a network attack with the red box representing the attacker and the lowest tier boxes representing IoT sensors/ devices

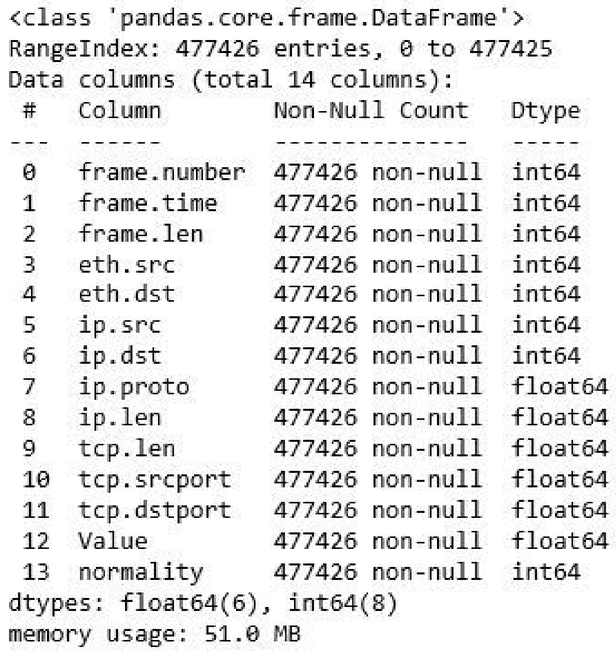


Algorithms provide effective solutions to security issues by encrypting better hashing technologies and such. In this case, we will make use of the Bat algorithm for improved security.

4. ML based approach: Classification Algorithms

DATASET 1: IoT Network Intrusion

In this case, we used Network Intrusions dataset not just through the most common means DDoS but also through other means such as man in the middle, wrong setup, scan attack and data probing. The data features have been described in the image. We extracted the normality column as that has been pre-processed to encode and account for the network attacks.

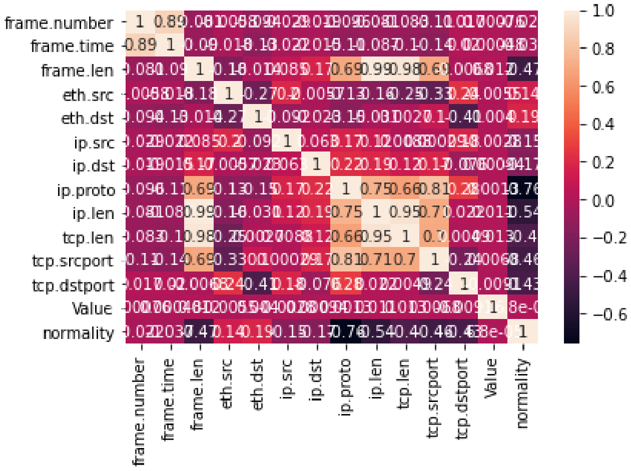


```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 477426 entries, 0 to 477425
Data columns (total 14 columns):
 #   Column        Non-Null Count   Dtype
---  ------        --------------   -----
 0   frame.number  477426 non-null  int64
 1   frame.time    477426 non-null  int64
 2   frame.len     477426 non-null  int64
 3   eth.src       477426 non-null  int64
 4   eth.dst       477426 non-null  int64
 5   ip.src        477426 non-null  int64
 6   ip.dst        477426 non-null  int64
 7   ip.proto      477426 non-null  float64
 8   ip.len        477426 non-null  float64
 9   tcp.len       477426 non-null  float64
 10  tcp.srcport   477426 non-null  float64
 11  tcp.dstport   477426 non-null  float64
 12  Value         477426 non-null  float64
 13  normality     477426 non-null  int64
dtypes: float64(6), int64(8)
memory usage: 51.0 MB
```

Man in the middle is “ a general term for when a perpetrator positions himself in a conversation between a user and an application—either to eavesdrop or to impersonate one of the parties, making it appear as if a normal exchange of information is underway.The goal of an attack is to steal personal information, such as login credentials, account details and credit card numbers. Targets are typically the users of financial applications, SaaS businesses, e-commerce sites and other websites where logging in is required[10]”. Scan attacks are done through ports and “to launch a port scan attack, hackers take advantage of a tool like Nmap to sort through the available hosts on your network. A port scan will return one of three potential classifications for identified ports:

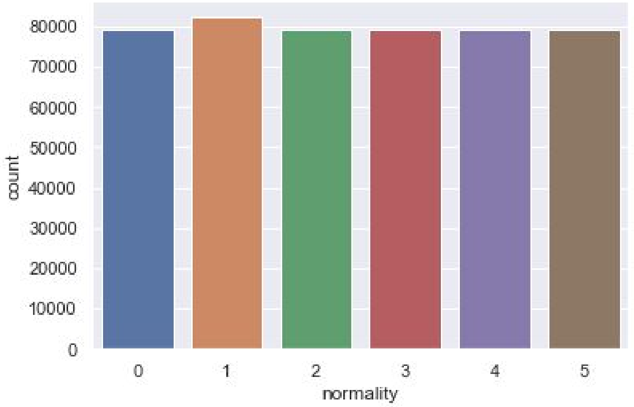
- Open: Target host is listening on the port and the service used in the scan is being used.
- Closed: Packet requests are received but the service isn’t listening on the port.
- Filtered: Packet request is sent but there’s no reply, indicating a firewall has filtered the request packet.

Mapping ports in this way gives attackers insight into the weak points of your network. Every open port indicates the potential for a vulnerable system that attackers can exploit to gain a foothold in your network or launch denial of service. campaigns



The correlation matrix for this dataset was plotted as well. The correlation matrix shows the level of relationship between the variables/attributes in the dataset and how strong it is. The plot for the normality is the IoT values encoded from 0-5, the bar graph plot was obtained using the seaborn library. The encoding is as follows:

Attack/Error	Encoding
0	Normal
1	Wrong Setup
2	DDoS
4	Scan Attack
5	Man in the middle



Machine learning has been the brain behind the emergence of IoT, so we decided to use a classification algorithm that would determine the kind of attack being done and would in turn lead to the development of a better IDS system. K-Nearest Neighbors is an algorithm that essentially categorizes new data based on existing data, and follows a semi-supervised approach with its learning. To train this classifier the scikit learn machine learning model was used. In order for the data to fit a semi-supervised form of learning it has to have training data as well as test data. The ML approach for first cleaning and preprocessing the data was used, the category column was mapped to the number 1 using one-hot encoding. The model was then split into x and y containing only the category of attack column. The test size was taken to be 20%; usually the training and test data have either a 8:2 or 7:3 ratio. The MinScaler function was used from the sklearn library to fit the data to be used by the classifier.

The decision tree classifier was used; a decision tree is a classification algorithm used typically for "classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. A tree can be seen as a piecewise constant approximation"[12]. The graph for the decision tree classifier as well the code has been pasted below. The primary reason for using a decision tree was that it is able to handle a multi-output problem. For example, if an intruder was trying to perform a man in the middle attack while another intruder was performing a scan attack, a decision tree classifier would be able to detect both and classify it. The other advantages of using a decision tree are: "

- Requires little data preparation. Other techniques often require data normalisation, dummy variables need to be created and blank values to be removed. Note however that this module does not support missing values.
- The cost of using the tree (i.e., predicting data) is logarithmic in the number of data points used to train the tree.
- Able to handle both numerical and categorical data" (Reference).

The accuracy of the algorithm came out be 1.0/1.0 and the confusion matrix as well as the classification report has been shown in Figure 6

Decision Tree Code:

```
scaler_fun = MinMaxScaler()
scaler_fun.fit(x_train)
x_train = scaler_fun.transform(x_train)
x_test = scaler_fun.transform(x_test)
from sklearn.tree import DecisionTreeClassifier
sc=[]
for T in range(1,30):
  DecisionTree=DecisionTreeClassifier(max_depth = T)
  DecisionTree.fit(x_train, y_train)
  sc.append(DecisionTree.score(x_test,y_test))
plt.plot(range(1,30),sc, color='black')
plt.show()
```

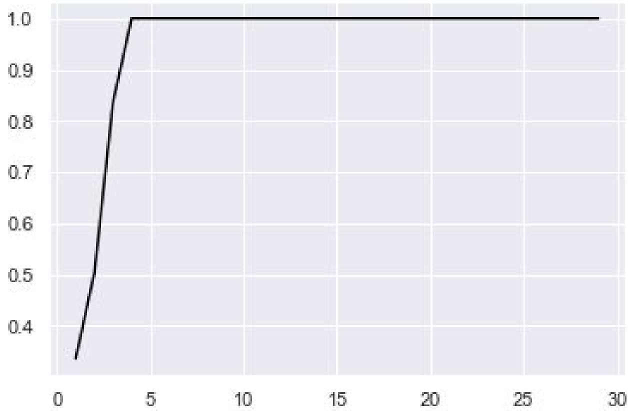


Fig 6: Decision Tree Classifier

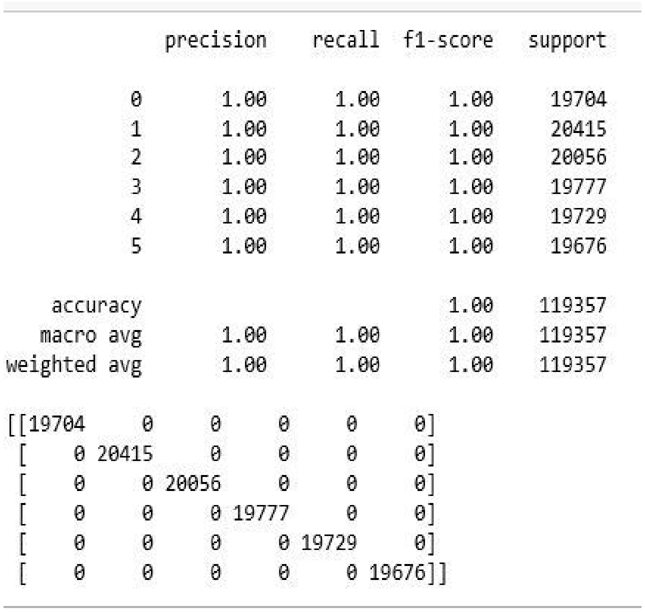
The accuracy score for the decision classifier as well as the classification report came out to be:

CODE:

```
DecisionTree=DecisionTreeClassifier(max_depth=10)
DecisionTree.fit(x_train, y_train)
print("The score for Decisison Tree Classifier is:", DecisionTree.score(x_test,y_test))
pred = DecisionTree.predict(x_test)
print(classification_report(y_test,pred))
print(confusion_matrix(y_test,pred))
```

RESULT:

```
The score for Decisison Tree
Classifier is: 1.0
```



```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     19704
           1       1.00      1.00      1.00     20415
           2       1.00      1.00      1.00     20056
           3       1.00      1.00      1.00     19777
           4       1.00      1.00      1.00     19729
           5       1.00      1.00      1.00     19676

    accuracy                           1.00    119357
   macro avg       1.00      1.00      1.00    119357
weighted avg       1.00      1.00      1.00    119357

[[19704     0     0     0     0     0]
 [    0 20415     0     0     0     0]
 [    0     0 20056     0     0     0]
 [    0     0     0 19777     0     0]
 [    0     0     0     0 19729     0]
 [    0     0     0     0     0 19676]]
```

Figure 7: Confusion Matrix (Decision Tree)

The confusion matrix:

In the above figure we see that the confusion matrix was run with an accuracy of a 1/1. As you can see here in figure 7, it creates a diagonal matrix.

Random Forest Classifier:

Similar to the decision tree, the "Random forest, like its name implies, consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model's prediction." The image below shows the structure of this algorithm.

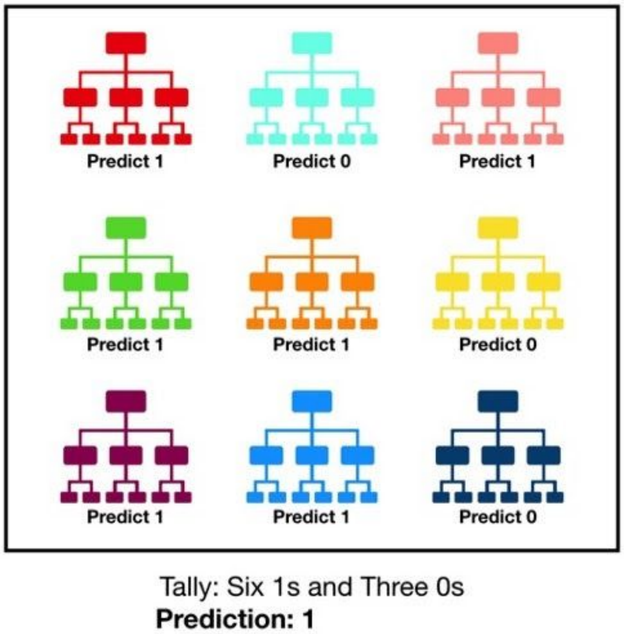


Figure 8: Random Forest Classifier

Extrapolating from the decision tree algorithm - we expected a similar result with random forest as it is a sub algorithm of decision tree classifier.

CODE:

```
sc=[]
for i in (np.arange(110,1000,110)):
  classifier_RF =
RandomForestClassifier(n_estimators =i,
max_depth=10, random_state=0)
  classifier_RF.fit(x_train, y_train)
  sc.append(classifier_RF.score(x_test,y_test))
plt.plot(np.arange(110,1000,110),sc)
plt.show()
```

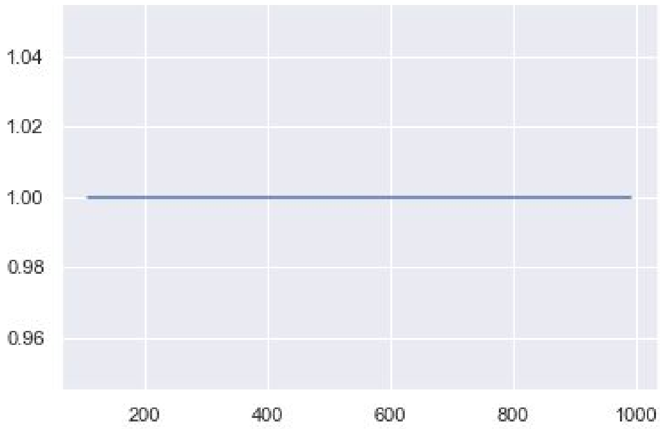
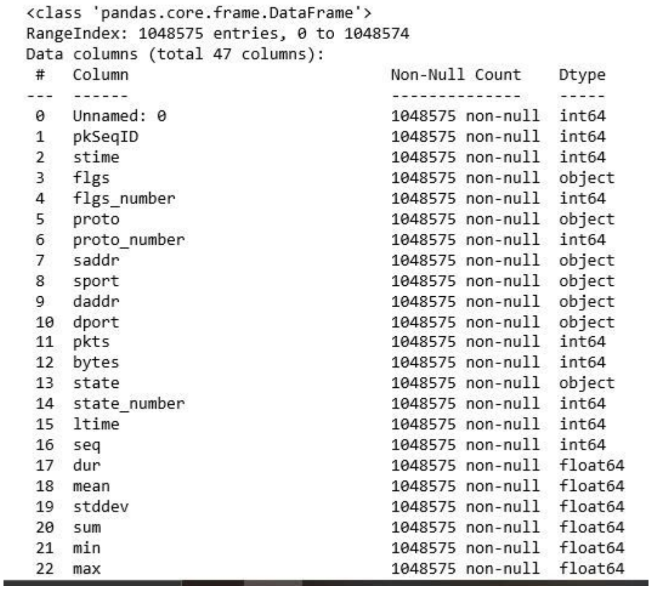


Figure 9: Decision Tree Classifier

The accuracy as well as the classification result reports were the same as the last one.

DATASET 2: DDoS Attacks

The DDoS attack dataset obtained from kaggle was used to analyze the DDoS attacks. The dataset description has been shown in the figure below



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1048575 entries, 0 to 1048574
Data columns (total 47 columns):
 #   Column                            Non-Null Count    Dtype  
---  ------                            --------------    -----  
 0   Unnamed: 0                        1048575 non-null  int64  
 1   pkSeqID                           1048575 non-null  int64  
 2   stime                             1048575 non-null  int64  
 3   flgs                              1048575 non-null  object 
 4   flgs_number                       1048575 non-null  int64  
 5   proto                             1048575 non-null  object 
 6   proto_number                      1048575 non-null  int64  
 7   saddr                             1048575 non-null  object 
 8   sport                             1048575 non-null  object 
 9   daddr                             1048575 non-null  object 
 10  dport                             1048575 non-null  object 
 11  pkts                              1048575 non-null  int64  
 12  bytes                             1048575 non-null  int64  
 13  state                             1048575 non-null  object 
 14  state_number                      1048575 non-null  int64  
 15  ltime                             1048575 non-null  int64  
 16  seq                               1048575 non-null  int64  
 17  dur                               1048575 non-null  float64
 18  mean                              1048575 non-null  float64
 19  stddev                            1048575 non-null  float64
 20  sum                               1048575 non-null  float64
 21  min                               1048575 non-null  float64
 22  max                               1048575 non-null  float64
```

Figure 11: Dataset Decscription

The correlation matrix was plotted using the seaborn library: The first image is with the labels and the second image is with the label numbers to provide a contrast as to how correlation works:

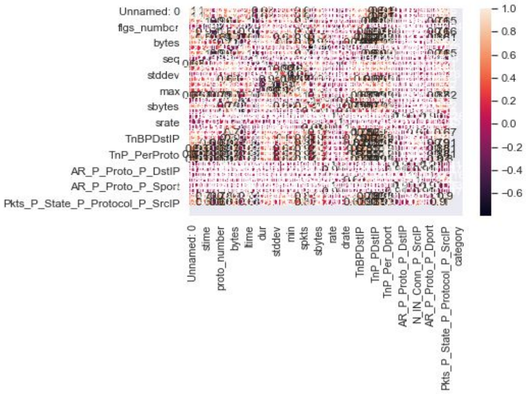


Figure 11: Correlation matrix with labels

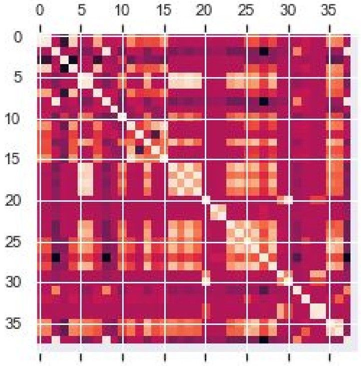


Figure 12: Correlation matrix without labels

The seaborn library was used to plot the number and kinds of attacks in the dataset. The DDoS attacks accounted for 1*10^6 attacks, which then goes on to further strengthen our motivation for this paper.

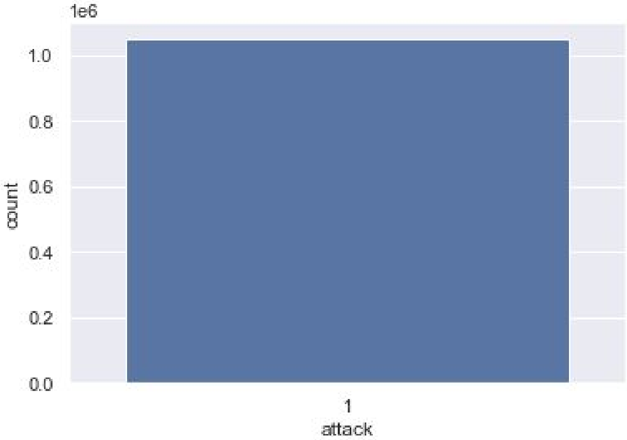


Figure 13: DDoS attack counts

The KNearestNeighbors Algorithm was applied which essentially created clusters based on the training data and then used the test data to categorize the new data. The higher the value of K the higher the more efficient the algorithm tends to be more efficient. However due to the inconsistencies in the dataset despite pre-processing and converting the values, the algorithm did not provide us with a graph to generate the confusion matrix as well as accuracy results. We even tried implementing the decision tree algorithm which basically follows a flow-chart and tree like structure in order to compare the accuracies; however, with the approach being right our dataset did not quite fit the category of being a good one as IoT network attack datasets are sparse in nature. The algorithm for KNN has been listed below in case a better dataset fits the case. Lastly, we also implemented the RandomForestClassifier with no avail as the dataset type was mismatched with the algorithm and even after multiple fixes the result was unfortunately blurred.

CODE:

```
accu=[]
for k in range(1,100):
  classifier_knn= KNeighborsClassifier(n_neighbors = k)
  classifier_knn.fit(x_train, y_train)
  accu.append(classifier_knn.score(x_test, y_test))

knn_arr=list(range(100))

plt.plot(knn_arr,accu, color='red')
plt.show()
```

5. RaspberryPi Firewall (RPI)

A RaspberryPi is a credit-card sized computer that costs about $35 dollars and is used most commonly by engineering enthusiasts to build projects of all sorts ranging from programming to creating a drone to hosting a minecraft server or twitter bot. The most eye-catching aspect of this is the cost of it. IoT security can be strengthened by just configuring a firewall using the RaspberryPi for the network to which the IoT devices are connected, this then creates a divide and conquer based approach to filter out IP addresses in the network. This approach seems very simple, but it is in fact quite effective given the cost associated and seething up an RPI firewall is something that can even be done by a non-programmer. The RPI firewall would serve as the gateway of communication to the cloud server. RPI offers various other tools that can be further used to strengthen the security of the network as well as protecting the IoT device.

The advantages offered by it are as follows.

- Enforce network traffic policies
- Ensure that abnormal packets does not get out or in the network
- DHCP server to distribute network parameters to LAN
- DNS cache/server to speed up DNS requests and filter out bad DNS queries
- NIDS to detect malicious traffic, such as malware or vulnerability exploits
- Central network monitoring node to watch and debug network traffic” [5]

Setting up the firewall allows the administrator direct access as to what the rules of the network are and this in turn leads to better collection of traffic to in-turn integrate an IDS system based on the most common type of traffic being stopped.

Given below is the network configuration code after the RPI has been set up with the OS and others tools required.

```
# vi /etc/rc.conf
#
---------------------------------------------------------------------
# NETWORKING
#
---------------------------------------------------------------------
# HOSTNAME: Hostname of machine. Should also be put in /etc/hosts
#
HOSTNAME="RSS"
# Static IP
interface=eth0
address=192.168.1.3
netmask=255.255.255.0
broadcast=192.168.1.255
gateway=192.168.1.1
```

```
# Disable DHCP by commenting these lines or else it will override the static IP configuration
# interface=eth0
# address=
# netmask=
# gateway=
```

Remove from startup the unneeded daemons :

```
DAEMONS=(!hwclock syslog-ng network @crond @sshd @openntpd)
```

```
# vi /etc/hosts
192.168.1.3 PiWall
127.0.0.1 localhost.localdomain localhost
::1 localhost.localdomain localhost

# rc.d restart network
```

This code is then followed by a series of another steps for various settings such as adding another admin etc.

6. PAST WORKS AND EXPERIENCES

Given the rise of Internet of Things network / device consumerism, hackers and individuals with malicious intent have their fair share of attacks within the IoT realm. A 2019 Gartner report estimates that 5.8 billion IoT devices will be implemented in the commercial sector by the end of 2020 - a 20.83% increase from 2019's 4.8 endpoints.

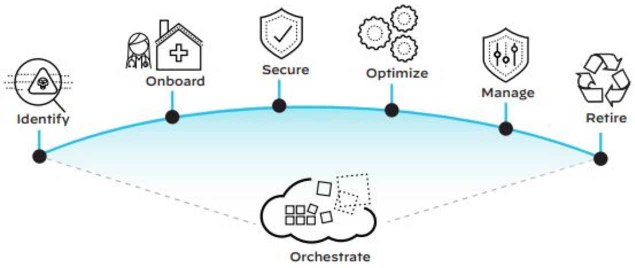
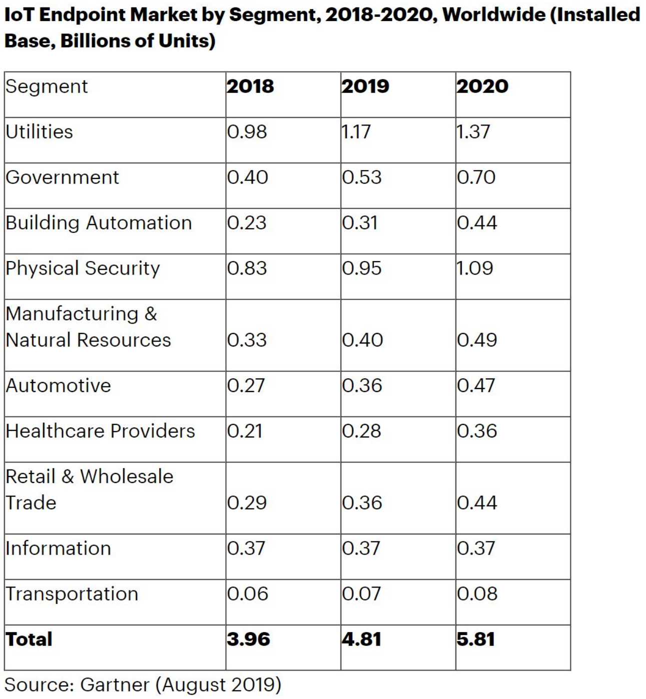


Figure 14: Ideal IoT device life cycle [4]



IoT Endpoint Market by Segment, 2018-2020, Worldwide (Installed Base, Billions of Units)

Segment	**2018**	**2019**	**2020**
Utilities	0.98	1.17	1.37
Government	0.40	0.53	0.70
Building Automation	0.23	0.31	0.44
Physical Security	0.83	0.95	1.09
Manufacturing & Natural Resources	0.33	0.40	0.49
Automotive	0.27	0.36	0.47
Healthcare Providers	0.21	0.28	0.36
Retail & Wholesale Trade	0.29	0.36	0.44
Information	0.37	0.37	0.37
Transportation	0.06	0.07	0.08
Total	**3.96**	**4.81**	**5.81**

Source: Gartner (August 2019)

Table 1: IoT Market Endpoint Market by Segment, 2018-2020 [5]

For our research, we initially investigated a multitude of IoT security reports within the past four years - more specifically, an IoT threat report from Unit 42 of Paloalto Networks. The overall concerns communicated within these reports consist of the following: compromised authentication, lack of session encryption, outdated software, and the lack of connection between IT and IoT. A survey conducted by 451 Research with IT professionals has led to the conclusion that IoT security is not a lighthearted topic: 55% of participants label it as a high priority when integrating IoT devices into their systems. As summarized by many, security professionals seek the following practices when securing their IoT platform:

- Automatic antivirus and firmware updates when published to the system
- Constant monitoring on all attached devices
- Stronger authentication credentials (on user's end)
- Ease of access and integration for security protocols
- Proof of defense against various attacks, not being limited to:
  - Denial of service
  - Penetration
- Secure communication between devices within the network

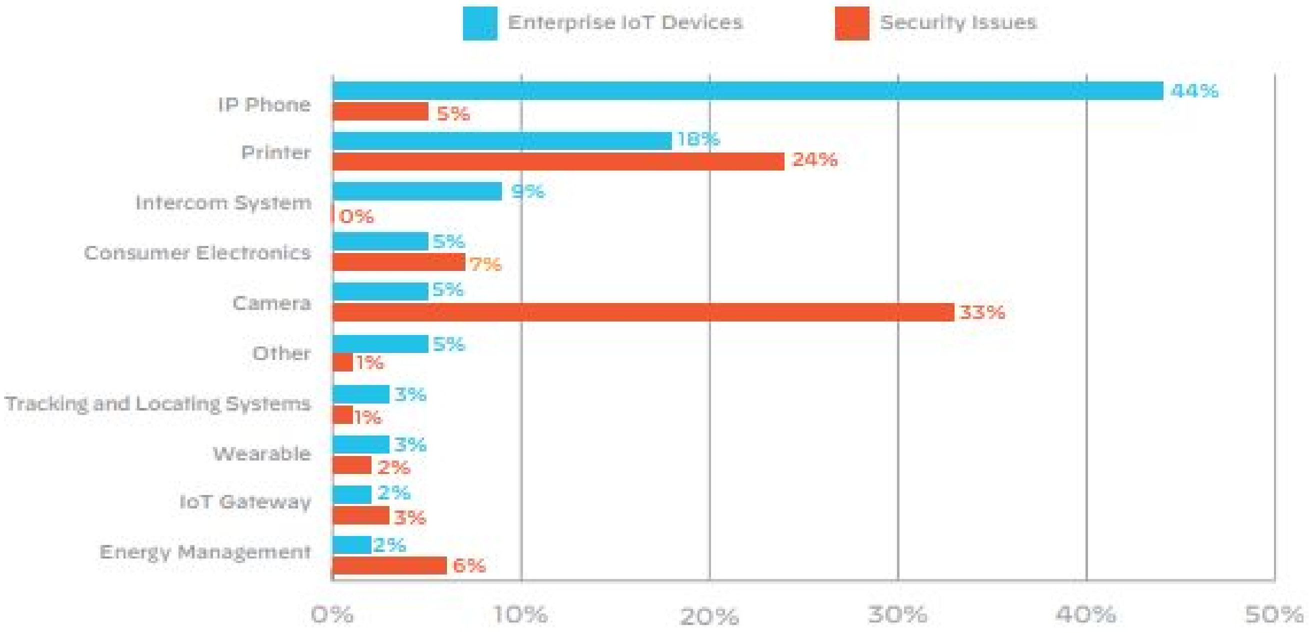


Figure 15: Percentage of certain IoT devices in enterprise versus percentage of total security issues [7]

In order to address the proposed solutions listed above, more concerned individuals need to take a deep dive at the root problems of IoT security as a whole. Unit 42's more critical commercial-level claims consist of only 2% of IoT endpoint traffic being encrypted, 57% of commercial level devices being vulnerable to medium-severity attacks and above, and 83% of medical-related IoT devices running on unsupported networks - due to the finished "lifeline" of Windows 7 support in January 2020. Mainly, these claims are backed by three reasons: [7]

- Information technology's (IT) lack of fine-tuned IoT device recognition
    - Identifying a device's IP address and operating system is only the basis
    - Most technicians do not bother to recognize the device's specifications:
        - Network access requirements, specifications for employment, and security protocols
- Modern day security system's lack of compatibility with IoT devices
    - Most IoT devices are using custom and/or outdated operating systems
    - Such security systems would end up labeling these devices as anomalies
- The lack of connection between information technology and operational technology (OT) teams and practices
    - As of currently, there is not much incentive to "combine" the knowledge of both teams in order to increase security on non-IT devices
        - The incentive being lack of acknowledgement
    - Most OTs are in charge of maintaining devices that have a high chance of running an unsupported operating system - in which ITs are usually responsible for maintaining

As of currently, a multitude of companies offer an array of "solutions" to consumers - all in the form of their own developed software. Ranging from cloud-based implementations from Paloalto Network to more specific streamlines from nCipher, these programs aim to offer network security and transparency to system administrators through the implementation of predefined security protocols.

In pursuit of a more "flushed-out" solution towards IoT security, researchers from Texas A&M, California State University, and Jordan University of Science and Technology collaborated in an effort to develop a new IoT model that integrates a stronger sense of security at its core.

Their proposed model includes components that strengthen the privacy and security of the system, while also implementing an in depth layer identification system. During their study, the team emulated the layers of the network with the help of Amazon Web Services as their bottom layer, a Raspberry Pi 4 with Greengrass Edge Environment as the middle layer, and AWS's IoT environment as the top layer - the cloud. [8]

Another stride taken to investigate the current state of IoT security is with a conducted study between researchers from Kyushu University and Universitas Gadjah Mada. This specific paper brings awareness to the rise of cyber-attacks on IoT devices and networks - and credits the lack of extra computational resources and memory within these devices for a good portion of their unoptimized security. In response, these researchers developed and tested a machine learning-based botnet attack detection framework - which differs from existing intrusion detection systems that are not fully capable of detecting all forms of attack. Through the implementation of machine learning, the proposed framework can be trained to identify almost all attack types and their variances. To avoid drawing too much power from devices with limited computational power, their system is integrated with a feature selection system. [8]

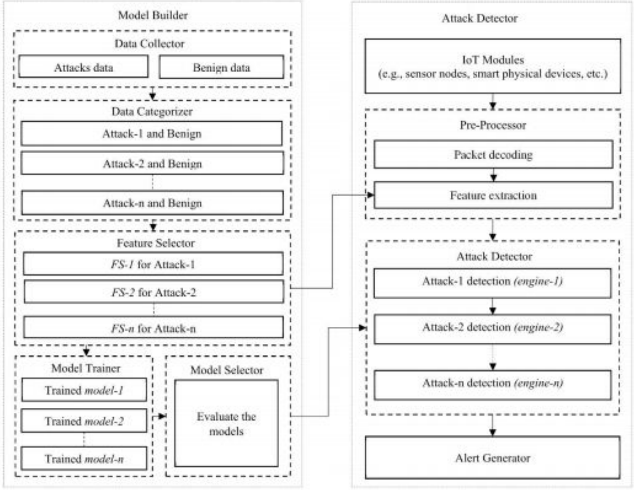
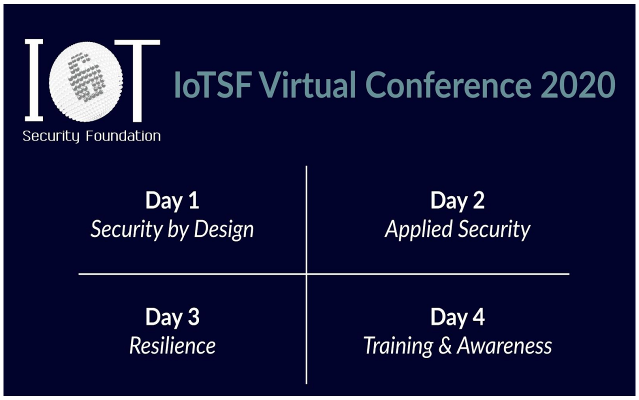


Figure : Identification architecture of proposed ML-based identification framework [8]

Stepping away from profit-based solutions, one major step towards improving IoT security is with the collaborative efforts of the IoT Security Foundation. Founded in 2015, technological and security experts gathered at Bletchley Park, England, to discuss the emerging dangers behind IoT security issues. United under the agreement that this form of security is not within the same realm as PC and mobile devices, the foundation aims to provide the necessary resources and influence for the IoT sector to become more comfortable with adoption of more secure protocols - while increasing the overall popularity of IoT to an international scale. [9]



Ranging from training workshops to international conferences, the Security Foundation has established a basis in which to train individuals interested in developing IoT devices in a safe and efficient fashion. One of the major goals behind the foundation is to construct a Compliance Framework, in which developers and consumers can utilize these "recommended steps" when constructing IoT devices and services. Given the fact that a large portion of developers find the services mentioned above to be a must in their security implementations, they can serve as a baseline to the foundation's framework.

7. THE FUTURE OF IOT SECURITY

Despite the flaws within current IoT devices, developers are taking major strides to strengthening their security.

The pursuit for more secure IoT devices and networks will be endless - as long their defenses seem permeable. Ultimately, as long as computational and IoT devices continue to evolve, the different attacks that can be implemented on them will follow the same path. History has shown that convenience comes at a cost if not implemented with care.

Neural Networks:

ACKNOWLEDGMENTS

We would like to thank our Professor Jorge Ortiz for this opportunity to research our topic and provide a potential solution to the aforementioned problem. Our extended thanks to all the authors mentioned in the references.

REFERENCES

[1]https://www.kaggle.com/speedwall10/iot-device-network-logs

[2]: https://www.kaggle.com/rainbowgirl/clustering-categorical-peoples-interests/notebooks https://seaborn.pydata.org/

[3]https://www.datasciencecentral.com/profiles/blogs/comparing-classifiers-decision-trees-knn-naive-bayes

[4]https://start.paloaltonetworks.com/unit-42 -iot -threat - report?utm_source=google -search&utm_medium=paid - search&utm_term=iot%20security&utm_campaign=INS -Americas -EN -Search - Lead_Gen -US - Key_Terms&utm_content=472456088048&utm_network=&sfdcid=7010g0000 01JJOZAA4&gclid=Cj0KCQjwit_8BRCoARIsAIx3Rj4fiqOAXFKn4jKgkkQi SFFsXIcLfjzS8Za_gOi5n1kUOx00VjlkiugaAu1PEALw_wcB

[5]https://ieee-dataport.org/open-access/iot-network-intrusion-dataset

[6]

[7]https://www.mdpi.com/2076-3417/10/12/4102?type=check_update&version=1

[8]https://www.mdpi.com/1424-8220/20/16/4372?type=check_update&version=1

[9]https://www.iotsecurityfoundation.org/

[10]https://blog.niagaranetworks.com/blog/port-scan-attack

[11]https://scikit-learn.org/stable/modules/tree.html

[12]https://towardsdatascience.com/understanding-random-forest-58381e0602d

[13]https://www.imperva.com/learn/application-security/man-in-the-middle-attack-mitm/

Figure 1.2: https://unit42.paloaltonetworks.com/iot-threat-report-2020/

[2] https://ieee-dataport.org/open-access/iot-network-intrusion-datase