Andro-profiler: Anti-malware System Based on Behavior Profiling of Mobile Malware

[Extended Abstract]

Jae-wook Jang
Korea University
changkr@korea.ac.kr

Jaesung Yun
Korea University
yjs8888@korea.ac.kr

Jiyoung Woo
Korea University
jywoo@korea.ac.kr

Huy Kang Kim*
Korea University
cenda@korea.ac.kr

Categories and Subject Descriptors

C.2.0 [**COMPUTER-COMMUNICATION NETWORKS**]: General—*Security and protection*

Keywords

Behavior profiling, Similarity, System call, Android, Malware

1. SUMMARY

The explosive diffusion of mobile devices running the Android platform has attracted the attention of hackers because sensitive information (e.g., phone number, SMS, banking information, and schedule information) are usually stored on mobile devices. Security for mobile devices then has become one of the most important challenge.

According to F-Secure's report [7], approximately fifty thousand new pieces of malware have been reported between January 2011 and January 2013. To react to much malware effectively and efficiently, malware analysts try to find unique behavior patterns of each malware family because malware creators make variants by reusing core codes. The unique behavior patterns can be represented by various symbols (e.g., permission set, API call, and system call). Many previous works have focused on finding unique behavior patterns and proposed various detection methods, permission-based, API call-based and system-call based. Permission-based detection methods are not efficient in classifying benign applications as benign since relevant rule sets merely focus on detecting the malware [6]. API call-based detection methods cannot generate distinct signatures until decompilation or disassembly processes have been completed [9]. System call-based detection methods can more accurately detect malicious behavior than other methods since it is impossible to modify original functionality of system calls; malware creators hardly disguise malicious behavior as normal behavior. However, proposed methods mainly dealt with frequency of system calls well found in malware [2, 3]. The number of invoked system calls is small, and most of the system calls used in malware (e.g., read(), write()) are mostly observed in both benign and malicious applications. We need to consider more features, such as arguments in the system call and network activities, for elaborate malware detection or classification.

To overcome the drawbacks in previous methods, we propose a novel anti-malware system based on behavior profiling called Andro-profiler. We exploit system calls, their arguments, and system logs (e.g., SMS, call, and network I/O) provided by Droidbox [5] as feature vectors; we define them as integrated system logs. Moreover, directly to infer behavior patterns via the system calls which are not readable, we make behavior profiling of malware represented by integrated system logs using the concept of behavior profiling described in [1].

Our proposed system parses the integrated system logs of malware, makes the behavior profile, and categorizes applications according to their behavior patterns. Our system computes the similarity score between the behavior profile of malicious application and representative behavior profile of each malware family, and then classifies the malicious application into the group with which it bears the most similarity. The similarity score is given by:

$$S = \sum_i w_i \cdot BFS_i \quad where \sum_i w_i = 1 \tag{1}$$

where BFS_i and w_i are the similarity and weight of behavior factor i, respectively. Similarity of behavior factor (BFS) is composed of four parts: similarity of sending premium-rate SMS (SS), that of calling premium-rate number (CS), that of collecting sensitive information (SIS), and that of converting data (CDS). We choose the weight (w_i) to be 0.33 for SS, 0.33 for CS, 0.21 for SIS, and 0.13 for CDS - we determined that the weight value to obtain the best performance through experiments. Table 1 shows similarity metrics for behavior factors. The representative behavior profile of each malware family has to depict the unique and common behavior patterns of each malware, then Andro-profiler chooses one of the methods updating the representative behavior profile as follows. The first updating method is Intersection (Profiler-INT). The representative behavior profile for each malware family is updated by the intersection of behavior profiles of members in each subgroup. The second updating method is Union (Profiler-UNI). The representative behavior profile for each malware family is updated by union of behavior profiles of members in each subgroup. In the

WWW'14 Companion, April 7–11, 2014, Seoul, Korea.
ACM 978-1-4503-2745-9/14/04.
http://dx.doi.org/10.1145/2567948.2579366.

Table 1: Similarity metric to apply to each behavior factor

Behavior factor	Behavior target	Example	Similarity metric
Sending SMS	Premium-rate	-	Binary (0 or 1)
Calling	Premium-rate	-	Binary (0 or 1)
Sending sensitive info.	System info.	IMEI, IMSI, Device ID	Jacard index [0, 1]
	Private info.	Storage contents, Location	
Converting data	Destination URL	http://localytics/Upload/	Modified levenshtein distance [0, 1]
	Encryption mode	DES, AES, Blowfish	Binary (0 or 1)
	Encoding mode	Gzip	Binary (0 or 1)

Table 2: Classification performance for 709 malware and 350 benign samples

Category		Accuracy			AUC		
		Profiler-		Crowdroid [3]	Profiler-		Crowdroid [3]
		INT	UNI		INT	UNI	
Malware (709)	AdWo (401)	**1.00**	**1.00**	0.54	**1.00**	**1.00**	0.60
	AirPush (60)	**0.95**	**0.95**	0.02	**0.99**	**0.99**	0.51
	Boxer (42)	**1.00**	**1.00**	0.43	**1.00**	**1.00**	0.66
	FakeBattScar (51)	**1.00**	**1.00**	0.18	**1.00**	**1.00**	0.58
	FakeNotify (59)	**1.00**	**1.00**	0.80	**1.00**	**1.00**	0.88
	GinMaster (96)	**1.00**	**1.00**	0.11	**1.00**	**1.00**	0.53
Benign (350)		**0.97**	**0.97**	0.35	**0.99**	**0.99**	0.63
Average		**0.99**	**0.99**	0.35	**0.99**	**0.99**	0.63

updating method of Profiler-UNI, as the members of each malware family increase, the representative behavior profiles increase.

2. RESULTS

For performance evaluation, 709 malware samples consisting of 7 malware families were collected from January 2013 to August 2013 through malware repositories such as virusshare [8], contagio [4], and 350 benign samples were collected through GooglePlay for same periods. Our performance evaluation focuses on the effectiveness and the efficiency of malware classification. We demonstrate that our system performs well in classifying malware families in Table 2. We used the accuracy and the area under the ROC curve (the AUC) as performance metrics.

The previous work similar to our approach is Crowdroid [3]. Crowdroid monitors invoked system calls and makes frequency table of system calls in client side. Crowdroid identifies malicious behavior, and detects malware utilizing K-means algorithm in server side. Andro-profiler performs well in detecting and classifying malware families with 99% classification accuracy on average, regardless of updating method, while Crowdroid has 35% classification accuracy on average. Our proposed methods also outperform 50% performance improvement than Crowdroid in terms of AUC. Moreover, our proposed system only took on average 55 seconds/MB to analyze a malicious application; we exclude the setup time for analysis such as booting time of the emulator. The majority of this time is spent in making the behavior profile; it takes 0.2 seconds to classify each target application.

3. REFERENCES

[1] U. Bayer, P. Comparetti, C. Hlauschek, C. Kruegel, and E. Kirda. Scalable, behavior-based malware clustering. In *Proceedings of the 16th Annual Network and Distributed System Security Symposium (NDSS '09)*, 2009.

[2] T. Blasing, L. Batyuk, A.-D. Schmidt, S. Camtepe, and S. Albayrak. An Android Application Sandbox system for suspicious software detection. In *Malicious and Unwanted Software (MALWARE), 2010 5th International Conference on*, pages 55–62, 2010.

[3] I. Burguera, U. Zurutuza, and S. Nadjm-Tehrani. Crowdroid: Behavior-based Malware Detection System for Android. In *Proceedings of the 1st ACM Workshop on Security and Privacy in Smartphones and Mobile Devices*, SPSM '11, pages 15–26, 2011.

[4] Contagio. Contagio mobile-mobile malware mini dump. http://contagiominidump.blogspot.kr/, 2013. Accessed Dec. 10, 2013.

[5] Droidbox. Droidbox - android application sandbox - google project hosting. http://code.google.com/p/droidbox/, 2013. Accessed Dec. 10, 2013.

[6] W. Enck, M. Ongtang, and P. McDaniel. On lightweight mobile phone application certification. In *Proceedings of the 16th ACM Conference on Computer and Communications Security*, CCS '09, pages 235–245, 2009.

[7] F-Secure. MOBILE THREAT REPORT Q4 2012. http://www.f-secure.com/static/doc/labs_global/Research/Mobile, 2013. Accessed Dec. 10, 2013.

[8] VirusShare. Virusshare.com-because sharing is caring. http://virusshare.com/, 2013. Accessed Dec. 10, 2013.

[9] 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 *Proceedings of the 19th Annual Network and Distributed System Security Symposium*, 2012.