DETECTION OF MALICIOUS BEHAVIOR IN ANDROID APPS AND MOBILE WEBPAGES

Ms. Nandini M. Kekare[1] , Mr.Sameer B. Patil[2]
[1,2] Department of Computer science & Engineering KIT's College of Engineering Kolhapur Maharashtra, India.

Abstract: This paper is a survey of " Detection of Malicious Behavior in Android Apps And Mobile Webpages". Current mobile malware detection and analysis technologies are "still imperfect, ineffective and incomprehensive"[3]. When an android app is installed or opened, many intrusions or attacks have occurred. To achieve this goal we developed an application to examine and analyze the apps using a tool. To test massive samples so as to collect dynamic app behavior records. So that we can predict the malicious API calls also, we aim to develop an application which will inspect a mobile URL and predict its a malware or benign. We aim to implement SVM (support vector machine) for the prediction.

Keywords:*Mobile webpages, web browser,Android malware-AMD_Data, real-time detection, dynamic behavior analysis, permission use analysis, Classification algorithm.*

1. INTRODUCTION

In recent times inspite of numerous anti-malware softwares are available still it is observed that the malware apps are on rise. Android devices are more vulnarable due to the presence of countless apps developed on the platform and the users granting the permissions for the installation of apps.An action of rummaging android its reveal more system permissions therefore it produces more heroism hazards. "Content" "functionality" and "layout" has implemented static analysis to resolve maliciousness in the Android [4], [5], [6] Features. In order to detect such malware app, considering their dynamic behavior –viz the calls to the API and functions. We have to propose a novel idea to dynamically trace the API calls and create a munky runner solution, which will effectively curb the presence of malware apps. Like malicious apps, the malicious web pages on mobile devices is also on the climb. The normal desktop anti-malware techniques cannot be applied for the mobile devices; due to multiple redirections and internal calls to mobile API'S. In the literature, existing malware discover modus operandi of different approaches while collecting features: static and dynamic [2]. In the paper, "Real-time Detection of Malicious Behavior in Android Apps" [2], author has implemented a changing behavior review and examination composition for malware operation discover in Android apps.

Static Analysis- Static analysiseach apk is statically extracted using dex2jar file and converted to bytecode and further extracting the source code of the apk.The manifest files is read for retriving the permissions and the source is studied for potential malicious function calls. Analysis is in order of introduced by "Enck et al." [7].

Dynamic analysis-Only looking for a API call which uses a similar set of function calls can't classify a apk as malicious as attackers tend to use different function calls in their apps.

In the paper, "Detecting Mobile Malicious Webpages in Real Time" [1], author presents kAYO, a speedy and trustworthy static analysis method to recognize malware web page.

Hence we propose a system and a plugin specifically for the mobile devices, which scans the content of mobile web pages and the calls to the API, which can detect the malware pages effectively.

2. OBJECTIVE

It is very essential to protect user privacy, data security, authority, confedential data for malicious webpages and android apps.

It having 3 main Modules-

Phase1. Mobile Malware WebPages Detection In Real Time

Phase 1.1 Scrap web data- In this phase, collect the webpages that is malicious webpages and benign webpages. So input of this phase is most visited websites in mobile and malware wepages. Crawl this webpages and store it. So output will be two datastores of unknown data.

Phase 1.2 **Extraction of feature set-** In this phase, output of previous phase is used as an input. Apply various feature set in our data and produce output for which page is clean which is malware.

Phase 1.3 **Classification of malware-** above phase of feature set or training dataset is used as input in this phase. So the features set is used to classify the web page as malware or benign.

Phase2 Mobile Malware Apps Detection In Real Time

Phase 2.1 Extract APK- To collect malicious and benign apps as an input and process them for extraction of java code. And output will be two sets of java files.

Phase 2.2 Dynamic Feature Extraction- In this phase, output of previous phase is used as an input. So process of them which is collected java files is extractions of permission and sensitive behavior sequence. Output will be of this phase is dynamic feature extration.

Phase 2.3 Malicious Behavior Recognition- Input as an known patters wich is malicious and benign and unknown apps. So process of them for feature extraction of unknown app. And output will be classify whether it is benign or malicious.

Phase 3. Analysis- We will check the static way of testing in all codes of apps and in depth survey of the probable functioning of apps as against the dynamic checking of app by calling the function through munky runner.

3. METHODOLOGY

Detection Methods Of Android Malware

3.1Key Extractions of feature Set

Feature set are divided into four classes.

Category	Features
Mobile specific	"# tel:, sms:, smsto:, mms:, mmsto:, geolocation, # of ipa"
JavaScript Features	"JS, embedded JS, noscript, internal JS, external JS, embedded JS"
HTM Features	"images;# of cookies, secure and HTTPOnly cookies, iframes, SSL, % of white spaces"
URL Features	"# of ." ,"?" ,"_" ,"," ," ;" ,"=" ,"-","&","%","0","1","2","3","4","5","6","7","8","9","\\

Table1. Accentuation of feature set

Implementation And Evaluation

3.2 Data Collection- The alexa top webpages, domaintyper.com webpages are considered for clean webpages malware domains list are considered for malware webpages.

The list of webpages are crawled and stored in folders. We created two folders such as first one is clean page and second one for malware webpage folder. This two folders are created which respectively store the clean and malicious url data.

3.3Classification of features-

Each file is parsed to search for corresponding features. The output obtained is the feature set and store it into two text files which is we have created. That is testout.txt file and another is testurlfeature.txt file.

Testurlfeature.txt file are used for stored the result of url featureset in this file. And testout.txt file are used for stored the result of mobile specific feature, JavaScript feature and HTML feature.

The count of each feature in each file is obtained and can be classified to the malware or clean.

1. We have collected webdata for clean webpages and malicious webpages.
2. Then stored these webpages in two folders. So we have created two folders, first one is clean folder for stored clean webpages and second one for malicious folder for stored malicious webpages.
3. Next extracted various feature sets.
4. We have used mobile specific feature set, HTML feature set, JavaScript feature set, URL feture set in my project.
5. Then count all files which is present in created folder and stored the results in two text files created.
6. Next classify those data which is clean or malicious by using feature set.

3.4 Model Choice And Implementation-

We think about each familiar clean android url's as a denial specimen and each familiar malwareandroid url as a definite specimen.

"Support Vector Machines" (SVM)- It is a well known binary classifier. In the analysis first, we build a ideal file from the "primary dataset" for classification ("C-SVC, nu-SVC"). We are efficient to bring the training dataset in an application, then selected the way to save into created model file and selected the suitable SVM . Forecasting purpose this standard file for later used.

4. PROPOSED ARCHITECTURE AND COMPARATIVE RESULT

Proposed Architecture

Figure 1 shows the proposed architecture for process for analysing webpages in android device. A user opens the URL he requires to meet in the extension browser. "from the webpage when mobile url's open, it selects key features. This feature set is input to our trained model. This input is having input to our trained model. This input which examines the webpage as malicious or benign. Then the output is sent back to the user's browser in real-time. When URL is clean in our model thenit renders the intended webpage in the browser automatically. Otherwise, it shows malware message to the user".

Comparative Results

Existing system extracted 33 key features in KAYO which divided into desktop and mobile datasets[1] but in our system, we used 40 key feature set specifically for mobile in our model.
The existing system achieved accuracy result is 40% when variables collected from KAYO model relating 33 features, were implemented to the mobile dataset[1].
In our model, using 40 features, we obtained 60% accuracy when key features received from our model which implemented to the mobile dataset.
We tested the 322 URLs,handle them manually in mobile browser. We observe 120 URLs to be malware by using SVM classification.

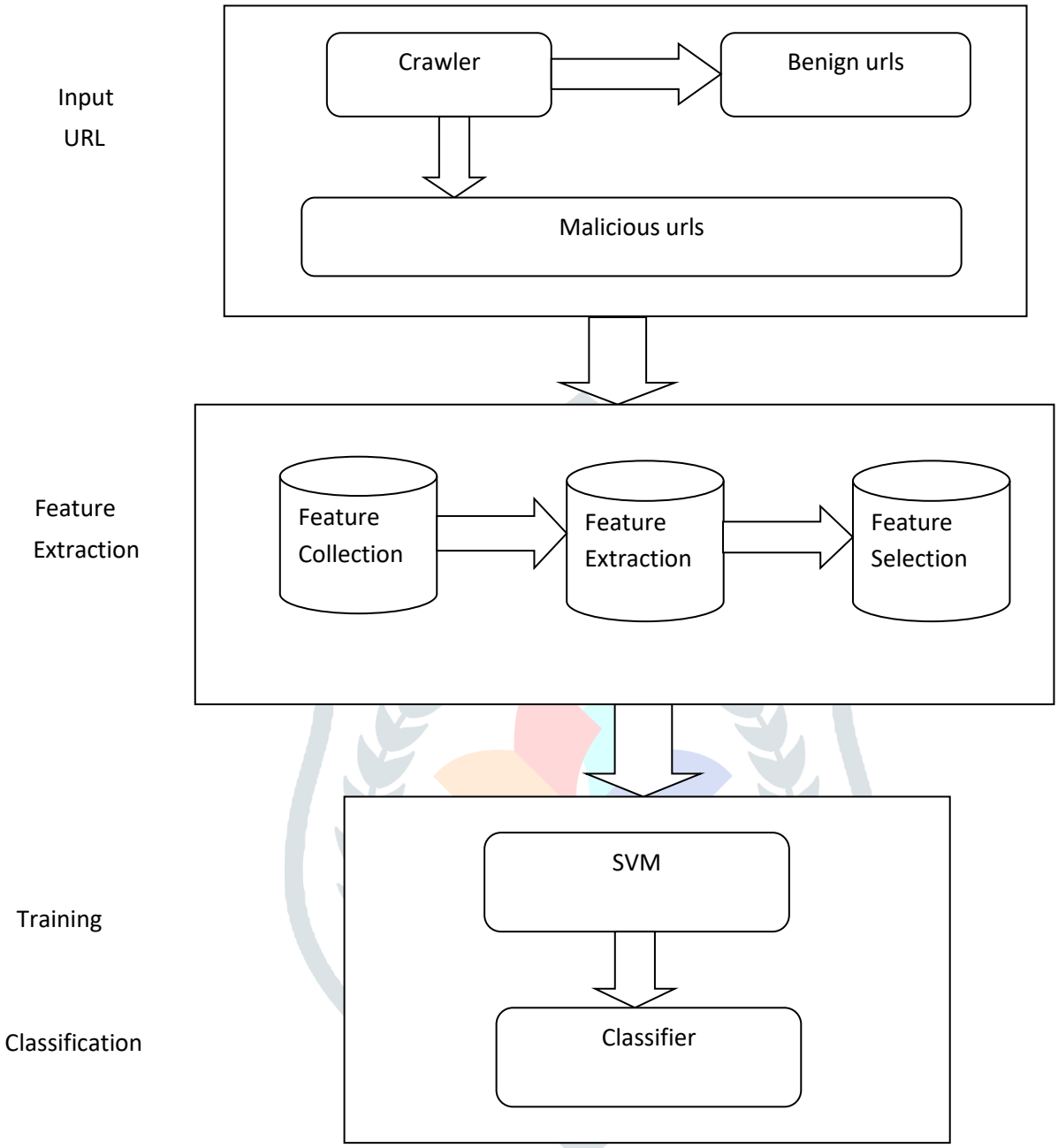

Fig1. Process for analysing webpages in android device.

5. CONCLUSION

In this paper, proposed system extracted 40 features set of url, mobile specific, html pages and javascript webpages. Trained the features against 322 urls including malware and benign. The algorithm used to train the data and test by using SVM (support vector machine). The live url insert is implemented further to test and predict whether the url is benign or malicious.

In our future work, we are going to develop a system using muncky runner to analyse the append their API calls. These calls again will serve as a traning data for the prediction of malware apk.

6. REFERENCES

[1] "C. Amrutkar, Y. S. Kim, and P. Traynor, Detecting Mobile Malicious Web pages in Real Time**",** IEEE Transactions on Mobile Computing, 2017**.**

[2] "Z. Ni, M. Yang, Z. Ling, J. Wu, J. Luo, Real-time Detection of Malicious Behavior in Android Apps, 2016 International Conference on Advanced Cloud and Big Data"

[3]"P. Yan, Z. Yan. "A Survey on Dynamic Mobile Malware Detection"Published in 13 May 2017 Springer Science+Business Media New York 2017"

[4] "D. Canali, M. Cova, G. Vigna, and C. Kruegel. Prophiler:a fast filter for the large-scale detection of malicious web pages. In Proceedings of the 20th International Conference on World Wide Web (WWW), 2011".

[5] "P. Likarish, E. Jung, and I. Jo. Obfuscated malicious javascript detection using classification techniques. In Proceedings of Malicious and Unwanted Software (MALWARE), 2009".

[6] "N. Provos, P. Mavrommatis, M. A. Rajab, and F. Monrose. All your iframes point to us. In Proceedings of the 17th USENIX conference on Security" (SECURITY), 2008".

[7] "Y. Yi et al., Dependency-based malware similarity comparison method," Journal of Software, vol. 10, no. 22, pp. 2439-2446, 2011".