Remote Access Trojan Detection On Android Based On Network Traffic Observation Using Machine Learning

Muhammad Alwi Shihab[1] and Benfano Soewito[2]

[1,2] Computer Science Department, BINUS Graduate Program – Master of Computer Science, Bina Nusantara University, Jakarta, Indonesia

Abstract **–** Cellphones, where people always use cellphones in their daily life and the number of cellphones is greater than human population. The internet has become an important for more than half of population. Consequently, the popularity of cellphones, especially Android phones make these devices attractive to criminals in the digital world. Based on statistics, Android is a mobile operating system that is vulnerable to attacks compared to other operating systems. This paper used data based on network traffic features to classify and operate remote access trojans. Based on references, accuracy of the result is above 90% for the random forest algorithm and above 80% for the decision algorithm. The study also proves that previous research methods regarding trojan remote access which refer to network traffic features were very effective in dealing with these trojans.

Keywords: *Remote Access Trojans, Network Based Detection.*

1. Introduction

In the modern era, technology is rapidly growing, as an example is technology of smartphone. Cellphones are classified into several categories based on the operating system. Android is one of the largest and the most commonly used smartphone operating systems and Android is playing an essential role in various fields in the current development phase of the earth [**16**], because of that Android is a desirable target for attackers. Portability is one of the reasons that cellphones have become popular in everyday life and are an integral part. Cellphones store a lot of information that users do including personal information. This is an important factor for individual or organization users to protect information from malware attacks that result in data leaks. Organizations can lose reputation and disrupt their business [**1**].

Malwares are basically the malicious softwares which the user unknowingly install and then these applications either start interrupting the functioning of the device and perform malicious activities without getting detected [**9**]. Remote Access Trojan is a malware variant that can infect the Android operating system, the number and variety of malicious mobile apps have increased drastically, which brings insurmountable challange for malicious app detection [**5**]. Analyzing and detecting malware on Android operating system become very necessary [**4**], it allows protection of mobile device users from malware that steals user data or spies on users it allows aggregation and analysis of an application's network traffic patterns to be used for protecting cellular infrastructure from malicious and "network unfriendly" applications [**11**]. This Trojan uses remote command and secretly controls via the internet network. Firewall, antivirus system detection and others prevention are some of the methods used to save the system from malwares. The author focuses on the network analysis that generated by RAT, then machine learning aids were used in classifying networks that produce Trojans and not Trojans. Malware detection in Android generally falls in one of the two categories namely static feature-based detection and dynamic feature-based detection performed using static and dynamic analysis respectively [**10**]. Network-based malware detection methods are very effective on cellphones [**6**] and it categories as dynamic feature-based. Resulting patterns on network traffic can be used to check for malicious applications and normal applications [**7**]. Resulting traffic network will then be classified using machine learning.

This research was conducted based on previous research. Based on Ref. 1 and 2 which carried out tests based on the attributes generated by the remote access Trojan. Then it was conducted experiment using different datasets to prove whether the method would produce the same accuracy using machine learning that would described in detail in the next point. The author conducted research on how to detect remote access Trojans based on network traffic generated by trojans and test them using machine learning. The basic reason for how this paper can also help other researchers on how to specifically detect remote access trojans based on the resulting network traffic dataset.

Machine learning is a branch of artificial intelligence which has the ability to access and analyze data rapidly and automatically [**15**]. Machine learning is also capable of learning the data and attributes as well as classifying or grouping [**14**], performing tasks that have been determined by algorithmic methods and statistical models based on datasets. The Confusion Matrix is used as a medium for classifying the accuracy generated by machine learning. Based on the Confusion Matrix, it can determine Accuracy, Precision, Recall and that will also be determined by four cases which will be described as True Positive, False Positive, False Negative and True Positive.

Previous researchers used four main stages before the classification experiments as Feature Selection, Classifier Selection, Malware Samples and Data Collection. The performance of malware detection approaches depends critically both on the extracted features and the classification techniques [**8**]. The collected data was in the form of network traffic on malware-infected devices, and then the researcher determined seven points to compare between network traffic from malware and from normal applications, such as Average Packet Size, Ration of Incoming to Out Going Bytes,

Average Number of Bytes Received per Second, Number of Packets Sent Per Flow, Average Number of Packets Receive per Flow, Number of Bytes Sent per Flow and Number of Bytes Received per Flow from thirteen malware samples. The dataset was tested using the machine learning algorithm Decision Trees and resulted in an accuracy of 90% [**2**]. Other researchers used seven attributes, those were Out Byte, In Byte, In Byte By In Packet, Out Byte By Out Packet, Duration, Out Byte By In Byte, Out Packet By In Packet from 300 trojans and 300 normal application samples. One data packet in the research dataset was an extraction from 20 packages that produced by trojans. It was then tested using the machine learning algorithm that produced 99.2% of Decision Tree, 99.3% of Random Forest and 87.8% of Naïve Bayes [**1**].

Other studied used two main stages. First stage was the Feature Ranking Method and it has two features, those were Information Gain and Chi-Square Test. The second stage was Feature Selection with 22 types of features, included Average Packet Size, Ratio of Incoming to Outgoing Bytes, Byte Received per Second, Packets Sent and Received per Flow, Bytes Sent and Received per Flow, Average Time Interval and Flow etc. [**3**]. All the features have been collected, and the researcher tested them using the Naïve Bayes machine learning algorithm and resulted in an accuracy of 83.3% for nine features and 87.25% for 22 features.

2. Methodology

The author planed based on a framework that has been arranged, including conducting a Literature Study by reading and reviewing previous research; Problem Formulation based on the results of literature reviews; Collecting data based on raw data generated by remote access trojans; Data Classification was a process needed to create training data and assisting the analysis process; and Analyzes was analyzing the results of data classification based on machine learning, for example, to produce a tree from a decision tree algorithm that helps explain the classification of the data. Machine-learning methods were used for learning and detection purposes [**11**].

Based on a review or literature study from several references that previously described, this paper conducts a research phase by taking three methods, there are Feature Extraction, Data Training and Detection. Formation of the labeled dataset is an important part of the training step and collecting the PCAP files to extract the features which would help in training the model efficiently [**10**].

Figure 1. Research Method Chart

Feature Extraction is the process of taking useful features from existing data. In this case, the second dataset used this stage because it uses the attributes contained in the raw data which will be useful for research and testing. Data Training is a test process that aims to introduce machine learning in determining action and in this classification case, machine learning (such as the Decision Tree) will learn how the data that is actually generated by trojans and not trojans is then used as a reference for classifying testing data. Detection is an attempt to find and determine the network traffic tested is the traffic whether from a trojan or not. Detailed explanation regarding trojan detection in network traffic will be explained in the next point.

The dataset was obtained based on several raw data export processes. The raw data was obtained from Kamila Babeyava from Stratosphere Laboratory and the raw data was named Android Mischief Dataset v1 [**17**]. The data taken by the researcher focuses on pcap data or data that contains the network traffic recordings. In the Mischief documentation, the researcher explains and provides in detail about the dataset.

Before the dataset was formed, the pcap data was opened using the Wireshark application which was then carried out filtering IP based on Trojan IP and normal application IP. Figure 2 shows an overview on the dataset formation process.

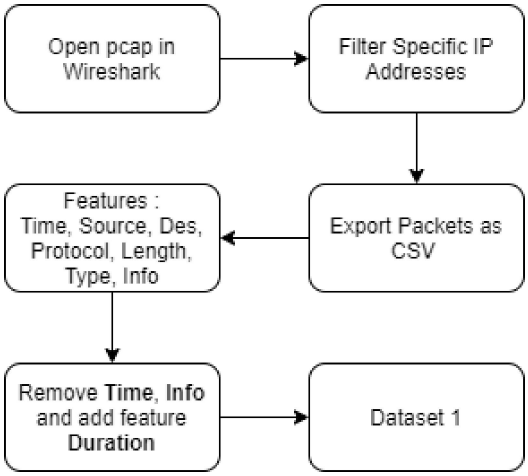


Figure 2. Chart of Formation of Dataset 1

IP filtering was done based on the IP address contained in the Mischief dataset documentation, for example filtering pcap data using client IP and server trojan "ip.addr == 10.8.0.57 && ip.addr == 104.244.42.2" then displayed packets only at the IP.

The Table 1 is a summary of the six features or attributes of the first dataset produced and used in this study.

Tabel 1. Summary of Attributes of Dataset 1

Name	**Description**
Source	The port number of the packet source
Destination	The port number of the destination packet
Protocol	The protocol type used by packets
Length	The data size generated by the packet (Byte)
Type	Type of packet in or packet out
Duration	The duration or distance between the previous packet to the next packet (Second)

In addition to the first dataset, the researcher also made a second dataset as test data refers to Ref. 1 which uses 20 data packages into one data line. The difference between dataset 1 and 2 can be seen from the attributes and data formation process, where dataset 1 comes from network traffic data

extraction while dataset 2 comes from extracting dataset 1 by creating new attributes. For example, the difference is the absence of port, protocol and type attributes that used in network traffic packets in dataset 2. The following chart is an overview of the formation of second dataset.

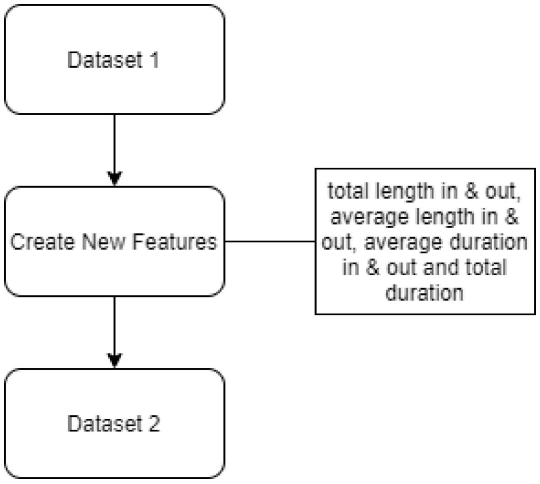


Figure 3. Chart of Formation of Dataset 2

This second dataset has seven attributes that will be used, here is a summary of the attributes used:

Table 2. Summary of Attributes of Dataset 2

Name	**Description**
Total Length Out	The amount of data size generated by packets out of 20 mixed packets (Byte)
Total Length In	The amount of data size generated by incoming packets of 20 mixed packets (Byte)
Avg. Packet Length Out	The average data size generated by packets out of 20 mixed packets (Byte)
Avg. Packet Length In	The average data size generated by incoming packets of 20 mixed packets (Byte)
Avg. Duration Out	The average duration required by packets out of 20 mixed packets (Second)
Avg. Duration In	The average duration required by incoming packets from 20 mixed packets (Second)
Total Duration	The total duration required by 20 mixed packets (Second)

Table 3. Dataset Distribution

	Dataset 1	Dataset 2
Training	14400	5400
Testing	3600	600
Total	18000	6000

Based on the literature, the author conducted several processes that will be carried out to detect Trojans in network traffic records using machine learning, it is included preprocessing, model design and model training.

2.1. Preprocessing

The data model that will be used must go through the formation stage of a raw dataset in the form of a pcap. Then it is collected into csv data consequently that machine learning is easy to process. The data formation process is taken manually and documented into a dataset. The dataset is divided into 2 types that called dataset 1 and 2.

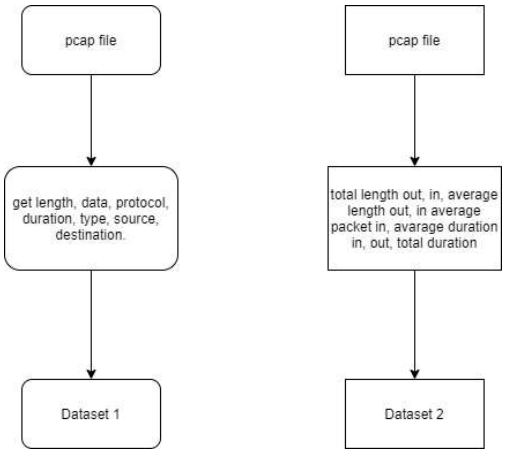
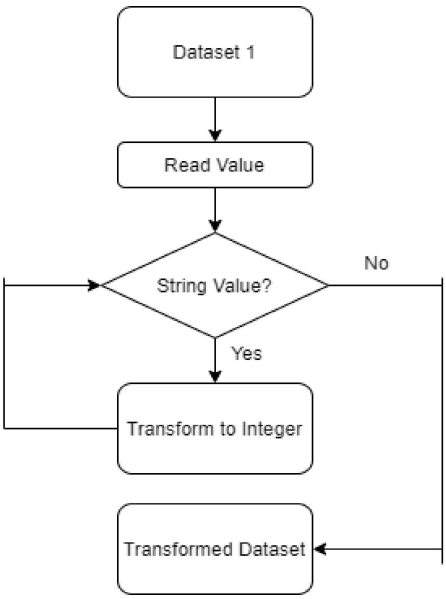


Figure 4. Dataset Formation

After the dataset is formed, the process of transforming the attribute value from string to integer in dataset 1 is carried out, which is then separated into two, those are training and testing data. Dataset 2 does not go through the transformation stage because the data form is already integrated. The following is an illustration of how this process occurs.

Figure 5. Chart of Transformation of Dataset 1



The set is divided into two, as training and testing data. The training data is used to introduce machine learners to the shape and attributes of the dataset, and the testing data is used as test material from the results of machine learning. Here is a chart regarding the distribution of training and testing data.

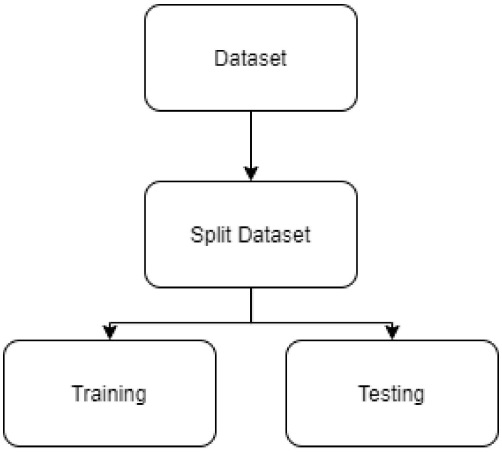


Figure 6. Chart of Test Data Formation

2.2. Model Design

Input from the model is preprocessing data that becomes a dataset. The two models are formed in this study has several attributes, those are seven attributes in dataset 1 and seven attributes in dataset 2 which can be seen in the image above. The following is an overview or plan of how the modelling will look like.

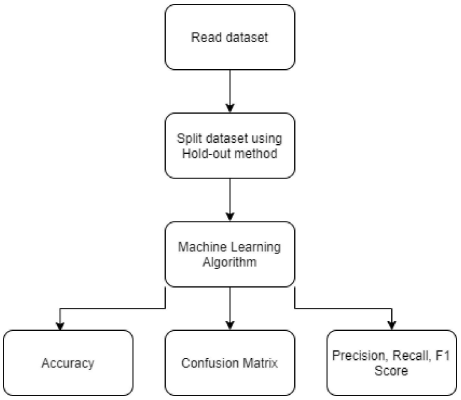


Figure 7. Chart of Model Design

2.3. Model Training

After the model is designed, the researcher tests the data with a machine learning algorithm to classify the data. The algorithm used are the Decision Tree, Random Forest and Naïve Bayes. Researcher uses these three machine learning algorithms because these algorithms were popular classification algorithms used in similar studies and the results show that these algorithms produce respectable accuracy values in testing to detect trojans. Machine learning can be used to automatically discover the rules by analyzing the data, and then the rules can be used to predict unknown data [**12**].

3. Result

In the testing process, researcher uses a dataset generated by six network traffic records for Remote Access Trojans and three normal applications. The Trojan used in this study is a well-known application type of Trojan. The network traffic generated by 3 normal applications is also one of the largest service providers on the internet. The performance of network traffic based malware detection mechanisms relies on selected traffic features for distinguishing between benign and malicious traffic [**13**]. The following are Trojans and normal applications used in this study:

Table 4. Types of Trojan and Normal Applications

No	Name	Type
1	AhMyth	Trojan
2	AndroidRAT	Trojan
3	AndroidTester	Trojan
4	DroidJack	Trojan
5	HawkShaw	Trojan
6	SpyMax	Trojan
7	Google	Benign
8	Facebook	Benign
9	Twitter	Benign

Testing was carried out based on the design of the model described in the Methodology. The first dataset was tested using three machine learning algorithms, Decision Tree, Random Forest and Naïve Bayes. The three algorithms are popular algorithms that are used to classify and in previous studies.

The k-Fold Validation is often used to validate classification results in machine learning tests. Accuracy, False Negative Rate (FNR) and False Positive Rate (FRP) were used to evaluate in this paper based on previous studies. Accuracy provides the correct classification number of normal and trojan packets in the dataset. False Negative Rate showed the proportion or distribution of negative classifications of trojan packages in the trojan dataset. False Positive Rate refers to the proportion of positive classification in the normal dataset. The smaller the FNR value when the accuracy is high, the better the detection system will not miss a malicious packet [**1**]. Equations 1-3 are used to calculate the accuracy value, FNR and FPR.

$Accuracy = TP+TN/TP+FP+FN+TN$	(1)
$False\ Negative\ Rate = FN/FN+TP$	(2)
$False\ Positive\ Rate = FP/FP+TN$	(3)

TP = True Positive TN = True Negative
FP = False Positive FN = False Negative

The attributes were generated based on the communication between the server and the client in the first dataset which was then tested based on the following illustration as an example of using the Decision Tree algorithm.

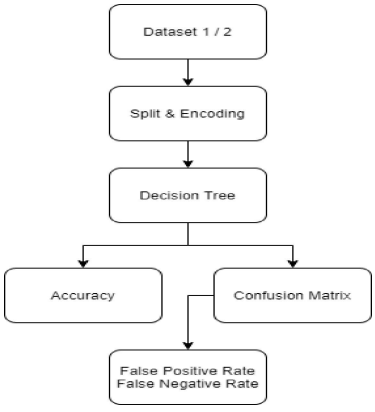


Figure 8. Example of Using Algorithm

Based on the accuracy formula, it can be calculated using the values contained in the confusion matrix, for example calculating the accuracy of dataset 1 with the normal number of packets 900 and trojan 90, obtained TP 90, FN 0, FP 1 and TN 899, then the calculations are as follows:

Accuracy = 90+899/90+1+0+899	*0.9989*

Then, it can be calculated all the data ratios that wanted to be tested and produced the following values:

Table 6. Value of Accuracy Dataset 1

Dataset 1	Accuracy		
	DT	RF	NB
N900-RAT90	0.9989	0.9990	0.1181
N1800-RAT90	0.9994	0.9989	0.0756
N1800-RAT1800	0.9997	0.9997	0.4691

Table 7. Value of FNR Dataset 1

Dataset 1	False Negative Rate		
	DT	RF	NB
N900-RAT90	0.0011	0.0011	0.9666
N1800-RAT90	0.0005	0.0011	0.9688
N1800-RAT1800	0.0005	0.0005	0.9688

Table 8. Value of FPR Dataset 1

Dataset 1	False Positive Rate		
	DT	RF	NB
N900-RAT90	0	0	0.0333
N1800-RAT90	0	0	0.0333
N1800-RAT1800	0	0	0.0927

Table 9 is the results of testing using dataset 2 in this research.

Table 9. Accuracy Value of Dataset 2

Dataset 2	Accuracy		
	DT	RF	NB
N150-RAT10	0.873	0.933	0.513
N300-RAT10	0.883	0.929	0.545
N300-RAT300	0.90	0.928	0.71

Table 10. Value of FNR Dataset 2

Dataset 2	False Negative Rate		
	DT	RF	NB
N150-RAT10	0.1428	0.0714	0.5214
N300-RAT10	0.12	0.0733	0.47
N300-RAT300	0.1	0.0833	0.47

Table 11. Value of FPR Dataset 2

Dataset 2	False Positive Rate		
	DT	RF	NB
N150-RAT10	0	0	0
N300-RAT10	0	0	0
N300-RAT300	0.1	0.06	0.11

The classification methods are included Decision Tree, Random Forest and Naïve Bayes. From the collected and tested dataset, there are 150 lines from normal applications and ten lines generated by trojans and so on, which are used to form the model. In the Decision Tree, the resulting accuracy is high with FNR 0.1428, 0.12 and 0.1 with increasing/decreasing FNR resulting in an effect on accuracy. In the Random Forest algorithm, the accuracy is not that far from the Decision Tree.

Based on the three algorithms, Decision Tree and Random Forest produce better accuracy, so it can be concluded that the two algorithms are respectable for classifying network traffic data generated by remote access trojans. Naive Bayes performed significantly worse than the other learners when using only network traffic-based features [**8**].

Tables 12-14 is the results of trials in previous researchers [**1**] which is a reference for comparison of the accuracy, false negative rate and false positive rate value.

Table 12. Accuracy Value of Previous Research

Dataset	Accuracy		
	DT	RF	NB
N150-RAT10	0.981	0.988	0.963
N300-RAT10	0.99	0.99	0.971
N300-RAT300	0.992	0.993	0.878

Table 13. Value of FNR of Previous Research

Dataset	False Negative Rate		
	DT	RF	NB
N150-RAT10	0.2	0.1	0.6
N300-RAT10	0.2	0.2	0.7
N300-RAT300	0.003	0.003	0.217

Table 14. Value of FPR of Previous Research

Dataset	False Positive Rate		
	DT	RF	NB
N150-RAT10	0.007	0.007	0
N300-RAT10	0.003	0.003	0.007
N300-RAT300	0.013	0.01	0.027

DT = Decision Tree, RF = Random Forest, NB = Naïve Bayes
N = Normal Packet, RAT = Trojan Packet

Based on the comparison of the accuracy value on this study and previous researches, the accuracy value between the Decision Tree and Random Forest algorithms has same value.

This study uses two types of datasets. The first aims to test the resulting datasets without carrying out any other processes. On the other hand, the second dataset is made based on several processes such as Feature Extraction for twenty packages which of course need more time to be processed before performing the classification test.

4. Conclusion

This paper proves that the classification using the previous research method and using the decision tree algorithm, random forest produces respectable accuracy. This algorithm gives the reason that machine learning algorithms are popular in classifying in this study. The attributes generated from the traffic network are also very well used for this test, with the comparison of the first dataset producing respectable accuracy and the second dataset which is the extraction of twenty network packets resulting in an accuracy of 90-92%. Dividing the number of ratios between normal and trojan packages used in model building based on reference [**1**], the greater the number of data ratios used also affects the accuracy for building the optimal model. Random forest is

the best algorithm in this test with an accuracy rate of 92.8% to 93.3%; FNR values of 0.0-0.08333 and FPR 0.0-0.06.

References

[1] Yin, K. S., & Khine, M. A. (2019). Optimal remote access trojans detection based on network behavior. International Journal of Electrical and Computer Engineering (IJECE), 9(3), 2177. doi:10.11591/ijece.v9i3.pp2177-2184

[2] Arora, A., Garg, S., & Peddoju, S. K. (2014). Malware Detection Using Network Traffic Analysis in Android Based Mobile Devices. 2014 Eighth International Conference on Next Generation Mobile Apps, Services and Technologies. doi:10.1109/ngmast.2014.57

[3] Arora, A., & Peddoju, S. K. (2017). Minimizing Network Traffic Features for Android Mobile Malware Detection. Proceedings of the 18th International Conference on Distributed Computing and Networking - ICDCN '17. doi:10.1145/3007748.3007763

[4] Cam, N. T., & Phuoc, N. C. (2016). NeSeDroid—Android Malware Detection Based on Network Traffic and Sensitive Resource Accessing. Proceedings of the International Conference on Data Engineering and Communication Technology Advances in Intelligent Systems and Computing, 19-30. doi:10.1007/978-981-10-1678-3_3

[5] Chen, Z., Yan, Q., Han, H., Wang, S., Peng, L., Wang, L., & Yang, B. (2018). Machine learning based mobile malware detection using highly imbalanced network traffic. Information Sciences, 433-434, 346-364. doi:10.1016/j.ins.2017.04.044

[6] Feng, J., Shen, L., Chen, Z., Wang, Y., & Li, H. (2020). A Two-Layer Deep Learning Method for Android Malware Detection Using Network Traffic. IEEE Access, 8, 125786-125796. doi:10.1109/access.2020.3008081

[7] Arora, A., & Peddoju, S. K. (2018). NTPDroid: A Hybrid Android Malware Detector Using Network Traffic and System Permissions. *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*. doi:10.1109/trustcom/bigdatase.2018.00115

[8] Jimenez, J. H., & Goseva-Popstojanova, K. (2019). Malware Detection Using Power Consumption and Network Traffic Data. *2019 2nd International Conference on Data Intelligence and Security (ICDIS)*. doi:10.1109/icdis.2019.00016

[9] Malik, J., & Kaushal, R. (2016). Credroid. *Proceedings of the 1st ACM Workshop on Privacy-Aware Mobile Computing - PAMCO '16*. doi:10.1145/2940343.2940348

[10] Rahmat, S., Niyaz, Q., Mathur, A., Sun, W., & Javaid, A. Y. (2019). Network Traffic-Based Hybrid Malware Detection for Smartphone and Traditional Networked Systems. *2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*. doi:10.1109/uemcon47517.2019.8992934

[11] Shabtai, A., Tenenboim-Chekina, L., Mimran, D., Rokach, L., Shapira, B., & Elovici, Y. (2014). Mobile malware detection through analysis of deviations in application network behavior. *Computers & Security, 43*, 1-18. doi:10.1016/j.cose.2014.02.009

[12] Wang, S., Chen, Z., Zhang, L., Yan, Q., Yang, B., Peng, L., & Jia, Z. (2016). TrafficAV: An effective and explainable detection of mobile malware behavior using network traffic. *2016 IEEE/ACM 24th International Symposium on Quality of Service* (IWQoS). doi:10.1109/iwqos.2016.7590446

[13] Wang, S., Yan, Q., Chen, Z., Yang, B., Zhao, C., & Conti, M. (2017). TextDroid: Semantics-based detection of mobile malware using network flows. *2017 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. doi:10.1109/infcomw.2017.8116346

[14] Zulkifli, A., Hamid, I. R., Shah, W. M., & Abdullah, Z. (2018). Android Malware Detection Based on Network Traffic Using Decision Tree Algorithm. *Advances in Intelligent Systems and Computing Recent Advances on Soft Computing and Data Mining,* 485-494.

doi:10.1007/978-3-319-72550-5_46

[15] Liu, A., Chen, Z., Wang, S., Peng, L., Zhao, C., & Shi, Y. (2018). A Fast and Effective Detection of Mobile Malware Behavior Using Network Traffic. *Algorithms and Architectures for Parallel Processing Lecture Notes in Computer Science,* 109-120. doi:10.1007/978-3-030-05063-4_

[16] Alqahtani, M. A. (2021). Machine Learning Techniques for Malware Detection with Challenges and Future Directions. *International Journal of Communication Networks and Information Security (IJCNIS),* 13(2), 258–270. doi:10.54039/ijcnis.v13i2.5047

[17] "Stratosphere Laboratory. Android Mischief Dataset v1. November 18th. Kamila Babayeva. https://stratosphereips.org/android-mischief-dataset"