ISSN: 2348 - 2117

International Journal of Engineering Technology and Computer Research (IJETCR)
Available Online at www.ijetcr.org
Volume 2; Issue 6; Page No. 71-75

Review on The Efficiency of Smartphone Anti-malware Against Transformation Attacks

Trupti D. Deshmukh[1], Vrunda K. Bhusari[2],

[1]JSPM's Bhivarabai Sawant Institute of Technology and Research, Wagholi Pune 412 207, India
deshmukh.trupti15@gmail.com

[2] JSPM's Bhivarabai Sawant Institute of Technology and Research, Wagholi Pune 412 207, India
vrundabhusari82@gmail.com

ABSTRACT
In this paper we have reviewed and analyzed different android anti-malwares. This is based on Information Forensics and security. Different transformation attacks are applied on anti-malwares. The term transformation is used to denote semantics preserving changes to program. We have reviewed Droid Chameleon technique which is applied to malware samples. By doing so the anti-malwares and their resistance to various common obfuscation techniques are analyzed. The possible remedies for improving the current state of malware detection on mobile devices can be found using a larger malware samples.

Key Words: mobile, malware, anti-malware, transformation.

INTRODUCTION:

Day by day the popularity of Smartphones and the Smartphone apps is increasing widely. These smartphones have advantage of user friendliness, ease of access within less time. But with these advantages smartphone OS i.e. Android is facing the problems related with security. The Android apps are getting attracted by malware attackers. There are many anti-malware softwares available in the market to protect the android apps against attacks. But with their presence the android platform is a "clearly today's target" for malware authors. Google Play offers different paid and free offerings for anti-malwares.

It can be observed that there are many malware threats attempting to break the security chains of android even though there is a huge security provided by different anti-malware softwares. It is necessary to check the anti-malwares and their vulnerability against different transformation attacks. To do so the term 'transformation' which refers to polymorphic changes is done on malware samples. It does not involve code-level changes. These transformed malwares are applied to the anti-malwares to check their efficacy. The technique called Droid Chameleon is used to conduct these common transformations. This technique transforms android applications automatically. That is the reason why we are trying to study the anti-malwares and their capability to defeat the attacks.

Generally, the efficacy of anti-malwares is concerned with following issues:

- Different Malwares and the way they are used for transformation attacks.
- How the anti-malwares are resistant to these transformation attacks.
- The probable ways to improve the efficiency of anti-malwares.

1. RELATED WORK:

A.OBFUSCATION TECHNIQUES

It is often that malware activity uses obfuscation technique to hide itself into the application. It thus compromises privacy by using some personal data such as financial or business importance.

a) ADAM

ADAM is an automated, generic, and extensible platform that evaluates the detection of Android malware detection systems. Fig 1. shows the design of ADAM.

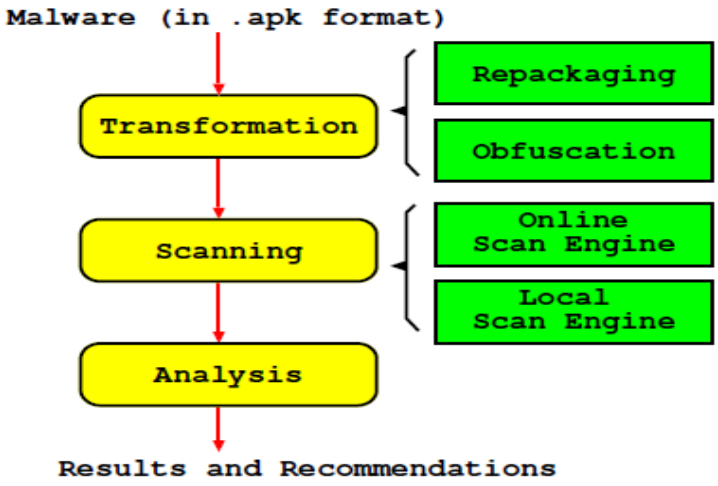

Figure 1: Design of ADAM

ADAM can automatically transform an original malware sample to different variants via repackaging and obfuscation techniques in order to evaluate the robustness of different anti-virus systems against malware mutation [1]. ADAM is designed by connecting different building blocks. These blocks are used to test different anti-viruses against malware samples.

1) Advantages-

1. It can be used for studies of very large-scale malware samples.
2. As transformation is done manually there is no need to apply manual modification of malwares. This results less overhead on codes.

2) Limitation-

ADAM is not always capable to avoid an anti-malware tool. It implements only some of transformations, such as renaming methods, introducing junk methods. It cannot be said that ADAM will always provide the better detection mechanisms.

b) Protection using various Obfuscation Techniques

The automatic code obfuscation is done to protect the messages [2]. This helps to preserve privacy policies between sender and receiver. Fig 2. shows how obfuscation technique provides the protection of messages between Alice and Bob. The source message is compiled and object code is created which is then obfuscated and passed to the server. Server then passes it to the client i.e. Bob. The obfuscated object code is then deobfuscated to object code which is decompiled to original source. Executer does the actual execution.

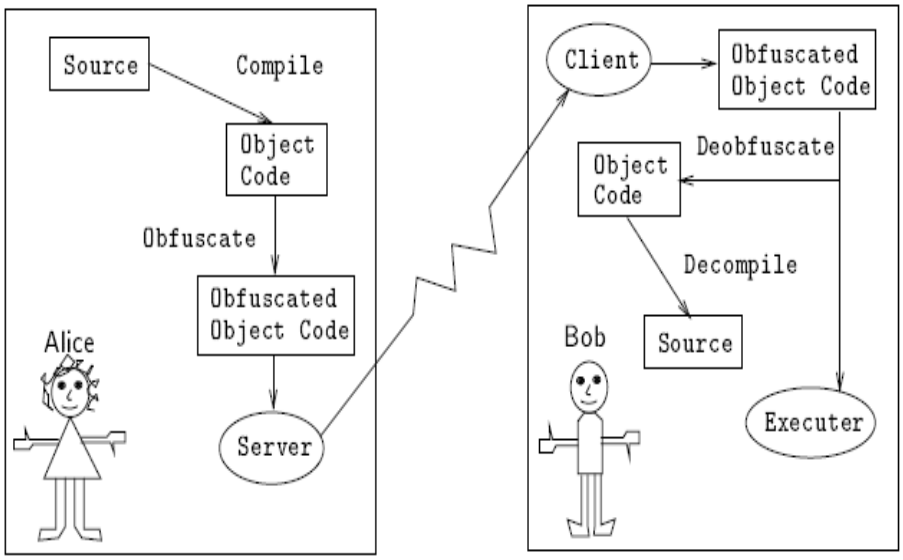

Figure 2: Protection through obfuscation

1) Advantages-
Obfuscation can be easily used to trace software pirates.
2) Limitations-
The obfuscated software remains secret until the powerful deobfuscator to be built. Therefore, there must be short time period between the releases of obfuscated software versions.

B. MALWARE DETECTION USING OBFUSCATION

a) Semantics-preserving Malware Detection

M. Christodorescu et al. [3] proposed malware detector that is used to find out the malicious behavior of a program. Most of times hackers use obfuscation to change the malwares. So, here the detectors use pattern-matching technique to search the obfuscations made by hackers.
1) Advantages-
1. It is totally syntax based technique. So, it is easy to be understood by detectors.
2. It has relatively low run time overhead.
2) Limitations-
It is mandatory to save the patterns of malicious instructions into templates which need to use of large databases.

b) Malware Specifications

Behavior- based techniques can be used to find out malicious behaviors [4]. This gives benefit for synthesizing near-optimal solutions. There is an automatic technique to extract and recognize optimally different specifications. It is used by behavior-based malware detector. It is based on graph mining and concept analysis.
1) Advantages-
1. It is used for large classes of programs due to its capability of probabilistic sampling of the specification space.
2. It is more accurate than commercial behavior-based detectors.
2) Limitation-
It requires more time delay i.e. near about 1-2 days. So, it is necessary to use multi-computing to improve this technique.

C. SMARTPHONE MALWARE RESEARCH

a) Automated Remote Repair for Malware

The malicious network traffic increases because of intruders. The problem can be solved by using Airmid, which is an automated system for remote remediation of mobile malware. After the detection of malicious traffic, the cellular network interacts with the source device to identify its originality of that traffic [5]. Fig 3. shows this approach.

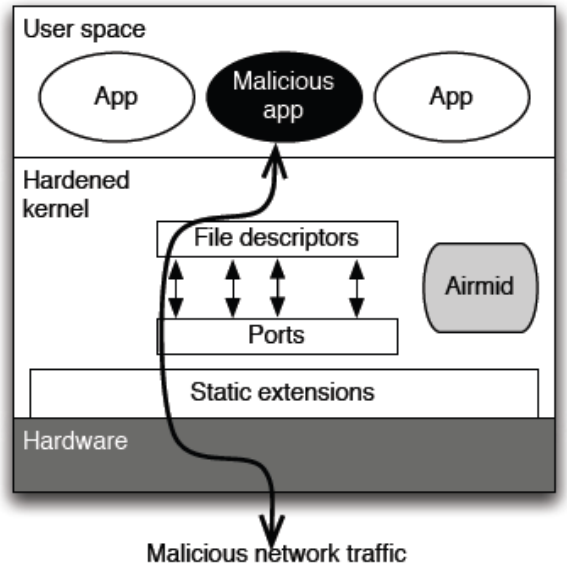

Figure 3: Malicious network traffic

1) Advantage-
It has very low overhead.
2) Limitations-
1. It does not tie to device and its security.
2. It is not able to characterize the traffic of large amount of malwares.

b) Automated Security Analysis of Smartphone Applications

To do the automation of security analysis the tool AppsPlayground is used. It integrates multiple components comprising different detection and automatic exploration techniques for this purpose [6].

The system can be evaluated using multiple large and small scale experiments involving real benign and malicious application.

1) Advantage-

It gives effective analysis even with large number of applications.

2) Limitation-

It is less effective at automatically detecting privacy leaks and malicious functionality in application

c) Detection of malicious apps in official and alternative Android markets

To find out malicious applications related to android permission based behavioral footprinting scheme is used. It is used for known malwares. Then a heuristics-based filtering scheme is applied to unknown malwares. This total system with known and unknown malicious families is called 'DroidRanger' [7].

1) Advantages-

1. It helps to focus on both official and unofficial Android markets for detecting malicious apps.
2. By using known and unknown malicious apps the detection proves to be scalable and efficient.

2) Limitation-

It needs rigorous policing process especially for unofficial marketplaces which is not satisfied by DroidRanger yet.

2. DROIDCHAMELEON FOR EVALUATING ANTI-MALWARE PRODUCTS:

V. Rastogi et al. [8] proposed DroidChemeleon a framework used to check the efficiency of anti-malwares. It does this by using different transformations on malware samples.

- Trivial Transformations: It uses different transformation categories such as repacking signed jar files, disassembling and reassembling the bytecode, changing package name.
- Transformation Attacks Detectable by Static Analysis: As per the name the transformations done here are can be detected by static analysis. The techniques used here are identifier renaming of class or method, data encoding in dex file, call indirections, code reordering in methods of program and encrypting payloads.
- Transformation Attacks Non-Detectable by Static Analysis: The encryption techniques which are not recognizable by static analysis are used here. Most commonly reflection of API and bytecode encryption is done to make code unavailable for static analysis.

Figure 4. shows how all these transofmrations are used.

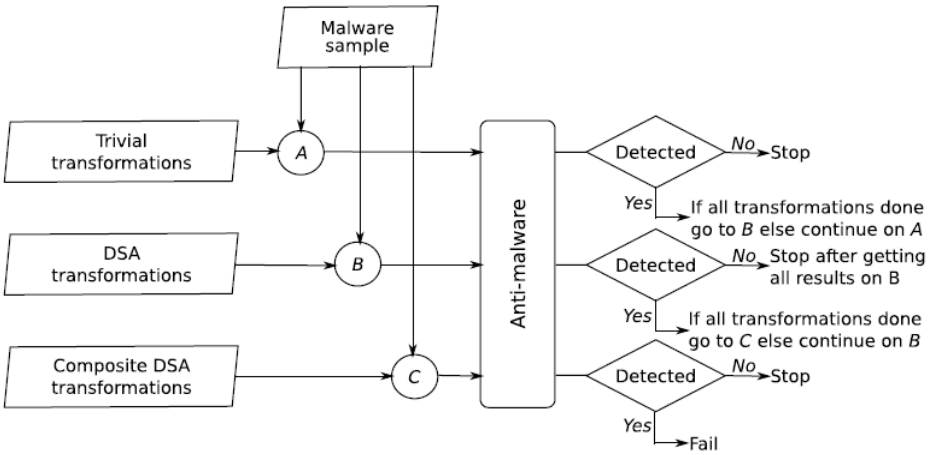

Figure 4: Evaluating anti-malware

Advantages-

1. There is no need to do the code level changes for applying transformations.
2. It can easily avoid the anti-malware tools. So, it is better than ADAM [1].

3. IMPORTANCE:

The android and its popularity causes to attract hackers. To prevent the smartphone from vulnerabilities and to provide the efficient way to use the android 'DroidChameleon' technique is used. It is able to apply more numbers of transformations to evaluate the efficiency of anti-malwares which will help to provide better security to users.

4. CONCLUSION:

In this paper, we have analyzed different anti-malwares that can be used for avoidance of different malware attacks. ADAM tool, obfuscation techniques can be used for privacy preserving but with fewer transformations.

Malware detectors that use obfuscation require pattern matching techniques. A framework based on DroidChameleon uses even more transformations using which more accurate efficiency of anti-malware tools can be found.

5. REFERENCES:

1. M. Zheng, P. Lee, and J. Lui, "ADAM: An automatic and extensible platform to stress test Android anti-virus systems," in *Proc. DIMVA*, Jul. 2012, pp. 1–20.
2. C. Collberg, C. Thomborson, and D. Low, "A taxonomy of obfuscating transformations," *Dept. Comput. Sci., Univ. Auckland, Auckland, New Zealand, Tech. Rep. 148, 1997.*
3. M. Christodorescu, S. Jha, S. Seshia, D. Song, and R. Bryant, "Semantics-aware malware detection," *in Proc. IEEE Symp. Security Privacy, May 2005, pp. 32-46.*
4. M. Fredrikson, S. Jha, M. Christodorescu, R. Sailer, and X. Yan, "Synthesizing near-optimal malware specifications from suspicious behaviors," *in Proc. IEEE Symp. SP, May 2010, pp. 45–60.*
5. Y. Nadji, J. Giffin, and P. Traynor, "Automated remote repair for mobile malware," *in Proc. 27th Annu. Comput. Security Appl. Conf., 2011, pp. 413-422.*
6. V. Rastogi, Y. Chen, and W. Enck, "AppsPlayground: Automatic security analysis of smartphone applications," *in Proc. ACM CODASPY, Feb. 2013, pp. 209–220.*
7. Y. Zhou, Z. Wang, W. Zhou, and X. Jiang, "Hey, you, get off of my market: Detecting malicious apps in official and alternative Android markets," *in Proc. 19th Netw. Distrib. Syst. Security Symp., 2012, pp. 1–13.*
8. Vaibhav Rastogi, Yan Chen, and Xuxian Jiang, "Catch Me If You Can: Evaluating Android Anti-Malware Against Transformation Attacks", *IEEE transactions on information forensics and security, VOL. 9, NO. 1, Jan 2014.*