USE OF MACHINE LEARNING FOR SECURING IoT

Authored by Arashpreet Singh

A Project Report

Submitted to the Faculty of Graduate Studies,

Concordia University of Edmonton

in Partial Fulfillment of the

Requirements for the

Final Research Project for the Degree

MASTER OF INFORMATION SYSTEMS SECURITY MANAGEMENT

Concordia University of Edmonton

FACULTY OF GRADUATE STUDIES

Edmonton, Alberta

December 2020

USE OF MACHINE LEARNING FOR SECURING IoT

Arashpreet Singh

Approved:

Sergey Butakov [Original Approval on File]

Sergey Butakov Date: December 12, 2020

Primary Supervisor

Edgar Schmidt [Original Approval on File]

Edgar Schmidt, DSocSc Date: December 14, 2020

Dean, Faculty of Graduate Studies

Use of machine learning for securing IoT

Arashpreet Singh

Student ID #140543

alnu12@student.concordia.ab.ca

Research Advisor: Dr. Sergey Butakov

sergey.butakov@concordia.ab.ca

Department of Information Systems Security Management

Concordia University of Edmonton, Edmonton T5B 4E4, Alberta, Canada

Abstract- IoT comprises of cyber-physical devices connected through the internet. These devices carry sensitive data and support critical services. Due to the sensitivity of user data shared across IoT, there is a risk of attacks such as spoofing, eavesdropping, and denial of service (DoS). These attacks can be detected and prevented by cybersecurity systems that use machine learning techniques for malicious pattern analysis.

The primary objective of the research is to develop an improved machine learning model that classifies network traffic as either malicious or benign. For this purpose, the IoT-23 dataset is used, which includes twenty-three scenarios of network traffic, out of which twenty are from malware-infected IoT devices, and three are from benign IoT devices. The dataset has several feature columns that are transformed in a way to feed into the model using feature engineering techniques. The model is constructed using a random classifier by choosing parameters to increase the accuracy of classifying network traffic. This accuracy can be improved by constructing an ensemble model that combines random forest classifier with other classifiers such as K-Nearest Neighbor and Gaussian Naïve Bayes using hard voting. This research provides a model so that malicious traffic can be detected with more accuracy using machine learning algorithms. The results of the model can be evaluated using confusion matrix.

Keywords –Traffic Analysis, Classification, KNN, Random Forest, Ensemble Learner

I. INTRODUCTION

The Internet of things (IoT) is the combination of various smart devices connected through the Internet that share data with each other with or without human intervention. Such smart devices as refrigerators, lightbulbs, medical monitors etc. are exchanging data with associated services through the Internet. This data could be less sensitive data such as maker of a device, a software version or it could be very sensitive piece of information such as exact location or health monitoring data of an individual [1].

Since vast number of heterogeneous devices connected to each other in IoT, security concerns related to privacy/information leakage or various attacks such as spoofing, denial of service also come into play [2] [3]. There have been various attacks already deployed on IoT applications such as Mirai attack in 2016 that infected around 2.5 million devices connected to the Internet and launched distributed denial of service (DDoS) attack [4]. After Mirai, Hajime and Reaper are the other botnet attacks launched against large number of IoT devices [4].

Regardless of these attacks, wearable IoT devices are gaining popularity among casual users and healthcare professionals. Gartner Inc. forecasted the IoT market will grow to 5.8 billion devices by the end of 2020 with 21% increase from 2019 (4.8 billion) [5]. They are collecting and sending out health-related data for billions of users and require careful protective mechanisms to be devised to avoid data leaks and other attacks[1].

One of the solutions to protect information that is being transmitted over networks with IoT devices is to use machine learning (ML) algorithms. ML techniques such as supervised learning, unsupervised learning and reinforcement learning can be combined with IoT security techniques for securing user data and protection against different types of attacks. These techniques could be as follows:

- Learning based authentication to protect from such attacks as eavesdropping, spoofing etc.

- Learning based malware detection for detecting viruses and trojans in IoT devices
- Learning based access control for protection against data leak and denial-of-service (DoS) attacks
- Secure IoT offloading with ML against man-in-the-middle attack, jamming and DoS attack [1]

In this paper, focus is on securing networks with IoT devices using learning-based malware detection. Specifically, the paper is looking into detection of malware in action by capturing and analyzing network traffic. Malware generated traffic is then compared to the typical traffic from non-infected devices. Specific features can be extracted from this comparison based on which one can easily classify traffic as either benign (normal) or malicious using various ML algorithms such as K-nearest neighbor (KNN), random forest etc.

The organization of the rest of the paper is as follows: Section II describe the previous research done in this field. Section III describe the methodology used in this research and discusses the results. Section IV presents the conclusion of the research done and future directions for the research.

II. RELATED WORKS

The main motivation to use ML for bad traffic detection on the IoT networks came from the numerous successful examples of ML based traffic analysis in more traditional networks. Of course, on the lower level protocols IoT networks use the same IP/TCP/UDP networks as traditional networks but there are also noticeable differences. IoT networks generate large amount of network traffic as compared to traditional networks due to high number of users and heterogeneous devices in IoT as well as wide variety of services available. For example, some devices may transmit just location sensor readings creating relatively low and stable traffic while others can transmit high volume of video traffic or can generate minor alarm traffic because of some unpredictable events such as heart attack of the sensor carrier. Due to these factors, traffic in IoT networks is more complex and less predictable and malicious traffic can be better hidden and can live longer using camouflage techniques [6].

Various IoT applications allow exchanging data automatically without human intervention. This require higher levels of security and privacy protection as well as arrangements for automatic recovery from various attacks. Hassija et al. discussed security issues at different layers in IoT system (i.e. sensing layer, network layer, middle-ware layer and application layer) [7]. After assessing security issues, various existing and emerging technologies such as blockchain technology, machine learning techniques were discussed. Particle swarm optimization and back propagation algorithm to train multi-layer perceptron (MLP) that helps in increasing the security of wireless networks has been discussed in this paper. Despite this, to secure privacy leakage, commodity integrity detection algorithm (CIDA) has been discussed which is based on Chinese remainder theorem (CRT) and for digital fingerprinting, support vector machine (SVM) and artificial neural networks (ANN) has been discussed that uses the digital value of various features in fingerprint to train the algorithm.

Due to limited resources available on IoT devices, they can be vulnerable to even trivial attacks. So, I. Hafeez et al. proposed IoT-Keeper to secure the communication of IoT by using an anomaly detection technique that perform traffic analysis at edge gateways [8]. To analyze network traffic and detecting malicious network activity, IoT-Keeper uses a combination of fuzzy C-means clustering and fuzzy interpolation scheme. IoT-Keeper automatically enforces network access restrictions against IoT device generating malicious activity and prevents it from attacking other devices. In this paper, three datasets have been used for anomaly detection: YTY2018 (Kitsune dataset), Keeper (dataset collected from IoT-Keeper testbed) and combination of both. IoT-Keeper achieves TPR=0.99 for DoS attack and TPR=0.98 for scanning attacks whereas DIoT (Defense Internet of Things) achieves TPR=0.89 for DoS attack and TPR=1.0 for scanning attacks. However, IoT-Keeper cannot secure user data as it does not perform deep packet inspection. Moreover, IoT-Keeper sets up network access restriction based on layer-2 MAC addresses. An attacker can perform MAC address spoofing and will attain network access without exhibiting any malicious activity. Also, MAC address spoofing can be employed to perform DoS attack.

R. Doshi et al. investigated the use of IoT specific network behaviors (for example regular time intervals between packets) for feature selection. It was demonstrated that such a selection can help to detect DDoS attack with higher accuracy in IoT traffic [9]. This research suggested that home gateway routers could automatically detect malicious traffic sources using low cost machine learning algorithms. In their

research, a combination of normal and DoS traffic is used, and classifier is trained using 85% of dataset used as training dataset and rest is used as test dataset. Linear SVM classifier performed worst on the dataset with least accuracy whereas decision tree and KNN achieve accuracy of 99%. However, these results need to be replicated by using normal traffic from more IoT devices and using different machine learning algorithms. This helps to achieve better DoS detection accuracy.

A. Sivanathan et al used network traffic characteristics to classify IoT devices as either compromised by attack or normal [10]. They first collect network traffic traces from an infrastructure consists of 28 IoT devices, then analyze traffic characteristics using statistical attributes such as port number and signaling patterns. Later, they develop multi-stage machine learning based classification algorithms to acquire results with over 99% accuracy. The dataset used in this research consists of network traffic collected using tcpdump tool running on OpenWrt. The classifier used in this technique is combination of bag-of-words and random forest classifier. However, the dataset used in this research consists of only 50,378 labeled instances that is small number as we cannot get actual levels of accuracy on such a small dataset. Moreover, this dataset was collected in 2018 due to which future work is required to get up-to-date dataset that includes latest attacks and to detect malicious behavior due to security breaches.

It is evident from the review of previous work that the field of using ML to protect IoT networks is relatively new, and lack of available datasets led to the lack of the projects that protect IoT devices against malware by using previously captured patterns of malicious traffic. In the next section, an approach to detect malicious traffic has been proposed to enhance security of the IoT devices. This approach uses IoT-23 dataset that consist of 23 scenarios out of which twenty scenarios are from malware infected IoT devices and three are from normal IoT devices [11]. This dataset consists of more than million labeled instances. By investigating malicious traffic, features are found that differentiate malicious traffic from normal traffic based on which we can classify any traffic easily.

III. USING MACHINE LEARNING TO SECURE IOT DEVICES

Presented research explores the following hypothesis: The ensemble model based on ML algorithms can provide acceptable classification accuracy on IoT-23 dataset.

The model employs a meta estimator which combines three classifiers - Random Forest Classifier, Gaussian Naïve Bayes and K-Nearest Neighbors (KNN) and use hard voting for the final decision. Acceptability of the classification will be defined by the percentage of false positives and false negatives and can vary by the application area.

A. METHODOLOGY

Methodology suggested below is prepared to test the following research hypothesis:

Dataset: Dataset used for this experiment is IoT-23 dataset which is a labeled dataset with malicious and benign IoT network traffic. This dataset contains 23 captures of IoT network traffic out of which twenty are from infected IoT devices and three are from normal IoT devices.

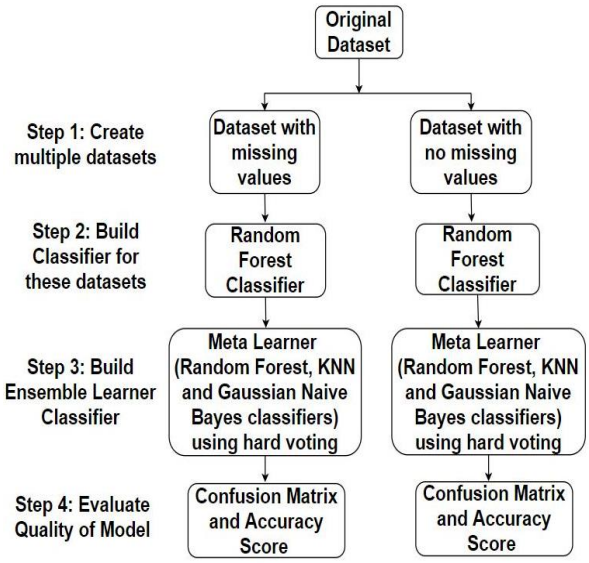

Figure 1. Schematic representation of methodology

B. DESCRIPTION OF DATASET

IoT-23 dataset consists of 23 features out of which two features specifies time, two specifies IP addresses, two specifies port numbers, two are Boolean values, two are strings, six are classes and rest of the features are numeric values [11]. From these 23 features, we used only first 21 features and apply feature engineering on these features only. Detailed description of dataset is provided in appendix A.

The actual size of the dataset is 22 GB. The experiment is performed on a smaller dataset by selecting 15% of the entire dataset randomly. The reduction was done due to limit of computational resources available for the project. The random selection was used to ensure proper representation of the traffic patterns from an entire dataset.

The dataset is combination of 23 scenarios. By analyzing dataset, it has been noted that values are missing for certain features (i.e. duration, orig_bytes, resp_bytes). Appendix D describes the actual size of captured patterns for each scenario and percentage of values missing for above mentioned features from these scenarios.

In 23 scenarios of IoT-23 dataset outlined in the Appendix D, several attributes are missing. For each scenario percentage of missing attributes varies from 0% to 99% (for example some scenarios such as CTU-IoT-Malware-Capture-33-1, CTU-IoT-Malware-Capture-36-1 etc. have attributes missing in almost all the samples).

Because of missing values for certain features, dataset is further divided into two categories: One dataset with values missing for these features and second dataset with all the values. All experiments are performed on these datasets only.

C. EXPERIMENTS

During experiment, 70% of data has been used as training data for the model and rest 30% has been used as testing data to test the model based on various factors such as accuracy, confusion matrix etc. 70/30 split is typical for data exploration tasks when first models are being built on a relatively new dataset.

In first model, random forest classifier is used with parameters n_estimators = 2 and random_state = 40. N_estimators is used to specify the number of decision trees in the forest. Random_State is used for reproducing the problem same every time it runs.

In second model, Ensemble learner model is built using random forest classifier, Gaussian Naïve Bayes and KNN by using hard voting. Hard voting process output from these three classifiers and selects best output. That output is then returned by the ensemble learner model.

Gaussian Naïve Bayes classifier is used with default parameters to process the data while KNN is used with n_neighbors = 100. N_neighbors basically specifies the number of nearest neighbors which is a core deciding factor in this classifier. Random forest classifier is used with same parameters as in first model.

D. DISCUSSION OF EXPERIMENTS

The results of the experiment have been evaluated with two criteria to evaluate quality of model: accuracy and confusion matrix.

The accuracy score describes how accurately model can classify traffic as either malicious or benign using previously available data. The following tables shows the accuracy score of the various classifiers on both datasets.

Algorithm	Accuracy Score
Random Forest	0.999
Ensemble Learner	0.896

Table 1. Accuracy Score of Dataset with missing values for certain features

Algorithm	Accuracy Score
Random Forest	0.999
Ensemble Learner	0.996

Table 2. Accuracy Score of Dataset with no missing values for certain features

The confusion matrix shows True Positives (TP), True Negatives (TN), False Positives (FP) and False Negatives (FN) evaluated by the model. Following figures shows the confusion matrix for ensemble model on both datasets. Malicious traffic is represented with 0 while benign traffic is represented with 1.

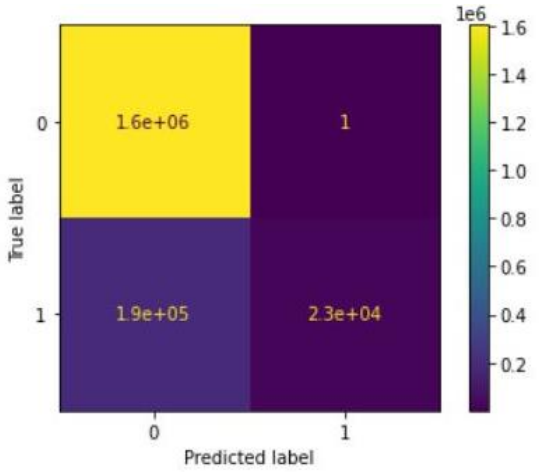

Figure 2. Confusion matrix for dataset with missing values

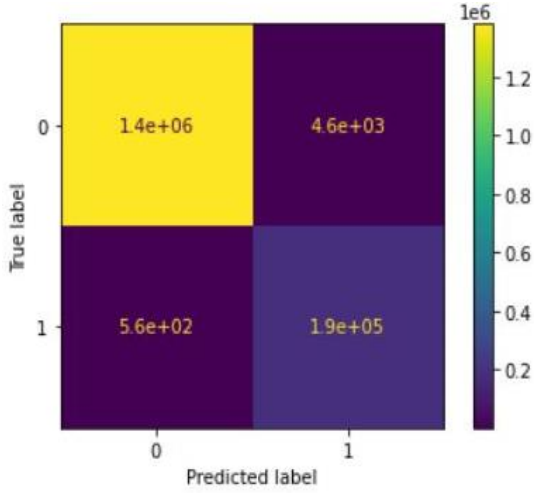

Figure 3. Confusion matrix for dataset without missing values

From above figures, it has been found that the model that works on the dataset with missing values

produces higher false positive rate (i.e. 10.6%) as compare to the one working on the dataset with no missing values (i.e. 0.04%).

False positives - when benign traffic is classified as malicious - may force the intrusion detection system (IDS) to cause unintentional DoS and benign traffic will not reach its destination. The very low percentage in the portion of the dataset with no missing values makes it less of the concern. Relatively high volume of misclassified traffic with missing values requires additional research. However, false negatives for dataset with no missing values is 0.33%.

It can be inferred, false negatives in case of all attributes present essentially do not exist and false negatives on the dataset with missing values are below 1%. This means that if those packets represent DDoS attack then the attack was effectively prevented. Future research is required to analyze the nature of the attacks that were improperly classified in the false negative outcomes.

IV. CONCLUSION

In this research, ML algorithms have been used to classify network traffic from larger dataset (IoT-23 dataset) as either benign or malicious. A meta learner model (i.e. ensemble learner) has been built using random forest, gaussian naïve bayes and KNN classifier. Using this model, traffic is classified with high accuracy (between 89% to 99%) and true positive and true negative rates are also high. However, false positive rates have been found on higher side on some parts of the dataset with missing values. So, future research is required to explore the possibility of improving the accuracy of the built model.

V. REFERENCES

[1] L. Xiao, X. Wan, X. Lu, Y. Zhang and D. Wu, "IoT Security Techniques Based on Machine Learning: How Do IoT Devices Use AI to Enhance Security?," *IEEE Signal Processing Magazine,* vol. 35, no. 5, pp. 41-49, September 2018.

[2] R. Roman, J. Zhou and J. Lopez, "On the features and challenges of security and privacy in distributed internet of things," *Computer Networks,* vol. 57, no. 10, pp. 2266-2279, 5 July 2013.

[3] S. Chen, H. Xu, D. Liu, B. Hu and H. Wang, "A Vision of IoT: Applications, Challenges, and Opportunities With China Perspective," *IEEE Internet of Things Journal,* vol. 1, no. 4, pp. 349-359, July 2014.

[4] "Mirai Botnet Linked to Dyn DNS DDoS Attacks," Flashpoint, 21 October 2016. [Online]. Available: https://www.flashpoint-intel.com/blog/cybercrime/mirai-botnet-linked-dyn-dns-ddos-attacks/. [Accessed 21 September 2020].

[5] L. Goasduff, "Gartner," 29 August 2019. [Online]. Available: https://www.gartner.com/en/newsroom/press-releases/2019-08-29-gartner-says-5-8-billion-enterprise-and-automotive-io. [Accessed 27 September 2020].

[6] H. Tahaei, F. Afifi, A. Asemi, F. Zaki and N. B. Anuar, "The rise of traffic classification in IoT networks: A survey," *Journal of Network and Computer Applications,* vol. 154, 15 March 2020.

[7] V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal and B. Sikdar, "A Survey on IoT Security: Application Areas, Security Threats, and Solution Architectures," *IEEE Access,* vol. 7, pp. 82721-82743, 2019.

[8] I. Hafeez, M. Antikainen, A. Y. Ding and S. Tarkoma, "IoT-KEEPER: Detecting Malicious IoT Network Activity Using Online Traffic Analysis at the Edge," *IEEE Transactions on Network and Service Management,* vol. 17, no. 1, pp. 45-59, March 2020.

[9] R. Doshi, N. Apthorpe and N. Feamster, "Machine Learning DDoS Detection for Consumer Internet of Things Devices," in *IEEE Security and Privacy Workshops (SPW)*, San Francisco, CA, 2018.

[10] A. Sivanathan, H. H. Gharakheili, F. Loi, A. Radford, C. Wijenayake, A. Vishwanath and V. Sivaraman, "Classifying IoT Devices in Smart Environments Using Network Traffic Characteristics," *IEEE Transactions on Mobile Computing,* vol. 18, no. 8, pp. 1745-1759, 1 August 2019.

[11] A. Parmisano, S. Garcia and M. J. Erquiaga, "A labeled dataset with malicious and benign IoT

network traffic," Stratosphere Laboratory, 22 January 2020. [Online]. Available: https://www.stratosphereips.org/datasets-iot23. [Accessed 10 May 2020].

APPENDIX A - DETAILED DESCRIPTION OF DATASET

Attribute	Type	Description
Ts	time	Timestamp of connection (represented in timestamp format with captures ranging from 2018 to 2019)
Uid	string	Unique id of connection (a string 18 characters long including alphabets and digits)
id.orig_h	addr	Originating endpoints IP address (includes IPv4 addresses for example 169.254.15.115,192.168.1.132, 192.168.2.3 etc. and IPv6 addresses for example fe80::5bcc:698e:39d5:cdf etc.)
id.orig_p	port	Originating endpoints TCP/UDP port (nominal values such as 1900, 46180, 56704 etc.)
id.resp_h	addr	Responding endpoints IP address (includes IPv4 addresses for example 91.42.47.63, 120.210.108.200 etc. and IPv6 addresses for example ff02::1:ffd5:cdf, ff02::16 etc.)
id.resp_p	port	Responding endpoints TCP/UDP port (nominal values such as 23, 80, 81, 123, 443, 53,37215 etc.)
proto	enum	Transport layer protocol of connection (tcp, udp, icmp)
service	string	Dynamically detected application protocol if any (for example http, dns, ssl etc.)
duration	interval	Time of last packet seen - time of first packet sent (decimal values ranging from 0.0005 to 78840.329305)
orig_bytes	count	Originator payload bytes; from sequence numbers if TCP (nominal values ranges from 0 to 15072)
resp_bytes	count	Responder payload bytes; from sequence numbers if TCP (nominal values ranges from 0 to 1539)
conn_state	string	Connection state (details mentioned in Appendix B)
local_orig	bool	If connection originated locally True; if remotely False
local_resp	bool	If connection responded locally True; if remotely False
missed_bytes	count	Number of missing bytes in content gaps (nominal value encountered 0 for most instances)
history	string	Connection state history (details mentioned in Appendix C)
orig_pkts	count	Number of originated packets (nominal values ranges from 0 to 186)
orig_ip_bytes	count	Number of originated IP bytes (nominal values ranges from 0 to 65664)
resp_pkts	count	Number of responded packets (nominal values ranges from 0 to 122)
resp_ip_bytes	count	Number of responded IP bytes (nominal values ranges from 0 to 178178)
tunnel_parents	set(string)	If tunneled connection UID of encapsulating parent (empty for all the instances)
label	string	Label of traffic (malicious or benign)
detailed-label	string	Detail description of traffic (attack, benign, C&C, DDoS, File Download, HeartBeat, Mirai, Okiru, PartOfAHorizontalPortScan, Torii)

APPENDIX B - DESCRIPTION OF CONNECTION STATE

State	Meaning
S0	Connection attempt seen, no reply
S1	Connection established, not terminated (no bytes sent)
SF	Normal establish and termination (bytes sent)
REJ	Connection attempt rejected
S2	Established, ORIG attempts close, no reply from RESP
S3	Established, RESP attempts close, no reply from ORIG
RSTO	Established, ORIG aborted (RST)
RSTR	Established, RESP aborted (RST)
RSTOS0	ORIG sent SYN then RST, no response SYN-ACK
RSTRH	RESP sent SYN-ACK then RST, no ORIG SYN
SH	ORIG sent SYN then FIN, no RESP SYN-ACK (“half open”)
SHR	RESP sent SYN-ACK then FIN, no ORIG SYN
OTH	No SYN, not closed. Midstream traffic. Partial connection.

APPENDIX C - DESCRIPTION OF CONNECTION HISTORY

Letter	Meaning
S	A SYN without the ACK bit set
H	A SYN-ACK(“handshake”)
A	A pure ACK
D	Packet with payload
F	Packet with FIN bit set
R	Packet with RST bit set
C	Packet with a bad checksum
I	Inconsistent packet

APPENDIX D - DESCRIPTION OF MISSING VALUES FOR EACH SCENARIO

Name of Scenario	Actual Size	Missing Values (%)
CTU-Honeypot-Capture-4-1	58 KB	0.2
CTU-Honeypot-Capture-5-1	172 KB	21.6
CTU-Honeypot-Capture-7-1	17 KB	0
CTU-IoT-Malware-Capture-1-1	138 MB	78.9
CTU-IoT-Malware-Capture-3-1	23 MB	47.4
CTU-IoT-Malware-Capture-7-1	1 GB	99.7
CTU-IoT-Malware-Capture-8-1	1 MB	59.4
CTU-IoT-Malware-Capture-9-1	923 MB	99.7
CTU-IoT-Malware-Capture-17-1	7 GB	0
CTU-IoT-Malware-Capture-20-1	397 KB	27.6
CTU-IoT-Malware-Capture-21-1	406 KB	35.3
CTU-IoT-Malware-Capture-33-1	7 GB	99.9
CTU-IoT-Malware-Capture-34-1	3 MB	77
CTU-IoT-Malware-Capture-35-1	1 GB	42.4
CTU-IoT-Malware-Capture-36-1	2 GB	99.9
CTU-IoT-Malware-Capture-39-1	10 GB	0
CTU-IoT-Malware-Capture-42-1	555 KB	22.4
CTU-IoT-Malware-Capture-43-1	8 GB	0
CTU-IoT-Malware-Capture-44-1	30 KB	34.2
CTU-IoT-Malware-Capture-48-1	494 MB	0
CTU-IoT-Malware-Capture-49-1	793 MB	0
CTU-IoT-Malware-Capture-52-1	3 GB	0
CTU-IoT-Malware-Capture-60-1	432 MB	99.9