NEXT GENERATION MALWARE DETECTION

Jalaj Pachouly[1], Prof. Varsha Dange[2]
[1]Student, Department of Computer Science, Dhole Patil College of Engineering Pune
[2]Professor, Department of Computer Science, Dhole Patil College of Engineering Pune

ABSTRACT

It is well known fact that there is a huge growth in Mobile computing, and at the same time. Android operating system became the preferred platform for developing the mobile applications, because of of Android being an Open source, free, Linux based with development in java language. Considering these facts Android also faces the most Malware attacks, in recent five years, with each new day, having a new capabilities. Traditional Malware detectors which are mostly based on Machine learning for Malware detection will not be enough to detect and prevent the Malware attacks, as their prominent assumption is not correct that Malware's feature set is stationary, where as in reality there is a huge population drift between the Malware family. To effectively handle the huge Malware growth and Population drift problem we need a Next Generation Malware Detector which is Context Aware, Adaptive, Scalable which we can abbreviate like "CASANDRA". Having the capability of being context aware help Malware detector to prevent reporting false positive by making rational difference between malicious versus genuine application. It also helps to increase the prediction to be more accurate. CASANDRA uses Contextual Weisfeiler -Lehman graph Kernal for Malware detection and uses the online learning for being Adaptive to cope up with the next generation Malware's. Apart of being Context aware and Scalable, it also deals with the problem of explain-ability, where it provides more information in a white box manner to user regrading why an application is labeled as Malware.

Key Terms: Population drift, Malware detection, on line learning, Context Aware, Graph

INTRODUCTION

Considering the Android a popular platform for mobile computing, day by day more applications are getting added. Considering the population growth in Mobile computing, there are new threats related to security and last five years has witnessed a huge growth in number of attacks and new Malware for mobile computing devices. As we know the trend of business like online sell, Mobile valet,Mobile banking , mobile computing has a great potential, and looks like in coming few years Mobile computing will completely take over to traditional desktop computing and laptops.

As mentioned above there is a great need for a security mechanism to ensure the safety and security of privacy, data breach and preventing financial losses which can occur if security mechanism is compromised.

What qualities should Malware detector posses?

Scalability : In general term scalability is an ability of a system to perform reasonably under increased load conditions. This is a general non function requirement, which is very important for any application to be successful. scalability property becomes important for Malware detector, because of two reasons

1- Huge Population growth for Mobile computing
2- Number of new Malware getting added every day in Mobile platform like android 3- Training data set for Malware detector is growing rapidly.

Hence if the Malware detector is not scalable, it will not be able to meet the desired outcome and

eventually will fail to protect the security mechanism.
Although it requires a considerable design consideration to make sure the scalability hooks are inbuilt in the system considering hardware and software both.
At glance system should be sclable both vertically and horizontally.

Adaptability : Adaptability is one of the complex aspect which Next generation Malware must posses. As we are seeing the huge population drift in the technology which is getting used in the recent Malware and new evasion technique getting evolved day by day, hence need a strong mechanism which will ensure that the Malware detector is capable of self learning and have a good pace to update the training data set in online fashion[8] and should not take adaption is a time consuming task, otherwise it will be difficult to respond to the new category of Malware in timely fashion. Hence new generation Malware detector should focus on Online learning and with each scan should become more adoptable for next set of detection. It should use a good combination of Machine Learning and Graph base behavioral analysis.

Efficiency : Efficiency in scanning an application is utmost important aspect, imagine if the Malware detector takes a huge time to white-list/blacklist an application, and if it consumes computing resources and memory in access, it will die its own death and will not see the face of production environment. Hence it is most desirable to keep the Malware detector light weight. It is most controversial fact that if we want to keep the Malware detector lightweight , then we may have to keep low content learning data on device or need to depend on advanced hardware deployed in cloud environment, to ensure that Malware detection is not eating up the majority of the resources in the end point mobile device. One other aspect is to use signature less technology which doesn't expect to keep high data contents on the device for Malware verification and analysis.

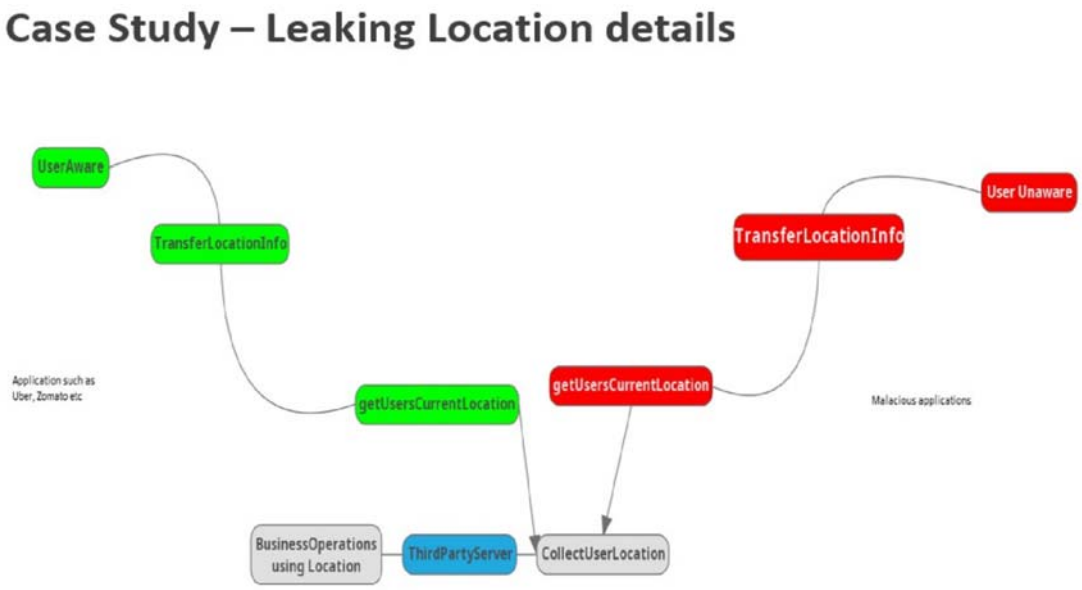
Accuracy : If the application has every other property but if it is not accurate it is seldom of any use.
Having said that, if the Malware detector is creating many False alarms , no one will respect and owner its alarm even if there is a true Malware detection. Hence desired expectation is that Malware detector need to ensure to generate accurate detection and reduce the false positive as much as possible. For avoiding false positive , it is important to make a clear distinction between genuine request of security sensitive event versus malicious. Being context aware is the key to reduce the false positive alarm.

Explain-ability : If the Malware detector is claiming an application as malicious then it should have a strong explanation to support its conclusion, it is not enough just to make an application is blacklisted without support matrix about the decision. To have a white box explanation, Malware detector should have proper auditing of application security sensitive activities with user context awareness and should be able to justify its decision to ensure it should not start blocking the benign applications.

Leaking Location analysis : It is interesting to see that in our Mobile device we may have two application which both want to access the users current location and then want to transmit the same information to some third party server. Transferring location information is a security sensitive operation hence Malware detector need to be smart enough when to prevent and when to allow or report illegal data or location information transfer from the Mobile device.

Malicious application: If an app is getting launched without user notice in a hidden way - like sending/receiving SMS or call. Now if this application connect to some other server, and start passing the user current location information.

Genuine application : To book a cab, in this case also app needs to transfer my location information to some server, to show the current cab's situation in the given user location, but in this case the user has initiated the transfer of location.
Although both apps are doing the same thing in terms of transferring the location info, one is malicious and one is genuine



User awareness:
Other than putting all responsibility to Malware detector, it is also important to educate the Mobile user to be aware of online frauds, few of the important asks could be :

1. Keep a watch on the permission asked by the applications
2. Track application based data usage
3. Use authenticated application from play store

Expectation from Malware detector:
Here are the few important expectation from the next generation Malware detector.

1. Grab the behavioral aspect of the application
2. To know and subsequently classify the security sensitive operation like storing and transferring sensitive data
3. Capturing behavioral data in multiple distributed devices and correlate to see the consistency check to be accurate
4. Collect the user context, or application execution context
5. should be able to classify the context as user aware or user unaware or unknown
6. Able to handle the large volume of data
7. should be optimized and self learn-able, hence can cope up with the population drift

Quantitative analysis for Malware detectors :
Following are the quality expected from a good Malware detector:

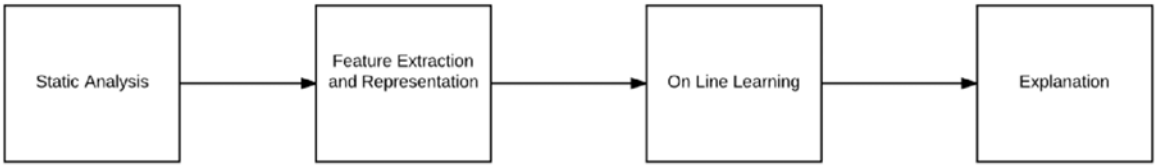
1. Accuracy of detection
2. Benchmark versus real world data set
3. Impact of contextual information on accuracy
4. Percentage reduction in false positive
5. Strategy for continuous learning new malware's
6. Online versus Batch
7. Time taken to adopt new malware feature
8. Justification for labeling app as Malware: Explain-ability
9. Metric such as reputation, behavior, execution context

Quality expectation from Malware detectors :

1. Fast enough to detect threats over older conventional systems combining state of the art technologies like
2. Machine learning
3. Graph-based malware analysis
4. Corrective action should be automatic process E.g recognition of network communication pattern with user context can quickly spot any malicious activity.
5. Detection should be signature less.
6. Signature based can be easily evaded by injecting junk code
7. Need to focus on various approach using other technologies like dependency graphs/semantic flow/behavioral flow etc.
8. Use of combination of technologies to correlate and see the attack features in various perspective(360 degree analysis)

9. Honeypots and deceptions can be used to extract the hidden features and behavior of application which can more accurately classify malicious applications
10. Special attentions should be given to non standard permission requested by the applications, and correlating all triggers to those resources
11. Rational security alarms low, medium, high
12. Adaptive in regards to malware population drift
13. Enrichment in learned training data set

High Level Architecture : To accomplish the mentioned facts of the Next Generation Malware, CASANDRA [1] is one the framework suggested, which tried to address the various quality attribute like Context awareness, Adaptability, Scalability.



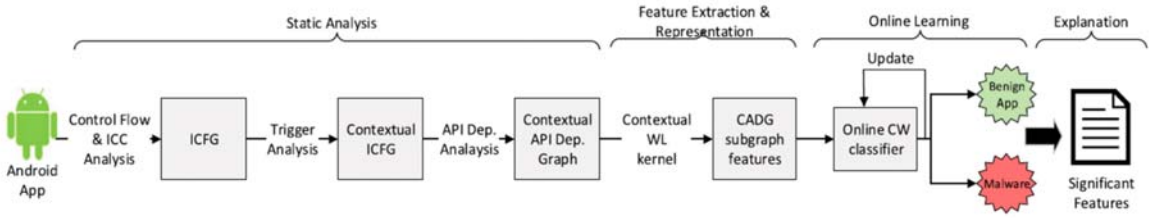
CWLK Major Steps

Static Analysis:
CASANDRA framework considers sub graphs from Contextual API Dependency Graphs (CADGs) as semantic features to perform malware detection. To this end, we first perform static program analysis to transform the given set of apps into their corresponding CADG representations.

Feature Extraction Representation:
Once the CAGD of the apps in the data set are constructed, those sub graphs which represent the security-sensitive events that happen in every app along with their context are extracted as using CWLK[1]. We follow a Bag-of-Features (BoF) model to construct the feature vector of individual apps.

Online Learning:
Once the feature vectors of all the apps in the training set are built, we train a CW classifier with them to detect malware. CW classifiers training and update procedures are explained in detail in Section III-D. Subsequent to this training CASANDRA is ready to perform malware detection at scale. It is important to note that the classifier is trained in an online fashion, meaning whenever a sample is presented, the classifier not only predicts its label but also updates the model based on the actual label of the sample.

Explanation:
ML based malware detection solutions are usually black-box methods as they do not explain why a particular sample is detected as malicious or benign. In CASANDRA, we address this shortcoming as follows: besides detection CASANDRA reports significant CADG subgraphs of an app that contribute to a detection. In most cases, these significant subgraphs reveal the app's characteristics related to malicious or benign behaviors.



Source of Events:

As mentioned earlier Events source plays an important role to trace the malicious activity versus genuine one. Events can be categorized as per below:

1- User Interactive : These events are generally interactive in nature and manually fired by the user to give any application specific access to any security sensitive operation, like while booking a cab, allowing permission to get the location access of Mobile device.

2- Anonymous : These are suspicious events and can easily fall in to Malicious category when exeuting any security sensitive operation without user aware context.

3- Application triggered : Some events are fired from one running application to other, but should not be treated as Malicious if the previous application is running in User Aware context and have legitimate permission of expected resources to complete its tasks.

REFERENCES

[1] Annamalai Narayanan, *Student Member, IEEE*, Mahinthan Chandramohan, *Student Member, IEEE*, Lihui Chen, *Senior Member, IEEE*, and Yang Liu, *Senior Member, IEEE "Context-Aware, Adaptive, and Scalable Android Malware Detection Through Online Learning"* IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTATIONAL INTELLIGENCE, VOL. 1, NO. 3, JUNE 2017

[2] A. Narayanan *et al.*, "Contextual Weisfeiler-Lehman graph kernel for malware detection," in *Proc. Int. Joint Conf. Neural Netw.*, 2016,pp. 4701–4708.

[3] N. Peiravian andX. Zhu, "Machine learning for androidmalware detection using permission and API calls," in *Proc. IEEE 25th Int. Conf. Tools Artif. Intell.*, 2013, pp. 300–305.

[4] K. W. Y. Au *et al.*, "PScout: Analyzing the android permission specification," in *Proc. ACM Conf. Comput. Commun. Secur.*, 2012, pp. 217–228.

[5] M. T. Ribeiro et al., "Why should i trust you?: Explaining the predictions of any classifier," in Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2016, pp. 1135–1144.

[6] M. M. Masud et al., "Cloud-based malware detection for evolving data streams," ACM Trans. Manage. Inf. Syst., vol. 2, no. 3, 2011, Art. no. 16.

[7] M. Xu et al., "Toward engineering a secure android ecosystem: A survey of existing techniques," ACM Comput. Surveys, vol. 49.2, 2016,Art. no. 38.

[8] A. Blum, On-Line Algorithms in Machine Learning. Berlin, Germany:Springer, 1998.

[9] Y. Aafer et al., "DroidAPIMiner: Mining API-level features for robust malware detection inAndroid," in Proc. Int.Conf. Secur. Privacy Commun. Syst., 2013, pp. 86–103

[10] K. W. Y. Au et al., "PScout: Analyzing the android permission specification," in Proc. ACM Conf. Comput. Commun. Secur., 2012, pp. 217–228.