DATA-DRIVEN MALWARE DETECTION SYSTEM FOR ANDROID PHONES

Ms. K. Anu, M.E, Assistant Professor of Computer Science Department,
Ms. Jenifer Grace D, Student of Computer Science Department,
Ms. Swetha P, Student of Computer Science Department,
St. Joseph College of Engineering, Sriperumbudur, Chennai

Abstract

Permission-based security model of Android restricts applications to access specific resources, but malicious applications can invade more easily in such user-centric pattern. Through the analysis of the Android Permission-based security model and the permission features of Android applications, we establish the permission model to quantify the functional characteristics of the application, and then provide an assessment method in which we use the network visualization techniques and clustering algorithm to determine whether the testing application is potentially malicious application or not so as to help users choose applications before installation. Security testing is known to be a notoriously difficult activity. This is partly because unlike functional testing that aims to show a software system complies with its specification, security testing is a form of negative testing, i.e., showing that a certain behaviour does not exist in the system.

1. Introduction

Machine Learning is the most popular technique of predicting the future or classifying information to help people in making necessary decisions. Machine Learning algorithms are trained over instances or examples through which they learn from past experiences and also analyse the historical data. Therefore, as it trains over the examples, again and again, it is able to identify patterns in order to make predictions about the future.

Data is the core backbone of machine learning algorithms. With the help of the historical data, we are able to create more data by training these machine learning algorithms. For example, Generative Adversarial Networks are an advanced concept of Machine Learning that learns from the historical images through which they are capable of generating more images. This is also applied towards speech and text synthesis. Therefore, Machine Learning has opened up a vast potential for data science applications.

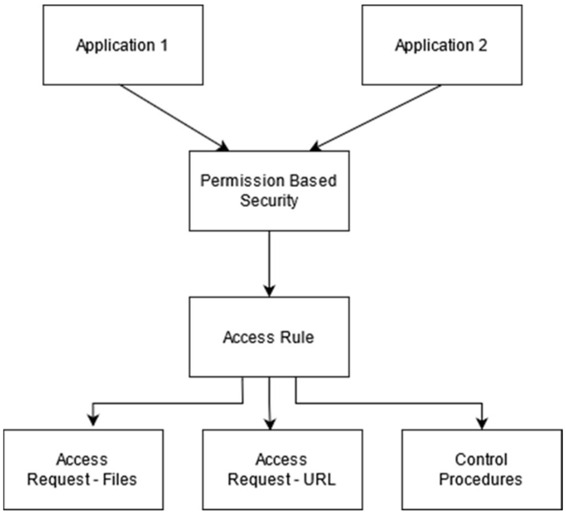
2. Literature Survey

Due to the changing behaviour of ransomware, traditional classification and detection techniques do not accurately detect new variants of ransomware. Attackers use polymorphic and metamorphic techniques to avoid detection of signature-based systems. We use machine learning classification to identify modified variants of ransomware based on their behaviour. To conduct our study, we used behavioural reports of 150 ransomware samples from 10 different ransomware families. Our data-set includes some of the newest ransomware samples available, providing an evaluation of the classification accuracy of machine learning algorithms on the current evolving status of ransomware. An iterative approach is used to identify optimum behavioural attributes used to achieve best classification accuracy. Two main parts of this study are identification of the behavioural attributes which can be used for optimal classification accuracy and classification of ransomware using machine learning algorithms. We have evaluated classification accuracy of three machine learning classification algorithms.

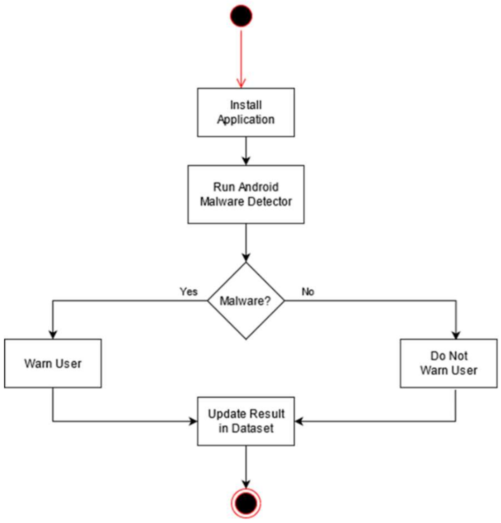
Development and dissemination of malicious software requires the creation of new methods for their detection. Therefore, we began to use proactive technologies that use the test program to detect the presence of certain symptoms, often occurring in malware. Dynamic analysis of the studied program launched for execution. There is a study of how the program interacts with the software environment that is read/write at certain registry keys, files, network activity the use of certain API calls. Due to the fact that studied the program is potentially harmful, to make its execution must be in an isolated environment. This paper discusses proactive methods based on API call analysis and propose a new method using a multiple sequence alignment to identify common traits in malware. The paper considers the scheme to detect malicious software, based on API calls, each of which is implemented in software. Also presented a completely new malware detection scheme based on multiple sequence API calls alignment. This scheme is described in detail and implemented in software. A test on a set of software and the legitimacy of the viral nature. Testing has shown that the established scheme of competitive shows and identifies malicious software with high accuracy.

3. System Design

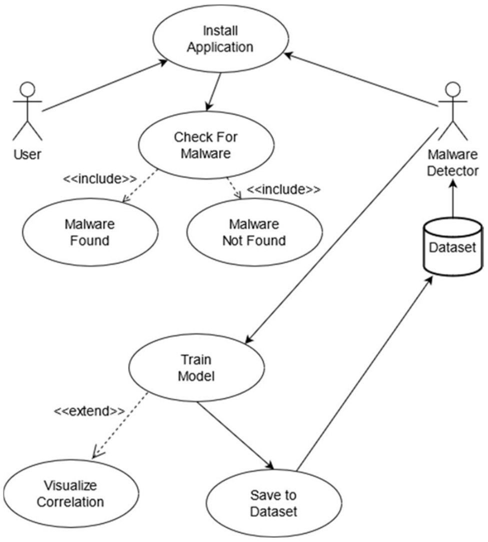
The existing system build a formal model of language-based technique in the human-assisted computer-based proof tool Coq using locally nameless representation. Furthermore, we demonstrate effectiveness of locally nameless representation in carrying out formal machine-readable proof of soundness of the language-based technique.



The proposed framework scales both in terms of code size and number of applications by leveraging the unprecedented computational power of cloud computing. The framework uses numerous heuristics and software analysis techniques to intelligently guide the generation of test cases aiming to boost the likelihood of discovering vulnerabilities. The proposed approach aims to answer the following overarching question: Given advanced software testing techniques and ample processing power, what software security vulnerabilities could be uncovered automatically?



In the proposed approach, a user cannot use a policy of choose one and reject other i.e., the user cannot selectively accept few permissions, while rejecting others in order to install the app. Among the list of all the permissions an app might call for a fishy permission to access some resource among the other valid permissions. The app will still be able to confirm the successful, though insecure, installation.



4. Implementation

Exploratory Data Analysis (EDA) is the first step in your data analysis process. Here, you make sense of the data you have and then figure out what questions you want to ask and how to frame them, as well as how best to manipulate your available data sources to get the answers you need. You do this by taking a broad look at patterns, trends, outliers, unexpected results and so on in your existing data, using visual and quantitative methods to get a sense of the story this tells.

EDA is performed in order to define and refine the selection of feature variables that will be used for machine learning. Once data scientists become familiar with the data set, they often have to return to feature engineering step, since the initial features may turn out not to be serving their intended purpose. Once the EDA stage is complete, data scientists get a firm feature set they need for supervised and unsupervised machine learning.

Sometimes you may find some data are missing in the dataset. We need to be equipped to handle the problem when we come across them. Obviously, you could remove the entire line of data but what if you are unknowingly removing crucial information? Of course, we would not want to do that. One of the most common idea to handle the problem is to take a mean of all the values of the same column and have it to replace the missing data.

The library that we are going to use for the task is called Scikit Learn pre-processing. It contains a class called Imputer which will help us take care of the missing data.

Now we need to split our dataset into two sets — a Training set and a Test set. We will train our machine learning models on our training set, i.e., our machine learning models will try to understand any correlations in our training set and then we will test the models on our test set to check how accurately it can predict. A general rule of the thumb is to allocate 80% of the dataset to training set and the remaining 20% to test set. For this task, we will import test_train_split from model_selection library of scikit.

dynamics that are conditioned on the expression.

Filter methods are generally used as a preprocessing step. The selection of features is independent of any machine learning algorithms. Instead, features are selected on the basis of their scores in various statistical tests for their correlation with the outcome variable. The correlation is a subjective term here.

5. Conclusion

The proposed system provides an assessment method in accordance with certain logic, reliable theoretical basis and existing research results in order to help users estimate the applications before installation. Via evaluation and statistical analysis, the assessment method can identify a part of malicious applications, mainly for those that apply for far too much permission or have relatively large differences with other applications which belong to the same category.

The proposed system provides an elevated level of security than the default security level provided by Android. The user is protected from installing and using malicious applications in their devices. Specifically, the user is made aware of the permission that are requested by the applications.

6. References

[1]: Hajredin Daku, Pavol Zavarsky, Yasir Malik, “Behavioral-Based Classification and Identification of Ransomware Variants Using Machine Learning”, 2018 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, pp 1-5, 2018.

[2]: Olga Hachinyan, “Detection of Malicious Software on Based on Multiple Equations of API-calls Sequences”, 2017 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus), pp 1-4, 2017.

[3]: Krzysztof Cabaj, Piotr Gawkowski, Konrad Grochowski, Alexis Nowikowski, Piotr ˙ Zórawski, “The impact of malware evolution on the analysis methods and infrastructure”, 2017 Federated Conference on Computer Science and Information Systems (FedCSIS), pp 1-4, 2017.

[4]: B. Fisseha Demissie, D. Ghio, M. Ceccato, A. AvanciniIdentifying Android Inter-App Communication Vulnerabilities Using Static and Dynamic Analysis, IEEE/ACM International Conference on Mobile Software Engineering and Systems (MOBILESoft), 2016.