Static Smartphone Malware Detection

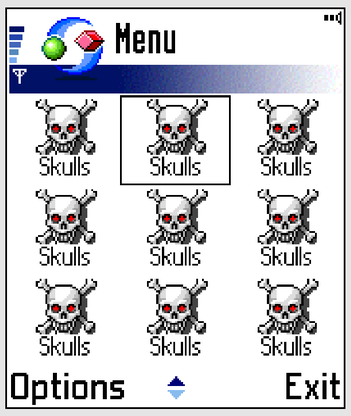
Aubrey-Derrick Schmidt, Ahmet Camtepe, and Prof. Dr.-Ing. Sahin Albayrak
<aubrey.schmidt, ahmet.camtepe, sahin.albayrak>@dai-labor.de
www.dai-labor.de Technische Universität Berlin

Static Analysis

Static anaylsis represents an approach of checking source code or compiled code of applications before it gets executed. Chess and McGraw state that static anaylsis promises to identify common coding problems automatically. While manual code checking is also a form of static analysis, software tools are used in most cases in order to perform the checks. Chess and McGraw additionaly claim that good static checkers can help to spot and eradicate common security bugs.

Smartphone Malware

Malwares (e.g. virus, worms and Trojan horses) have been threats to computer systems for many years and it was only a question of time when the first malicious software writers would get interested in increasingly popular mobile platforms, such as Symbian OS. In 2004, the first articles about malware for smartphones appeared, stating that the next generation of targets are mobile devices. Since then, the number of malwares increased every month and variants for various smartphone platforms appeared. .



Static Analysis for Malware Detection on Smartphones

In this context, the new smartphone platform Android gained special interest among developers. Since it set open source, security tools can be developed even at kernel level. This allows comprehensive security mechanism to be deployed on Android handsets only being limited by the typical resource constraints of mobile devices. Due to these constraints, we focus on static and light-weight mechanisms for detecting malware presence on Android devices. Our static approach for detecting malware allows us to use simple classifiers which are not very resource consuming and therefore fit very well to mobile needs. Additionally, these classifiers tend to have high detection rates while keeping false positive rate low. We compare Function call lists of benign software with malware executables for classifying them with PART, Prism and Nearest Neighbor Algorithms

Example Function Calls:
Symbol table '.dynsym' contains 104 entries:
Num: Value Size Type Bind Vis Ndx Name
0: 00000000 0 NOTYPE LOCAL DEFAULT UND
1: 00000000 622 FUNC GLOBAL DEFAULT UND abort@GLIBC_2.0 (2)
2: 00000000 29 FUNC GLOBAL DEFAULT UND __errno_location@GLIBC_2.0 (2)
3: 00000000 84 FUNC GLOBAL DEFAULT UND sigemptyset@GLIBC_2.0 (2)
4: 00000000 52 FUNC GLOBAL DEFAULT UND sprintf@GLIBC_2.0 (2)
5: 00000000 433 FUNC GLOBAL DEFAULT UND localeconv@GLIBC_2.2 (3)
6: 00000000 10 FUNC GLOBAL DEFAULT UND dirfd@GLIBC_2.0 (2)
7: 00000000 87 FUNC GLOBAL DEFAULT UND __cxa_atexit@GLIBC_2.1.3 (4)
[...]

Countermeasures

Commercially available countermeasures to smartphone malware suffer from weaknesses since they mostly rely on signatures. This approach leaves users exposed to new malware until the signature is available. Bulygin showed that in worst case a MMS worm targeting random phone book numbers can infect more than 700000 devices in about three hours. Additionally, Oberheide et al. state that the average time required for a signature-based anti-virus engine to become capable of detecting new threats is 48 days. These numbers request extended security measures for smartphones as a malware can seriously damage an infected device within seconds .

ACCURACY VALUES OF CLASSIFIERS ACCORDING TO ATTRIBUTE SETS

	relocation			dynamic			combined		
	mutual attributes								
Accur.	CC	DR	FP	CC	DR	FP	CC	DR	FP
Prism	0.78	0.70	0.00	n.V.	n.V	n.V.	0.78	0.70	0.00
PART	0.94	0.99	0.15	0.97	1.00	0.12	0.97	1.00	0.12
n. Nb	0.92	0.98	0.21	0.90	0.92	0.13	0.96	0.98	0.11
	all attributes								
Prism	0.81	0.76	0.00	0.83	0.76	0.00	0.83	0.77	0.00
PART	0.95	1.00	0.16	0.97	1.00	0.12	0.97	1.00	0.12
nNb	0.94	0.99	0.12	0.96	0.99	0.10	0.96	0.99	0.10

A.-D. Schmidt, R. Bye, H.-G. Schmidt, J. Clausen, O. Kiraz, K. Yüksel, A. Camtepe, and S. Albayrak. Static analysis of executables for collaborative malware detection on android. In *IEEE International Congress on Communication (ICC) 2009 - Communication and Information Systems Security Symposium*, 2009

A.-D. Schmidt, J. H. Clausen, S. A. Camtepe, and S. Albayrak. Detecting Symbian OS Malware through Static Function Call Analysis. In *Proceedings of the 4th IEEE International Conference on Malicious and Unwanted Software (Malware 2009)*, pages 15–22. IEEE, 2009

A.-D. Schmidt, H.-G. Schmidt, L. Batyuk, J. H. Clausen, S. A. Camtepe, S. Albayrak, and C. Yildizli. Smartphone malware evolution revisited: Android next target? In *Proceedings of the 4th IEEE International Conference on Malicious and Unwanted Software (Malware 2009)*, pages 1–7. 2009

A. Schmidt and S. Albayrak. Malicious software for smartphones. Technical Report TUB-DAI 02/08-01, Technische Universität Berlin, DAI-Labor, Feb. 2008. http://www.dai-labor.de

DAI-Labor
Distributed Artificial Intelligence Laboratory