Data Analysis for Anomaly Detection in IoT Honeypot

Muhaimin Omar (1004914)

Outline

- Introduction
- Background
- Scope of Work
- Implementation
- Result and Evaluation
- Conclusion
- Future Works

Outline

- Introduction
- Background
- Scope of Work
- Implementation
- Result and Evaluation
- Conclusion

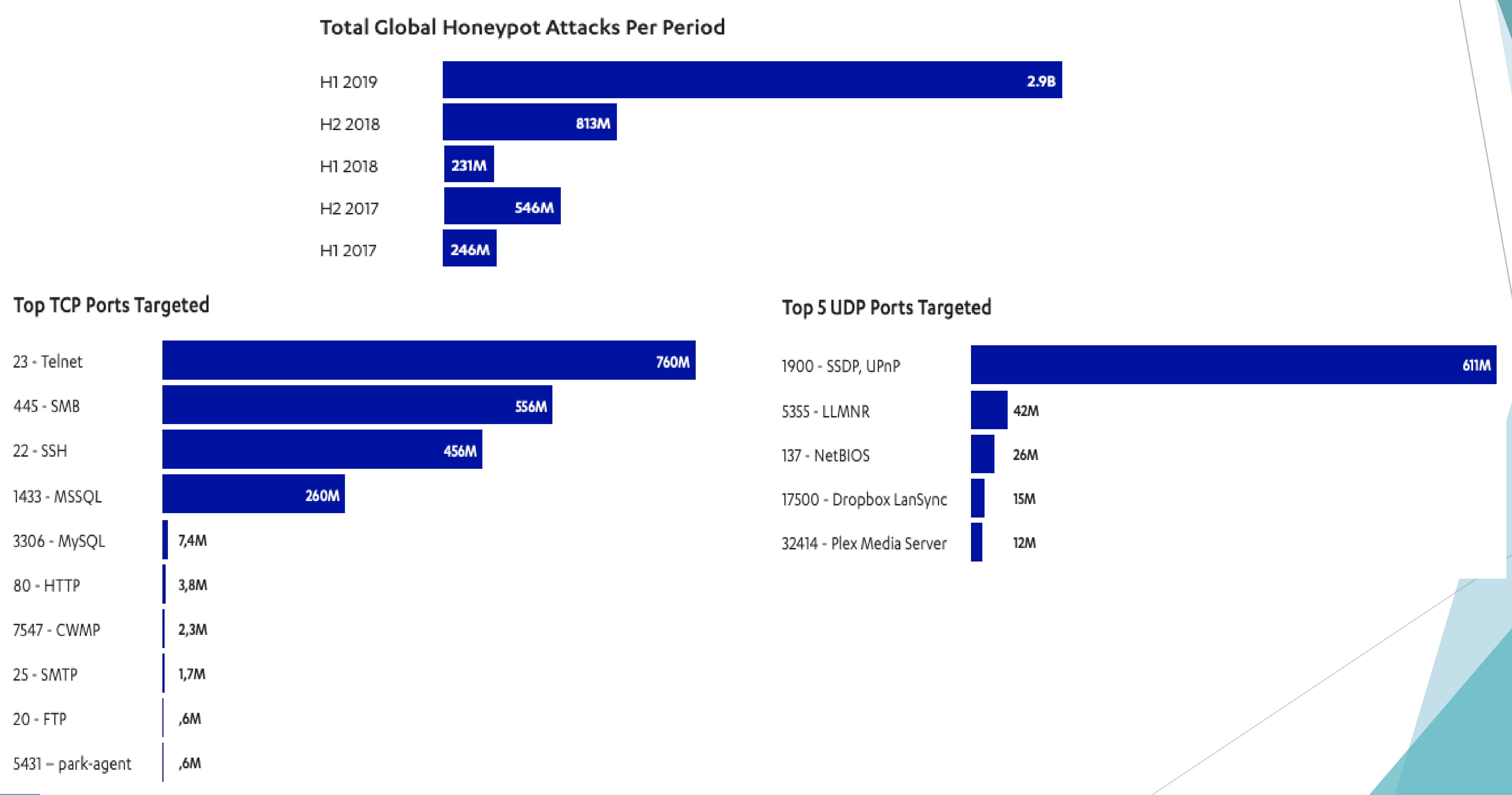
Introduction

Internet of Things (IoT)

- The **Internet of Things (IoT)** will reach an installed base of **more than 80 billion units in the next 3 years**, an **increase from 35 billion reported in 2020.**
- The **growth of around 130%** has been acclaimed as a revolution in the way that **society and organizations will function**.
- Problems like **data ownership, governance, and security** are posing **new challenges**.

Source: Juniper Research, “IOT - THE INTERNET OF TRANSFORMATION 2020”, 2020.

Cyber-attacks on IOT devices



Source: F-Secure, “Attack Landscape H1 2019”, 2019, https://blog-assets.f-secure.com/wp-content/uploads/2019/09/12093807/2019_attack_landscape_report.pdf

Honeypots

- It is a **sacrificial computer system** that, like a decoy, is designed to **attract cyberattacks.**
- It **imitates a hacking target** and **leverages infiltration attempts** to **gather information about cybercriminals and their methods of operation**, or to divert them from other targets.

Data Analysis in Cybersecurity

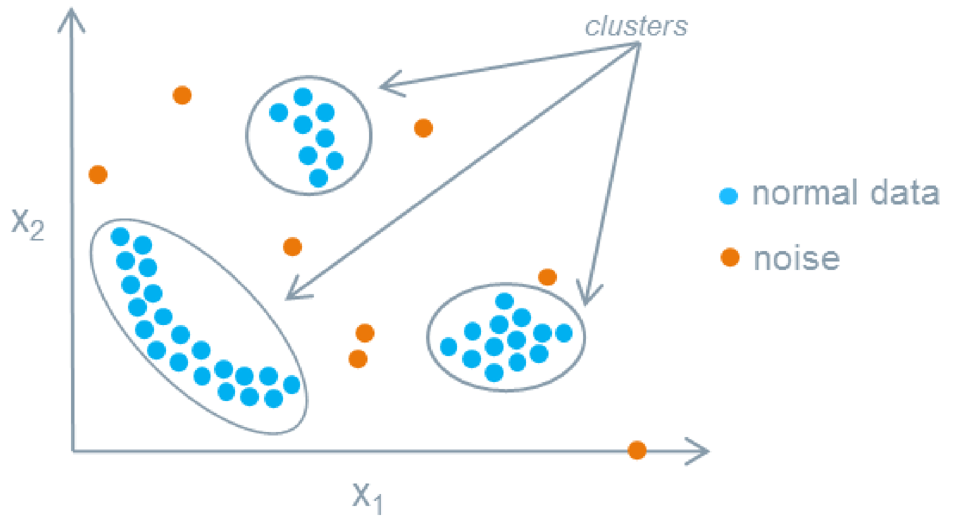
- Data analysis **helps in detecting vulnerabilities** that have arisen as a result of the exponential growth of technology and the Internet, as well as our growing reliance on both.
- By **alerting decision maker**s about potential fraud, strange network traffic patterns, hardware problems, and security breaches, data analysis **may provide a comprehensive view of internal and external dangers**.
- It **transforms data into useful information**, allowing firms to **evolve from a reactive to a proactive cybersecurity posture**.

Machine Learning in Cybersecurity

- Machine learning may be used to **check for network vulnerabilities and automate responses** in addition to early threat detection.
- Cybersecurity systems can use machine learning to **evaluate patterns** and learn from them in order to **help prevent repeated attacks** and respond to changing behavior.
- It can assist **cybersecurity teams in being more proactive** in preventing threats and responding to live attacks. It can help firms use their resources more strategically by reducing the amount of time spent on regular tasks.

Anomaly Detection

- Procedure that **detects the outliers**
- Anomalies could indicate unexpected network traffic, reveal a malfunctioning sensor, malicious behavior, or simply identify data that has to be cleaned before analysis.



Supervised & Unsupervised

- **Supervised Anomaly Detection**
  - Describes the data arrangement in which **the training and test data sets are properly labelled.**
  - Involves the **use of goal or outcome variables**. It searches future data for cases that are comparable to those in the past
- **Unsupervised Anomaly Detection**
  - The most **versatile arrangement which does not require any labels.** A distinction between a training and a test dataset is also not made.
  - Has **no target or result variable**, is more technically difficult than supervised learning and necessitates more subject-matter expert input.

Background

Background

- The value of using **honeypots to survey the security landscape** and **detect threats to IoT devices early** is **undeniable**.
- However, the **data generated** by these IoT honeypots **has been few and limited**, making it difficult to improve research into IoT security.
- A group of researchers developed a method for easily **integrating commercial off-the-shelf IoT devices into a honeypot architecture**. Using connections to commercial and private VPN services, the strategy projects a small number of heterogeneous IoT devices (that are **physically at one location**) as numerous (**geographically spread**) devices over the Internet.
- The intention was for those **devices to be discovered and attacked**, disclosing previously **unknown flaws**.

Background (cont’d)

- During the years 2017-2018, **network traffic was collected** by high-interaction IoT honeypots that were placed in the field for **1.5 years**.
- The honeypots are active in the wild, with **40 public IP addresses** directing traffic to **11 genuine IoT devices**.
- The dataset was extracted in JSON format using the Zeek tool from 258,871 PCAP files, yielding almost **81.5 million logs**.

Scope of Work

Scope of Work

- The goal is to **construct a data analysis workflow** that includes feature engineering **from honeypot network traffic** data and the application of appropriate **unsupervised machine learning technique** to better **understand the threat landscape** over the honeypot's operational period.
- The following tasks are expected:
  - Understanding of the IoT Honeypot setup
  - Feature extraction and engineering from the honeypot network traffic data
  - Formulation of suitable unsupervised machine learning techniques and identification of various attacks
  - Validation of the unsupervised machine learning model
- A software tool that implements the machine learning model is expected to be the final outcome from this research project.
- Only **HTTP logs** will be in scope of this research project.

Implementation

Implementation

- The following were used as part of the implementation:
  - Python programming language
  - Jupyter notebook
  - Associated libraries:
    - Pandas
    - Numpy
    - Seaborn
    - Matplotlib
    - Zat
    - Sklearn
    - Prettytable

Dataset Organization

- JSON logs were **split and compressed into 40 zip files**. To **extract only HTTP logs**, unzip.py, a simple python script, was created to unzip the zip files

unzip.py

```python
import zipfile
from pathlib import Path
p = Path('.')
for f in p.glob('*.zip'):
    with zipfile.ZipFile(f, 'r') as archive:
        archive.extractall(path=f'./{f.stem}')
        print(f'Done {f.stem}')
```

```
VPN-forwarded_Honeypots_Dataset — -zsh — 80×47
Last login: Sat Apr 17 16:37:35 on console
muhaiminomar@Muhaimins-Air ~ % cd /Volumes/Seagate\ Backup\ Plus\ Drive/VPN-forw
arded_Honeypots_Dataset/
muhaiminomar@Muhaimins-Air VPN-forwarded_Honeypots_Dataset % python3 unzip.py
Done 24-173.225.c128.a067
Done 25-173.225.f157.adf6
Done 26-173.225.49be.6699
Done 27-154.16.f0b8.45c9
Done 28-76.73.4c13.5a9b
Done 29-199.66.8472.9dee
Done 30-209.200.b859.8088
Done 31-209.200.b374.e4ae
Done 32-209.200.3167.3959
Done 33-188.227.d704.0b96
Done 34-188.227.f224.190d
Done 35-193.138.3180.23df
Done 36-173.225.0691.2493
Done 37-78.157.fd41.459a
Done 38-81.94.cf6c.28eb
Done 39-81.94.cfdf.2ece
Done 40-93.190.3d78.5e1f
Done 01-5.175.8717.aad2
Done 02-31.132.3c0b.bc11
Done 03-31.132.d063.8013
Done 04-46.246.fe1d.9416
Done 05-76.73.fa3a.5d9f
Done 06-92.240.6ddd.5c03
Done 07-213.184.2d21.9dd9
Done 08-154.16.513e.1a2a
Done 09-213.184.50de.192b
Done 10-173.225.d4bd.03e4
Done 11-209.200.ae9c.e04e
Done 12-213.184.ecbf.7293
Done 13-93.190.abcd.1234
Done 14-94.229.2f7c.94dd
Done 15-94.229.e0e9.f8e7
Done 16-77.78.541a.814d
Done 17-77.78.e513.a78b
Done 18-94.229.d716.62a3
Done 19-108.62.acb9.b7ae
Done 20-109.200.3d7c.3d77
Done 21-109.200.2806.636a
Done 22-154.16.4513.6c50
Done 23-173.225.1539.32ec
muhaiminomar@Muhaimins-Air VPN-forwarded_Honeypots_Dataset %
```

Dataset Organization

- Copyfiles.py, a simple python script, was created to **copy all http logs out from all 40 unzipped folders.**

copyfiles.py

```python
import glob
import os
import shutil

src = '.'
dest = r'/Volumes/Seagate Backup Plus Drive/VPN-forwarded_Honeypots_Dataset/HTTP_logs'

for file_path in glob.glob(os.path.join(src, '**', '*http*'), recursive=True):
    new_path = os.path.join(dest, os.path.basename(file_path))
    shutil.copy(file_path, new_path)
    print('Done', file_path)
```

HTTP_logs

Name	Date Modified	Size
109.200.2806.636a_00105_20180227224947-http.logreplaced.log	1 May 2021 at 2:48 PM	1.9 MB
5.175.8717.aad2_00320_20180502123150-http.logreplaced.log	1 May 2021 at 12:02 PM	1.6 MB
94.229.2f7c.94dd_00019_20180320041817-http.logreplaced.log	1 May 2021 at 1:44 PM	1.4 MB
94.229.2f7c.94dd_00016_20180320011817-http.logreplaced.log	1 May 2021 at 1:43 PM	1.2 MB
5.175.8717.aad2_00619_20171030022542-http.logreplaced.log	1 May 2021 at 12:03 PM	1.2 MB
5.175.8717.aad2_00618_20171030012542-http.logreplaced.log	1 May 2021 at 12:03 PM	1 MB
188.227.d704.0b96_00077_20170604151924-http.logreplaced.log	1 May 2021 at 4:06 PM	845 KB
213.184.2d21.9dd9_00023_20180109091612-http.logreplaced.log	1 May 2021 at 12:44 PM	823 KB
94.229.2f7c.94dd_00017_20180320021817-http.logreplaced.log	1 May 2021 at 1:44 PM	818 KB
94.229.2f7c.94dd_00020_20180320051817-http.logreplaced.log	1 May 2021 at 1:44 PM	815 KB
213.184.2d21.9dd9_00003_20171009115441-http.logreplaced.log	1 May 2021 at 12:43 PM	777 KB
188.227.d704.0b96_00117_20180930111754-http.logreplaced.log	1 May 2021 at 4:08 PM	771 KB
5.175.8717.aad2_01174_20171122052543-http.logreplaced.log	1 May 2021 at 12:04 PM	759 KB
213.184.2d21.9dd9_00344_20171121174458-http.logreplaced.log	1 May 2021 at 12:48 PM	758 KB
213.184.2d21.9dd9_00343_20171121164458-http.logreplaced.log	1 May 2021 at 12:48 PM	750 KB
94.229.2f7c.94dd_00018_20180320031817-http.logreplaced.log	1 May 2021 at 1:44 PM	742 KB
173.225.49be.6699_00053_20180711114731-http.logreplaced.log	1 May 2021 at 3:10 PM	720 KB
173.225.1539.32ec_00114_20180711042155-http.logreplaced.log	1 May 2021 at 2:57 PM	712 KB
81.94.cfdf.2ece_00310_20180502023158-http.logreplaced.log	1 May 2021 at 5:01 PM	710 KB
76.73.fa3a.5d9f_00157_20170621060914-http.logreplaced.log	1 May 2021 at 12:33 PM	699 KB
188.227.d704.0b96_00119_20180930131754-http.logreplaced.log	1 May 2021 at 4:09 PM	696 KB
188.227.d704.0b96_00118_20180930121754-http.logreplaced.log	1 May 2021 at 4:08 PM	695 KB
209.200.3167.3959_00399_20181027053716-http.logreplaced.log	1 May 2021 at 4:00 PM	681 KB
213.184.2d21.9dd9_00352_20171122014458-http.logreplaced.log	1 May 2021 at 12:49 PM	651 KB
77.78.e513.a78b_00291_20180501070736-http.logreplaced.log	1 May 2021 at 2:10 PM	621 KB
188.227.d704.0b96_00134_20180424180757-http.logreplaced.log	1 May 2021 at 4:09 PM	605 KB
188.227.d704.0b96_00120_20180930141754-http.logreplaced.log	1 May 2021 at 4:09 PM	603 KB
109.200.2806.636a_00001_20180418175824-http.logreplaced.log	1 May 2021 at 2:40 PM	590 KB
173.225.49be.6699_00102_20180227151814-http.logreplaced.log	1 May 2021 at 3:12 PM	554 KB
173.225.1539.32ec_00145_20180227181440-http.logreplaced.log	1 May 2021 at 2:58 PM	554 KB
77.78.e513.a78b_00102_20180227150327-http.logreplaced.log	1 May 2021 at 2:08 PM	551 KB
108.62.acb9.b7ae_00099_20180227164849-http.logreplaced.log	1 May 2021 at 2:24 PM	550 KB
109.200.3d7c.3d77_00109_20181023084735-http.logreplaced.log	1 May 2021 at 2:35 PM	511 KB
81.94.cfdf.2ece_00440_20180212232312-http.logreplaced.log	1 May 2021 at 5:02 PM	501 KB
108.62.acb9.b7ae_00157_20171024003606-http.logreplaced.log	1 May 2021 at 2:25 PM	484 KB
109.200.2806.636a_00104_20180227214947-http.logreplaced.log	1 May 2021 at 2:47 PM	483 KB
109.200.2806.636a_00027_20171006142828-http.logreplaced.log	1 May 2021 at 2:43 PM	481 KB
94.229.2f7c.94dd_00015_20180320001817-http.logreplaced.log	1 May 2021 at 1:43 PM	477 KB
108.62.acb9.b7ae_00158_20171024013606-http.logreplaced.log	1 May 2021 at 2:25 PM	457 KB

Dataset Organization

- Finally, command “cat * > mergedhttplogs.log” were executed via Terminal to **merge all https log files into a single log file** named **mergedhttplogs.log**

Understanding the Dataset

- There are 28 features (columns) and 1571285 rows in the dataset.

```
1  df.columns
```

```
Index(['ts', 'uid', 'id.orig_h', 'id.orig_p', 'id.resp_h', 'id.resp_p',
       'trans_depth', 'method', 'host', 'uri', 'version', 'user_agent',
       'request_body_len', 'response_body_len', 'status_code', 'status_msg',
       'tags', 'resp_fuids', 'resp_mime_types', 'proxied', 'username',
       'orig_fuids', 'orig_mime_types', 'referrer', 'origin', 'orig_filenames',
       'info_code', 'info_msg'],
      dtype='object')
```

```
1  len(df.index)
```

```
1571285
```

Features

Field	Type	Description
ts	time	Timestamp of request
uid	string	Connection unique id
id	record	ID record with orig/resp host/port. See conn.log
trans_depth	count	Pipelined depth into the connection
method	string	HTTP Request verb: GET, POST, HEAD, etc.
host	string	Value of the HOST header
uri	string	URI used in the request
referrer	string	Value of the “referer” header
user_agent	string	Value of the User-Agent header
request_body_len	count	Actual uncompressed content size of the data transferred from the client
response_body_len	count	Actual uncompressed content size of the data transferred from the server
status_code	count	Status code returned by the server
status_msg	string	Status message returned by the server
info_code	count	Last seen 1xx info reply code by server
info_msg	string	Last seen 1xx info reply message by server
filename	string	Via the Content-Disposition server header
tags	set	Indicators of various attributes discovered
username	string	If basic-auth is performed for the request
password	string	If basic-auth is performed for the request
proxied	set	Headers that might indicate a proxied request
orig_fuids	vector	An ordered vector of file unique IDs from orig
orig_mime_types	vector	An ordered vector of mime types from orig
resp_fuids	vector	An ordered vector of file unique IDs from resp
resp_mime_types	vector	An ordered vector of mime types from resp

Preprocessing of Dataset

- Convert Epoch to yyyy-mm-dd hh-mm-ss.sss format.

```
data = 'mergedhttplogs.log'
```

```
df=pd.read_json(data,lines=True)
df.head()
```

	ts	uid	id.orig_h	id.orig_p	id.resp_h	id.resp_p	trans_depth	method	host	
0	1.497436e+09	CuxTn81YmKFUFo41f6	209.126.136.4	54818	108.62.acb9.b7ae	80	1	GET	108.62.acb9.b7ae	
1	1.497507e+09	CUFsTf3jdvUrAydVC4	91.230.47.3	53360	108.62.acb9.b7ae	80	1	GET	NaN	
2	1.497510e+09	Ckal2E4oWp62g2ibaj	91.196.50.33	46511	108.62.acb9.b7ae	80	1	GET	testp3.pospr.waw.pl	http://testp3.pospr.wa
3	1.500863e+09	CTzP4i1734cb4sTlkd	108.62.acb9.b7ae	50692	91.189.95.15	80	1	GET	changelogs.ubuntu.com	
4	1.502951e+09	CVrs8bcvwLvRstobl	90.100.32.151	42618	108.62.acb9.b7ae	80	1	GET	NaN	

5 rows × 28 columns

```
df['ts'] = pd.to_datetime(df['ts'],unit='s')
df.head()
```

	ts	uid	id.orig_h	id.orig_p	id.resp_h	id.resp_p	trans_depth	method	host	
0	2017-06-14 10:31:36.047461888	CuxTn81YmKFUFo41f6	209.126.136.4	54818	108.62.acb9.b7ae	80	1	GET	108.62.acb9.b7ae	
1	2017-06-15 06:15:38.334464000	CUFsTf3jdvUrAydVC4	91.230.47.3	53360	108.62.acb9.b7ae	80	1	GET	NaN	
2	2017-06-15 07:01:16.574737920	Ckal2E4oWp62g2ibaj	91.196.50.33	46511	108.62.acb9.b7ae	80	1	GET	testp3.pospr.waw.pl	http://testp3.po
3	2017-07-24 02:22:30.761751040	CTzP4i1734cb4sTlkd	108.62.acb9.b7ae	50692	91.189.95.15	80	1	GET	changelogs.ubuntu.com	
4	2017-08-17 06:21:27.464701952	CVrs8bcvwLvRstobl	90.100.32.151	42618	108.62.acb9.b7ae	80	1	GET	NaN	

5 rows × 28 columns

Create “URI_Length” feature

- In various instances, the **length of a request parameter** may be **utilized to identify anomaly.**
  - E.g., to cause **buffer overflow** in an application, the shell code and additional padding, depending on the length of the target buffer, must be shipped. As a result, the attribute's length may be extremely long.
- Hence, **‘uri_length’** attribute is **created and used** as part of the analysis

```python
df['uri_length'] = df['uri'].str.len()
df.head()
```

t	uri	...	proxied	username	orig_fuids	orig_mime_types	referrer	origin	orig_filenames	info_code	info_msg	uri_length
e	/	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	1.0
N	/	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	1.0
l	http://testp3.pospr.waw.pl/testproxy.php	...	[PROXY-CONNECTION -> Keep-Alive]	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	40.0
n	/meta-release-lts	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	17.0
N	/anony/mjpg.cgi	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	15.0

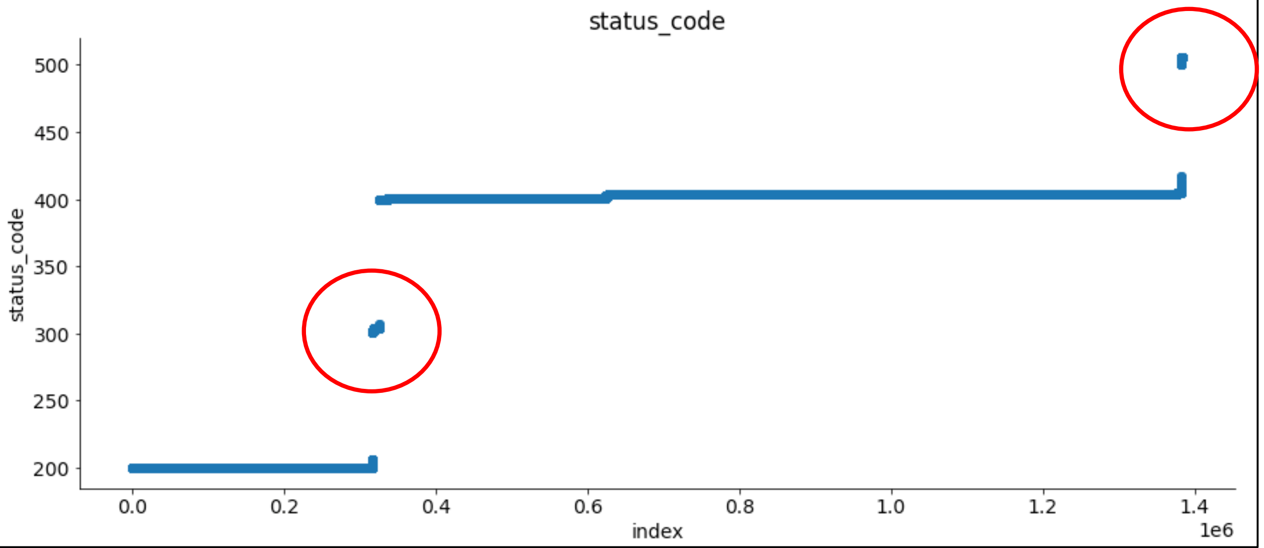
Sri Shaila G. Sri Shaila G, Ahmad Darki, Michalis Faloutsos, Nael Abu-Ghazaleh, and Manu Sridharan, “IDAPro for IoT Malware analysis?”, 2019, https://www.usenix.org/system/files/cset19-paper_g.pdf

Exploratory Analysis

- Exploratory Data Analysis refers to the **critical process of performing initial investigations on data** so as to discover patterns or to spot anomalies.
- We perform exploratory analysis on **several features** in order to **identify the right ones that should be part of further analysis**.

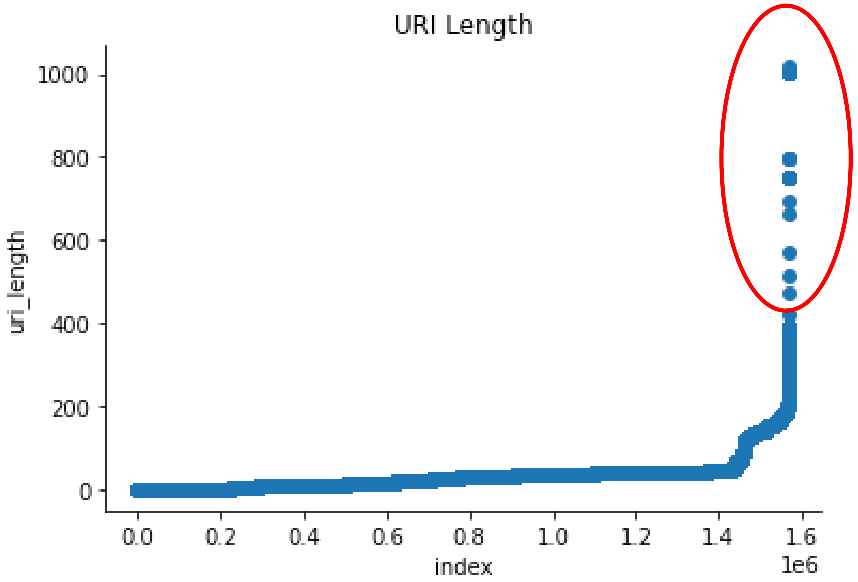
Exploratory Analysis on “status_code”

```python
plt.scatter(range(df.shape[0]), np.sort(df['status_code'].values))
plt.xlabel('index')
plt.ylabel('status_code')
plt.title("status_code")
sns.despine()
```



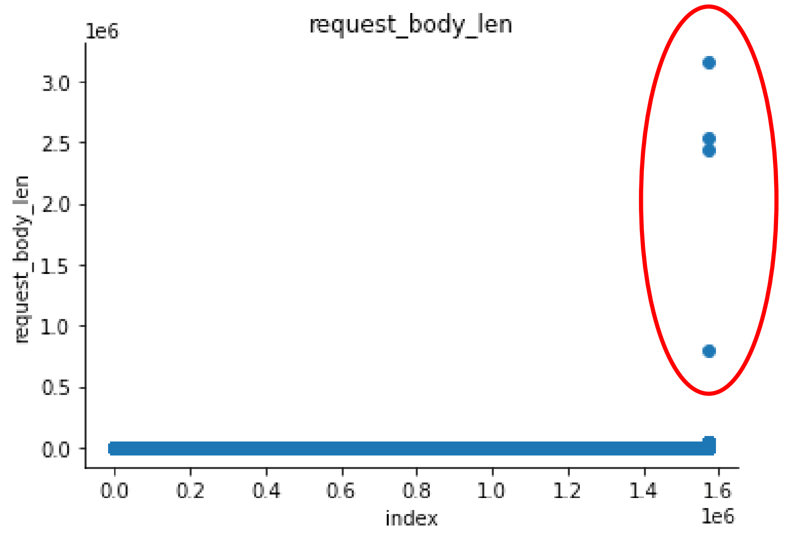
Exploratory Analysis on “uri_length”

```
plt.scatter(range(df.shape[0]), np.sort(df['uri_length'].values))
plt.xlabel('index')
plt.ylabel('uri_length')
plt.title("URI Length")
sns.despine()
```



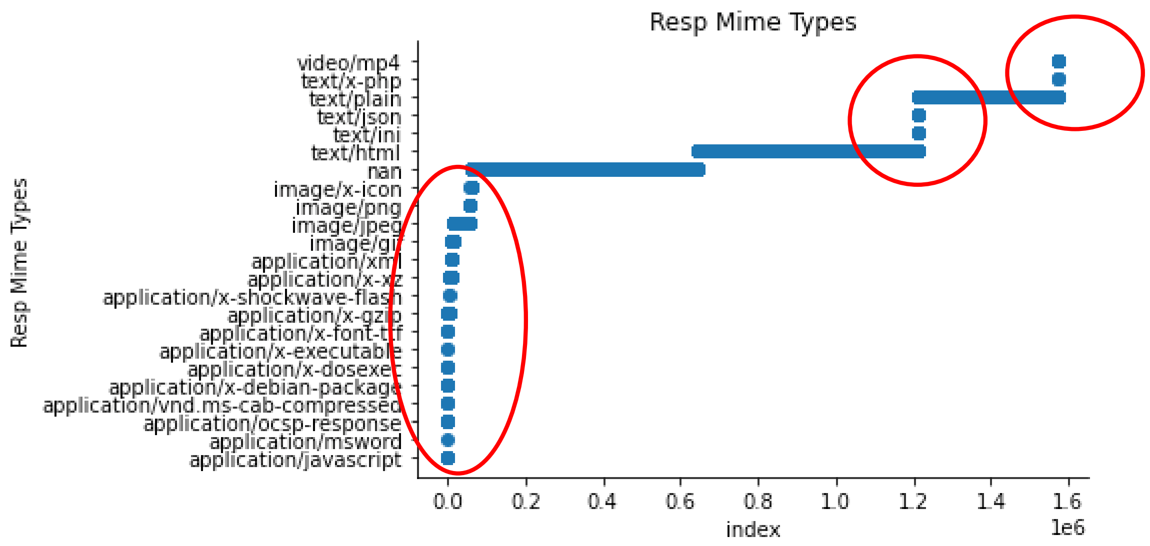
Exploratory Analysis on “request_body_len”

```python
plt.scatter(range(df.shape[0]), np.sort(df['request_body_len'].values))
plt.xlabel('index')
plt.ylabel('request_body_len')
plt.title("request_body_len")
sns.despine()
```



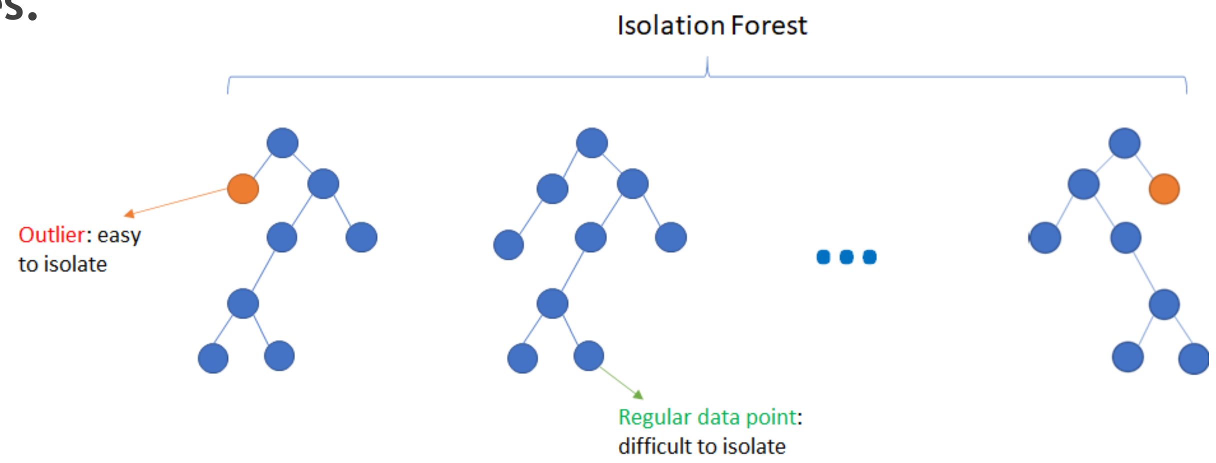
Exploratory Analysis on “resp_mime_types”

```
plt.scatter(range(df.shape[0]), np.sort(df['resp_mime_types'].values))
plt.xlabel('index')
plt.ylabel('Resp Mime Types')
plt.title("Resp Mime Types")
sns.despine()
```



Isolation Forest

- **Machine learning and unsupervised learning approach** for **detecting anomalies by isolating outliers.** The **Decision Tree method** is used in Isolation Forest. It **separates outliers by selecting a feature at random** from a set of features and then **selecting a split value between the feature's max and min values at random.**
- The **anomalous data points will be distinguished from the rest of the data** by this random partitioning of features, which will **result in shorter routes in trees.**



Defining the model

```
to_matrix = DataFrameToMatrix()
features = ['uri_length', 'resp_mime_types', 'request_body_len', 'status_code']
df_matrix = to_matrix.fit_transform(df[features])
odd_clf = IsolationForest(contamination=0.1)
odd_clf.fit(df_matrix)
```

```
Changing column resp_mime_types to category...
WARNING: resp_mime_types will expand into 23 dimensions...
Normalizing column uri_length...
Normalizing column request_body_len...
Normalizing column status_code...

IsolationForest(contamination=0.1)
```

- Shortlisted feature: ***‘resp_mime_types’***, ***‘request_body_len’***, ***‘status_code’***, ***‘uri_length’***.
- These features were selected since they **provide valuable information**. Also, based on **exploratory data analysis** that was done earlier, we can visually see the outliers found on these features.
- **Contamination parameter** sets the percentage of anomalous points in our data. Based on the visual and exploratory analysis that was done earlier, we will fit this to an isolation forest model with a contamination parameter of 0.1 (10%).

Anomaly Prediction

```
to_matrix = DataFrameToMatrix()
features = ['uri_length', 'resp_mime_types', 'request_body_len', 'status_code']
df_matrix = to_matrix.fit_transform(df[features])
odd_clf = IsolationForest(contamination=0.1)
odd_clf.fit(df_matrix)
```

```
Changing column resp_mime_types to category...
WARNING: resp_mime_types will expand into 23 dimensions...
Normalizing column uri_length...
Normalizing column request_body_len...
Normalizing column status_code...

IsolationForest(contamination=0.1)
```

- The **Isolation Forest will be generated** once the model has been **adequately trained.**
- **'anomaly_pred'** parameter, which refers to anomaly prediction result, is created after the model is defined and fit
- To get the values of the 'anomaly_pred' column, execute the trained model's predict() function and supplying the feature as a parameter.
- A '-1' denotes the presence of an anomaly by default, while a '1' reflects normal data.
  - However, to avoid any misunderstandings and to adhere to the **traditional notations of positive 1 and negative 0**, we will refer to '1' as anomaly data and '0' as normal data throughout this paper.

```
df['anomaly_pred'] = odd_clf.predict(df_matrix)
```

```
df['anomaly_pred'] = df['anomaly_pred'].replace(1,0)
df['anomaly_pred'] = df['anomaly_pred'].replace(-1,1)
```

Result and Evaluation

Result and Evaluation

- Result validation is a **critical phase** in the process since it assures that our model produces accurate results.
- In **supervised learning**, performance metrics such as accuracy, precision, recall, AUC, and others are **typically measured** on the **training and test data**. Such performance indicators aid in determining the viability of a model.
- However, because **we don't have the ground truth** in **unsupervised learning**, the method isn't as simple. It is **quite difficult to identify KPIs** that may be utilized to validate results in the **absence of labels**.

Result and Evaluation

- The process of detecting unusual items or events in datasets that deviate from the norm is known as anomaly detection. Anomaly detection is based on **two fundamental assumptions**:
  - Anomalies appear in the data only **infrequently**
  - Their characteristics **differ dramatically** from those of typical instances.
- As a result, **not all anomalies must be malicious**. Misconfigurations, benign data that does not appear frequently, and a **variety of other factors could all contribute to anomalies**.

Investigating the anomalies

- Since we **do not have the ground truth** in unsupervised learning, the method is not as simple. It is quite difficult to identify KPIs that may be utilized to validate results in the absence of labels.
- For this reason, we **set our target to 25 different kinds of variants** or keywords that could normally be found in IoT malwares and cyberattacks. We will call them ‘**Test Subjects**’.

Test Subjects

Test Subject	Description
Yakuza	IoT Attack User Agent
Hello, World	IoT Attack User Agent
Gemini	IoT Attack User Agent
RIAALABS	Reconnaissance
Ronin/2.0	IoT Attack User Agent
CarlosMatos/69.0	IoT Attack User Agent
Botnet	Botnet keyword
Hades	Hades Malware
Screaming Frog SEO Spider	Bots
Hakai	IoT Attack User Agent
.mips	Malware binaries
wordpress/xmlrpc	WordPress XML-RPC vulnerability
tftp	Vulnerable protocol
research	Research purpose logs
killall	Remote Code Execution
hakai	“hakai” keyword
sora	Mirai Variant
.arm	Malware binaries
seraph	Seraph Malware
mirai	“mirai” keyword
port=21	Vulnerable port
exploit	“exploit” keyword
wget	Remote Code Execution
chmod	privilege escalation
busybox	Remote Code Execution

Trial on 10 different combination of features

- It is **crucial that the right features** were selected as part of the process. To ensure we get the best accuracy score, trial on **10 different combinations** between the shortlisted features was done. The combinations are as follow:

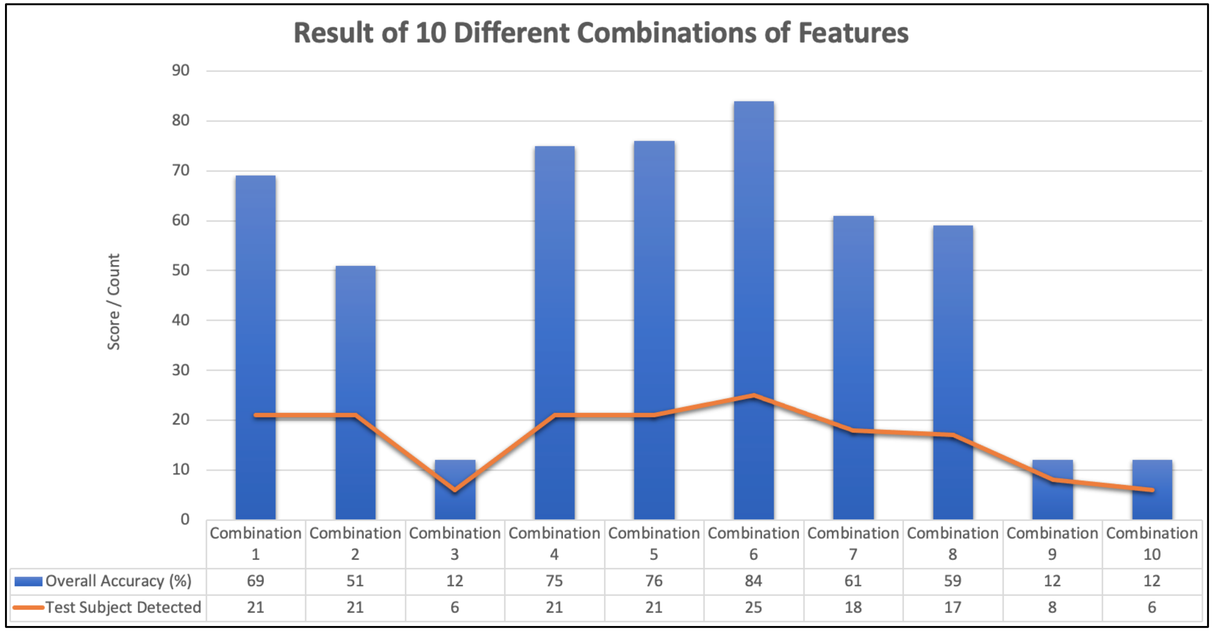
Combination	Features
1	“resp_mime_types”, “request_body_len”, “uri_length”, “status_code”
2	“resp_mime_types”, “request_body_len”, “uri_length”
3	“resp_mime_types”, “request_body_len
4	“request_body_len”, “uri_length”, “status_code”
5	“request_body_len”, “uri_length”
6	“uri_length”, “status_code”
7	“resp_mime_types”, “uri_length”, “status_code”
8	“resp_mime_types”, “uri_length”
9	“resp_mime_types”, “request_body_len”, “status_code”
10	“request_body_len”, “status_code”

Investigating the anomalies

- The method used to perform the test is by **creating a table of 4 columns**:
  - ‘**Anomaly**’ which refers to the **Test Subjects**
  - ‘**Predicted Count**’ refers to the **count of prediction done by the model** for each of the Test Subject
  - ‘**Actual Count**’ refers to the **count** of each of the Test Subject **found in the dataset**
  - ‘**Accuracy (%)**’ refers to the **accuracy score** based on comparison done between **predicted count against actual count**

Result of the trial

- **"Combination 6" ['uri_length', 'status_code']** yields the best result with an overall accuracy score of **84%**. The model also managed to **predict and detect all the 25 test subjects as anomaly**, with at least **15 out of 25** of them having accuracy score of **90% or more**.



Combination 6

```
Features Selected: ['uri_length', 'status_code']
Anomaly Found: 25 out of 25
Overall Accuracy(%): 84

+---------------------------+-----------------+---------------+--------------+
|          Anomaly          | Predicted Count | Actual Count  | Accuracy(%)  |
+---------------------------+-----------------+---------------+--------------+
|          Yakuza           |       50        |      99       |      51      |
|       Hello, World        |      1225       |     2243      |      55      |
|          Gemini           |      1092       |     1309      |      83      |
|         RIAALABS          |        2        |       2       |     100      |
|         Ronin/2.0         |       18        |      20       |      90      |
|      CarlosMatos/69.0     |       67        |      134      |      50      |
|          Botnet           |        5        |       5       |     100      |
|          Hades            |        4        |       8       |      50      |
|  Screaming Frog SEO Spider|        2        |       2       |     100      |
|          Hakai            |      1487       |     1487      |     100      |
|          .mips            |      2030       |     2031      |     100      |
|     wordpress/xmlrpc      |        2        |       5       |      40      |
|           tftp            |      3253       |     3342      |      97      |
|         research          |       19        |      47       |      40      |
|          killall          |       78        |      93       |      84      |
|          hakai            |       379       |      380      |     100      |
|           sora            |       134       |      138      |      97      |
|           .arm            |      17448      |     17493     |     100      |
|          seraph           |      1103       |     1103      |     100      |
|          mirai            |       21        |      21       |     100      |
|         port=21           |      65788      |     92568     |      71      |
|         exploit           |       977       |      981      |     100      |
|           wget            |      17190      |     17250     |     100      |
|          chmod            |      11085      |     11094     |     100      |
|         busybox           |       90        |      104      |      87      |
+---------------------------+-----------------+---------------+--------------+
```

Conclusion

Conclusion

- This research project gave an **outline of the Internet of Things (IoT)** and **how it relates to cybersecurity in today's world**. The study also highlighted **how Honeypot functions** in general and how **Machine Learning is typically used for anomaly detection.** We also contrasted **supervised versus unsupervised** learning, their benefits as well as drawbacks.
- We provided a high-level overview of the **honeypot setup**, built a **data analysis workflow** that includes feature engineering from **honeypot network traffic data** and the **application of an appropriate unsupervised machine** learning technique called **Isolation Forest** to **find anomalies** and better **understand the threat landscape** over the **honeypot's operational period.**
- The **validation and results** of our model and **implementation** brought the project to a close.

Future Work

Future Work

- The research project has presented a framework for the analysis of IoT network traffic. The work described in this project points to various areas for future research.
  - Detection of Anomalies in Real Time
    - Other datasets can benefit from the methods presented in this paper. It will be interesting to see how effective the proposed technique is with a network trace that includes both normal and malicious activities. This will help to demonstrate how well the suggested approach can be adapted to various datasets in order to detect malicious behaviour.
  - Application of the proposed model to different dataset
    - The suggested approach has been utilized to evaluate a huge library of http traffic in an offline mode. We believe they can be extended further to monitor traffic in real time. Malicious actions must be identified and diagnosed quickly and accurately in a real-time context.

Thank You