Electronic Communications of the EASST
Volume 079 (2020)

Interactive Workshop on the Industrial Application of Verification and Testing ETAPS 2020 Workshop (InterAVT 2020)

Timed Automata for Mobile Ransomware Detection

Fabio Martinelli, Francesco Mercaldo and Antonella Santone

13 pages

Guest Editors: Stylianos Basagiannis, Goetz Botterweck, Anila Mjeda
ECEASST Home Page: http://www.easst.org/eceasst/ ISSN 1863-2122

Timed Automata for Mobile Ransomware Detection

Fabio Martinelli[1], **Francesco Mercaldo**[2,1] **and Antonella Santone**[2]

fabio.martinelli, francesco.mercaldo@unimol.it, antonella.santone@unimol.it
[1] Institute for Informatics and Telematics, National Research Council of Italy, Pisa, Italy
[2] Department of Medicine and Health Sciences "Vincenzo Tiberio", University of Molise, Campobasso, Italy

Abstract: Considering the plethora of private and sensitive information stored in smartphone and tablets, it is easy to understand the reason why attackers develop everyday more and more aggressive malicious payloads with the aim to exfiltrate our data. One of the last trend in mobile malware landascape is represented by the so-called ransomware, a threat capable to lock the user interface and to cipher the data of the mobile device under attack. In this paper we propose an approach to model an Android application in terms of timed automaton by considering system call traces i.e., performing a dynamic analysis. We obtain encouraging results in the experimental analysis we performed exploiting real-world (ransomware and legitimate) Android applications.

Keywords: ransomware, mobile, Android, timed automaton, formal methods, model checking, security

1 Introduction

Researchers discovered several cases of malware on the official Android app store, and while all of the dangerous apps were taken down rapidly, they still managed to infect thousands of devices worldwide[1].

However, even more, severe mobile malware hides in applications downloaded from shady Internet sources. For instance, the DoubleLocker, LokiBot, GhostClicker, and Sockbot malware families are just a few of examples that target Android mobile users daily.

The most worrying fact is that out of all 10 million shady Android apps, the majority (77%) turned out to be malicious. The remaining 23% are potentially unwanted applications[2].

Sophos security experts highlight that ransomware is a problem for Android users[3].

Mobile ransomware is a kind of malware afflicting mobile devices, for instance tablets and smartphones [ASKA16]. A cybercriminal can use mobile malware to steal sensitive data from a smartphone or lock a device, before demanding payment to return the data to the user or unlock the device.

After the malware is downloaded onto a device, it will show a fake message accusing the victim of unlawful engagement before encrypting files and locking the phone. After the payment

[1] https://www.techrepublic.com/article/malware-laden-apps-in-google-play-store-mine-cryptocurrency-from-mobile-victims/
[2] https://www.2-spyware.com/malware-forecast-for-2018-more-ransomware-android-mac-viruses
[3] https://www.sophos.com/en-us/en-us/medialibrary/PDFs/technical-papers/malware-forecast-2018.pdf?la=en

Timestamp	*System call*
t_1	*recv*
t_2	*recv*
t_3	*recv*
t_4	*read*
t_5	*futex*

Table 1: A system call trace fragment.

is processed, often via Bitcoin, the ransomware will send a code to unlock the phone or decrypt the data [IM07, QWR18].

Ransomware writers leverage users fear from loosing valuable data to obtain the payment of ransoms [MMMS18]. In particular, in the mobile device realm, as opposed to other threats that silently try to get into possession of data by copying them and leaking through network [CMN+18], ransomware denies access to data to the users either by encryption or simply by locking the device and scaring users (i.e., data are not encrypted, but the device is locked with a message that makes the user believe that data have been encrypted) [AMS18]. In some cases even paying the requested ransom does not guarantee that the access to the data will be restored [MNS16].

Usually Android malware writers modify some existing malware, merging together parts of different existing malicious codes [XZM+17]. This explains also why Android malware is usually grouped in families: in fact, given this way of generating Android malware, the malware belonging to the same family shares common parts of code and behaviors [MMM17].

In this paper an approach to model Android applications in terms of timed automata is proposed, with the aim to dynamically discriminate between legitimate and ransomware applications.

2 The Method

The proposed approach consists in two main phases: the *Formal Model Creation* and the *Formal Model Verification*.

The *Formal Model Creation* phase output is the formal model representing the Android application under analysis.

In this step a timed automaton from the system call traces is generated. Whether in the system call trace the same system call is repeated between consecutive temporal instances the automaton will contain a loop. To better understand how the proposed method build the automaton from the system calls let us consider the system call trace fragment depicted in 1

As shown from the example in Table 1, the system call traces exhibit for three consecutive timestamp the *recv* system call, subsequently it exhibits the *read* and the *futex* system calls.

From the system call trace frament in Table 1 in Figure 1 we show the relative automaton. The automaton generated from the *recv* system call will contain a loop for the t_1, t_2 and t_3 time intervals (whether it is repeated three times in the system call trace). The exit condition from the loops is guaranteed by a guard, while the entering one is guaranteed from an invariant. In detail,

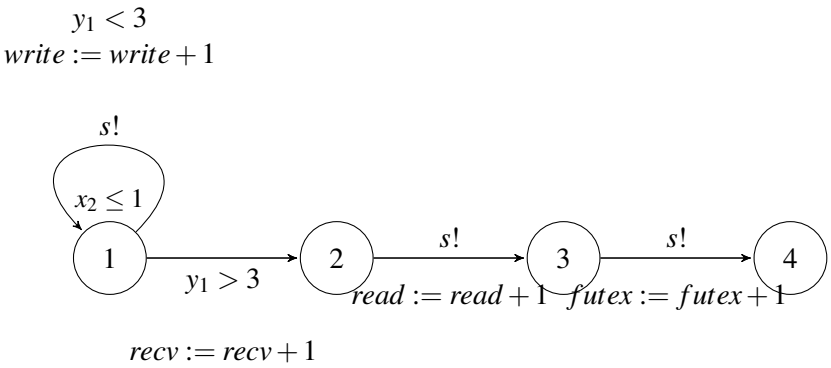

Figure 1: An example of automaton generated from a system call trace.

two different clocks are considered for each automaton: the first one (i.e., x) to respect the loop entering condition, while the second one (i.e., y) to respect the exit one.

The *Formal Model Verification* receives as input a timed automaton built in the previous phase and a set of timed temporal logic properties. In detail timed temporal logic properties are considered are aimed to detect whether an malicious behaviour is occurring.

The set of logic properties is checked against the timed automata obtained using the UPPAAL model checker. Whether the UPPAAL formal verification environment outputs true when is verifying a timed temporal logic property on a network of timed automata, it means that the proposed method labelled the formal model is belonging to the ransomware specified by the analysed formula. Otherwise, the formal verification environment outputs false, meaning that the model under analysis is not belonging to the attacks described in the analysed formula.

3 Experimental Evaluation

We performed an experiment aimed to evaluate the effectiveness of the proposed method. Different applications, legitimate and ransomware, are considered.

The aim of the experiment is to evaluate the effectiveness of the the proposed model based on timed automata to discriminate between ransomware and legitimate Android samples.

A dataset made of 100 legitimate and 100 ransomware Android applications was collected: legitimate applications by several categories (call & contacts, education, entertainment, GPS & travel, internet, lifestyle, news & weather, productivity, utilities, business, communication, email & SMS, fun & games, health & fitness, live wallpapers, personalization) were downloaded from Google Play (and then controlled by Google Bouncer, a Google service that checks each app for malicious behaviour before publishing it on the official market [CMV16]) from January to June 2019, while ransomware applications belonging to 10 widespread ransomware families were obtained through the VirusTotal API [4]. All the applications were checked by means of

[4] https://developers.virustotal.com/reference

Table 2: The Android ransomware families involved in the evaluation with details related to the discovery year.

Family	description	Cipher	Locker	year
doublelocker	change the PIN of the device	X	X	2017
fusob	It demands to pay a fine from $100 to $200 USD		X	2015
koler	Mobile version of Reveton ransomware		X	2014
locker	It shows a ransom messages		X	2014
lockerpin	It is able to set a PIN on the device		X	2015
pletor	Mobile version of Cryptolocker	X		2014
porndroid	It resets the screen-lock PIN		X	2015
simplelocker	It encrypt SD cad file using the AES cipher	X		2014
svpeng	Android banking malware		X	2017
xbot	It steals victims' banking credentials	X	X	2016

Table 3: Experimental analysis result.

Precision	Recall	F-Measure
0.96	0.97	0.96

VirusTotal [5], a service that runs 57 different antimalware on the submitted applications. The analysis confirmed that our trusted samples did not contain any known malicious payload, while the malicious samples contained ransomware specific family payloads.

Table 2 shows the widespread Android ransomware families evaluated in the study.

We executed the Android application in a simulated environment with the aim to collect the syscall traces. From the traces belonging to the considered 200 different applications (100 legitimate and 100 ransomware) we build the models representing the application under analysis.

To evaluate the effectiveness of the proposed method in Android ransomware detection, three different metrics are considered: Precision, Recall and F-Measure.

The *Precision* has been computed as the proportion of the examples that truly belong to class X among all those which were assigned to the class. It is the ratio of the number of relevant records retrieved to the total number of irrelevant and relevant records retrieved: $Precision = \frac{tp}{tp+fp}$, where tp indicates the number of true positives and fp indicates the number of false positives.

The *Recall* has been computed as the proportion of examples that were assigned to class X, among all the examples that truly belong to the class, i.e., how much part of the class was captured. It is the ratio of the number of relevant records retrieved to the total number of relevant records: $Recall = \frac{tp}{tp+fn}$, where tp indicates the number of true positives and fn indicates the number of false negatives.

The *F-Measure* is a measure of a test's accuracy. This score can be interpreted as a weighted average of the precision and recall: $F\text{-}Measure = 2 * \frac{Precision*Recall}{Precision+Recall}$.

The obtained results are shown in Table 3.

As shown from the experimental results, the proposed method obtains a precision equal to

[5] https://www.virustotal.com/

0.96 and a recall of 0.97 confirming their effectiveness in Android ransomware detection.

4 Related Work

The main difference between ransomware and other widespread mobile malware types is in their behaviour: as demonstrated in [ZJ12a], Android malware generally focuses on remaining hidden while gathering and sending to the attackers user sensitive and private information. Ransomware, instead, leverages the knowledge of its presence by users to obtain the payment of a ransom. Due to this, current methods proposed by the research community for the identification of malware are not necessarily effective in detecting ransomware, as the recent rise in ransomware attacks demonstrates. As also stated in [AZM15], this ineffectiveness of traditional methods, when used alone, exposes more than a billion users to this threat. Effective solutions for detection of ransomware on mobile devices are needed, but up to date, there are only a few works addressing this issue in the literature.

The first method that introduced ransomware detection for Android is HelDroid [AZM15]. This tool includes a text classifier based on NLP features, a lightweight smali emulation technique to detect locking strategies, and the application for detecting file-encrypting flows. This tool includes a text classifier based on NLP features, a lightweight smali emulation technique to detect locking strategies, and the application for detecting file-encrypting flows. The main weakness of HelDroid is that it strongly depends on a text classifier: as a matter of fact, the authors trained it on generic threatening phrases, similar to those that typically appear in ransomware or scareware. From the performance point of view, they identify rightly 375 Android ransomware on a dataset composed of 443 samples: 11 ransomware were not detected due to unsupported language (e.g., Spanish, Russian) with 9 out of 12,842 false positives.

In [YYQ+15], the authors propose a performance tool in order to help to understand what can be done to cope with Android ransomware detection. However, this method remained at the level of a proposal, with no implementation. Therefore, there are no results that can prove its effectiveness.

Song et al. [SKL16] designed an approach with the aim of identifying mobile ransomware by using process monitoring. They consider features representing the I/O rate as well as the CPU and memory usage. They evaluate the proposed method with only one ransomware sample developed by the authors. This sample has the ability to encrypt the file by using AES.

R-PackDroid [MMG+17] implements an approach to detect Android ransomware considering the API calls frequency. Authors evaluated a real-world ransomware dataset, demonstrating that R-PackDroid can detect novel ransomware only using previously released training samples.

An approach based on formal methods that is able to detect Android ransomware and to identify the malicious sections in the application code is described in [MNSV16, CMN+17]. Starting from the payload behaviour definition, the authors formulate logic rules that are later applied to detect ransomware. The main weakness of the proposed method is represented by the human analyst effort required to build the logic rules. As a matter of fact the proposed method foresees the payload identification but the process rule building has to be done by hand, and as such, is a time consuming task.

A hybrid static-dynamic approach is proposed in [GG17]. The static approach is based on

text and image classification as well as on API calls and application permissions. The dynamic approach is based on sequences of API calls that are compared, with a periodicity of five minutes, against malicious sequences identified for each malware family. Dynamic detection takes a minimum of five minutes to identify a ransomware sample.

Researchers in [FMM+17] propose a hybrid method able to effectively counter ransomware. The proposed method first examines applications to be used on a device prior to their installation (static approach) and then observes their behavior at runtime and identifies if the system is under attack (dynamic approach). To detect ransomware, the static detection method uses the frequency of opcodes while the dynamic detection method considers CPU usage, memory usage, network usage and system call statistics.

The main difference between the discussed methods and the one we propose is represented by the adoption of a dynamic analysis to build a timed automaton.

5 Conclusion Remarks

In this paper we propose to model Android applications in terms of timed automata. We consider system call traces to build the proposed model. The experiment is aimed to demonstrate the ability to discriminate between legitimate and ransomware applications. We evaluate 200 real-world applications, obtaining an F-Measure equal to 0.96. Future works will consider to experiment the proposed method on an extended dataset of application and to apply the proposed method on applications developed for Apple mobile environment.

Acknowledgments

This work has been partially supported by MIUR - SecureOpenNets and EU SPARTA contract 830892, CyberSANE projects, and the EU project CyberSure 734815.

Bibliography

[Add02] P. S. Addision. The Illustrated Wavelet Transform Handbook: Introductory Theory and Applications in Science, Engineering, Medicine and Finance. *Taylor & Francis Group*, 2002.

[AKMP03] I. N. Athanasiadis, V. G. Kaburlasos, P. A. Mitkas, V. Petridis. Applying machine learning techniques on air quality data for real-time decision support. In *ITEE*. 2003.

[AMS08] S. Attaluri, S. McGhee, M. Stamp. Profile hidden Markov models and metamorphic virus detection. *Journal of Computer Virology and Hacking Techniques* 5(2):179–192, November 2008.

[AMS18] B. A. S. Al-rimy, M. A. Maarof, S. Z. M. Shaid. Ransomware threat success factors, taxonomy, and countermeasures: a survey and research directions. *Computers & Security*, 2018.

[ASH+14] D. Arp, M. Spreitzenbarth, M. Huebner, H. Gascon, K. Rieck. Drebin: Efficient and Explainable Detection of Android Malware in Your Pocket. In *Annual Network and Distributed System Security Symposium (NDSS)*. Pp. 1–15. February 2014.

[ASKA16] S. Arshad, M. A. Shah, A. Khan, M. Ahmed. Android malware detection & protection: a survey. *International Journal of Advanced Computer Science and Applications* 7(2):463–475, 2016.

[AZM15] N. Andronio, S. Zanero, F. Maggi. HelDroid: Dissecting and detecting mobile ransomware. In *International Workshop on Recent Advances in Intrusion Detection*. Pp. 382–404. 2015.

[bak15] An assembler / disassembler for Android's dex format. https://code.google.com/p/smali/, last visit 26 January 2015.

[BDP93] M. Bowman, S. K. Debray, L. L. Peterson. Reasoning About Naming Systems. *ACM Trans. Program. Lang. Syst.* 15(5):795–825, November 1993.

[BKK06] U. Bayer, C. Kruegel, E. Kirda. TTAnalyze: A tool for analyzing malware. In *European Institute for Computer Antivirus Research Annual Conference*. 2006.

[BLS13a] D. Baysa, R. M. Low, M. Stamp. Structural entropy and metamorphic malware. *Journal of computer virology and hacking techniques* 9(4):179–192, 2013.

[BLS13b] D. Baysa, R. M. Low, M. Stamp. Structural entropy and metamorphic malware. *Journal of Computer Virology and Hacking Techniques* 9(4):179–192, November 2013.

[Bor11] M. Borda. Fundamentals in Information Theory and Coding. *Springer*, 2011.

[bou15] Dissecting the Android Bouncer. https://jon.oberheide.org/files/summercon12-bouncer.pdf, last visit 30 January 2015.

[Bra91] J. Braams. Babel, a Multilingual Style-Option System for Use with LaTeX's Standard Document Styles. *TUGboat* 12(2):291–301, June 1991.

[CCP01] M. Cannataro, A. Cuzzocrea, A. Pugliese. A probabilistic approach to model adaptive hypermedia systems. In *in Proceedings of the International Workshop for Web Dynamics*. Pp. 12–30. 2001.

[CFR13] A. Cuzzocrea, G. Fortino, O. F. Rana. Managing Data and Processes in Cloud-Enabled Large-Scale Sensor Networks: State-of-the-Art and Future Research Directions. In *13th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing, CCGrid 2013, Delft, Netherlands, May 13-16, 2013*. Pp. 583–588. 2013.

[Cla91] M. Clark. Post Congress Tristesse. In *TeX90 Conference Proceedings*. Pp. 84–89. March 1991.

[CMM+19] M. F. Carfora, F. Martinelli, F. Mercaldo, V. Nardone, A. Orlando, A. Santone, G. Vaglini. A "pay-how-you-drive" car insurance approach through cluster analysis. *Soft Computing* 23(9):2863–2875, 2019.

[CMN+17] A. Cimitile, F. Mercaldo, V. Nardone, A. Santone, C. A. Visaggio. Talos: no more ransomware victims with formal methods. *International Journal of Information Security*, Dec 2017.
doi:10.1007/s10207-017-0398-5
https://doi.org/10.1007/s10207-017-0398-5

[CMN+18] A. Cimitile, F. Mercaldo, V. Nardone, A. Santone, C. A. Visaggio. Talos: no more ransomware victims with formal methods. *International Journal of Information Security* 17(6):719–738, 2018.

[CMV13] G. Canfora, F. Mercaldo, C. A. Visaggio. A classifier of Malicious Android Applications. In *International Conference on Availability, Reliability and Security*. Pp. 607–614. 2013.

[CMV16] G. Canfora, F. Mercaldo, C. A. Visaggio. An hmm and structural entropy based detector for android malware: An empirical study. *Computers & Security* 61:1–18, 2016.

[CRTE13] S. Chakradeo, B. Reaves, P. Traynor, W. Enck. MAST: triage for market-scale mobile malware analysis. In *ACM conference on Security and privacy in wireless and mobile networks (WiSec)*. Pp. 13–24. April 2013.

[DLSV14] N. De Francesco, G. Lettieri, A. Santone, G. Vaglini. GreASE: A tool for efficient "Nonequivalence" checking. *ACM Transactions on Software Engineering and Methodology* 23(3), 2014.

[DLSV16] N. De Francesco, G. Lettieri, A. Santone, G. Vaglini. Heuristic search for equivalence checking. *Software and Systems Modeling* 15(2):513–530, 2016.

[DMS04] D. Dagon, T. Martin, T. Starner. Mobile Phones asComputing Devices:The Viruses are Coming! *Pervasive Computing, IEEE* 3(4):11–15, October-December 2004.

[DNL14] L. Deshotels, V. Notani, A. Lakhotia. DroidLegacy: Automated Familial Classification of Android Malware. In *ACM SIGPLAN on Program Protection and Reverse Engineering Workshop*. 2014.

[FADA14] Y. Feng, S. Anand, I. Dillig, A. Aiken. Apposcopy: semantics-based detection of Android malware through static analysis. In *ACM SIGSOFT International Symposium on Foundations of Software Engineering*. Pp. 576–587. 2014.

[FGL+13] P. Faruki, V. Ganmoor, V. Laxmi, M. S. Gaur, A. Bharmal. AndroSimilar: robust statistical feature signature for Android malware detection. In *International Conference on Security of Information and Networks*. Pp. 151–159. 2013.

[FMM+17] A. Ferrante, M. Malek, F. Martinelli, F. Mercaldo, J. Milosevic. Extinguishing Ransomware-a Hybrid Approach to Android Ransomware Detection. In *The 10th International Symposium on Foundations Practice of Security*. 2017.

[fra15] On the Effectiveness of Malware Protection on Android. http://www.aisec.fraunhofer.de/content/dam/aisec/Dokumente/Publikationen/Studien_TechReports/deutsch/042013-Technical-Report-Android-Virus-Test.pdf, last visit 21 January 2015.

[fse15] Mobile Threat Report. https://www.f-secure.com/documents/996508/1030743/Threat_Report_H1_2014.pdf, last visit 7 February 2015.

[gar15] Gartner Predicts by 2017, Half of Employers will Require Employees to Supply Their Own Device for Work Purposes. http://www.gartner.com/newsroom/id/2466615, last visit 21 January 2015.

[GG17] A. Gharib, A. Ghorbani. DNA-Droid: A Real-Time Android Ransomware Detection Framework. In Yan et al. (eds.), *Network and System Security: 11th International Conference, NSS 2017, Helsinki, Finland, August 21–23, 2017, Proceedings*. Pp. 184–198. Springer International Publishing, Cham, 2017.

[GSVV04] S. Gradara, A. Santone, M. Villani, G. Vaglini. Model checking multithreaded programs by means of reduced models. *Electronic Notes in Theoretical Computer Science* 110:55–74, 2004.

[Her93] M. Herlihy. A Methodology for Implementing Highly Concurrent Data Objects. *ACM Trans. Program. Lang. Syst.* 15(5):745–770, November 1993.

[HH09] J. Hühn, E. Hüllermeier. FURIA: an algorithm for unordered fuzzy rule induction. *Data Mining and Knowledge Discovery* 19(3):293–319, 2009.

[idc15] Smartphone OS Market Share, Q3 2014. http://www.idc.com/prodserv/smartphone-os-market-share.jsp, last visit 7 February 2015.

[IM07] N. Idika, A. P. Mathur. A survey of malware detection techniques. *Purdue University* 48:2007–2, 2007.

[IY04] H. Ishibuchi, T. Yamamoto. Fuzzy rule selection by multi-objective genetic local search algorithms and rule evaluation measures in data mining. *Fuzzy sets and systems* 141(1):59–88, 2004.

[JC08] R. Jensen, C. Cornelis. A new approach to fuzzy-rough nearest neighbour classification. In *International Conference on Rough Sets and Current Trends in Computing*. Pp. 310–319. 2008.

[JC11] R. Jensen, C. Cornelis. Fuzzy-rough nearest neighbour classification. In *Transactions on rough sets XIII*. Pp. 56–72. Springer, 2011.

[KF06] T. Kinjo, K. Funaki. On HMM Speech Recognition Based on Complex Speech Analysis. In *Annual Conference on Industrial Electronics*. Pp. 3477–3480. 2006.

[kin15] Kindsight Security Labs Malware Report - Q4 2013. http://www.tmcnet.com/tmc/whitepapers/documents/whitepapers/2014/9861-kindsight-security-labs-malware-report-q4-2013.pdf, last visit 21 January 2015.

[kre15] Who wrote the Pincer Android Trojan? http://krebsonsecurity.com/2013/08/who-wrote-the-pincer-android-trojan/, last visit 21 January 2015.

[Lam86] L. Lamport. *LaTeX User's Guide and Document Reference Manual*. Addison-Wesley Publishing Company, Reading, Massachusetts, 1986.

[LD14] S. Liang, X. Du. Permission-Combination-based Scheme for Android Mobile Malware Detection. In *International Conference on Communications*. Pp. 2301–2306. 2014.

[LH07] R. Lyda, J. Hamrock. Using Entropy Analysis to Find Encrypted and Packed Malware. *Security & Privacy, IEEE* 5(2):40–45, March-April 2007.

[LL14] X. Liu, J. Liu. A Two-layered Permission-based Android Malware Detection Scheme. In *International Conference on Mobile Cloud Computing, Service, and Engineering*. Pp. 142–148. 2014.

[LTTA14] E. Lagerspetz, H. T. T. Truong, S. Tarkome, N. Asokan. MDoctor: A Mobile Malware Prognosis Application. In *International Conference on Distributed Computgin Systems Workshops*. Pp. 201–206. 2014.

[mca15] Update: McAfee: Cyber criminals using Android malware and ransomware the most. http://www.infoworld.com/article/2614854/security/update--mcafee--cyber-criminals-using-android-malware-and-ransomware-the-most.html, last visit 21 January 2015.

[met15] metamorphic and polymorphic malware. http://searchsecurity.techtarget.com/definition/metamorphic-and-polymorphic-malware, last visit 21 January 2015.

[MMG+17] D. Maiorca, F. Mercaldo, G. Giacinto, C. A. Visaggio, F. Martinelli. R-PackDroid: API package-based characterization and detection of mobile ransomware. In *Proceedings of the Symposium on Applied Computing*. Pp. 1718–1723. 2017.

[MMM17] F. Martinelli, F. Marulli, F. Mercaldo. Evaluating Convolutional Neural Network for Effective Mobile Malware Detection. *Procedia Computer Science* 112:2372–2381, 2017.

[MMMS18] F. Martinelli, F. Mercaldo, C. Michailidou, A. Saracino. Phylogenetic Analysis for Ransomware Detection and Classification into Families. In *ICETE (2)*. Pp. 732–737. 2018.

[MMN+18a] F. Martinelli, F. Mercaldo, V. Nardone, A. Orlando, A. Santone. Cluster analysis for driver aggressiveness identification. In *ICISSP*. Pp. 562–569. 2018.

[MMN+18b] F. Martinelli, F. Mercaldo, V. Nardone, A. Orlando, A. Santone. Who's Driving My Car? A Machine Learning based Approach to Driver Identification. In *ICISSP*. Pp. 367–372. 2018.

[MNS16] F. Mercaldo, V. Nardone, A. Santone. Ransomware inside out. In *Availability, Reliability and Security (ARES), 2016 11th International Conference on*. Pp. 628–637. 2016.

[MNSV16] F. Mercaldo, V. Nardone, A. Santone, C. A. Visaggio. Ransomware Steals Your Phone. Formal Methods Rescue It. In *International Conference on Formal Techniques for Distributed Objects, Components, and Systems*. Pp. 212–221. 2016.

[nfc15] Using NFC? 3 Security risks to be aware of. http://www.makeuseof.com/tag/using-nfc-3-security-risks-to-be-aware-of/, last visit 21 January 2015.

[PF05] T. Plotz, G. A. Fink. A new approach for HMM based protein sequence family modeling and its application to remote homology classification. In *Workshop on Statistical Signal Processing*. Pp. 1008–1013. 2005.

[Pie04] M. Piercy. Embedded devices next on the virus target list. *Electronics Systems and Software* 2(6):42–43, December-January 2004.

[QCZD11] Z. Qin, N. Chen, Q. Zhang, Y. Di. Mobile Phone Viruses Detection Based on HMM. In *International Conference on Multimedia Information Networking and Security*. Pp. 516–519. 2011.

[QWR18] L. Qiu, Y. Wang, J. Rubin. Analyzing the analyzers: Flowdroid/iccta, amandroid, and droidsafe. In *Proceedings of the 27th ACM SIGSOFT International Symposium on Software Testing and Analysis*. Pp. 176–186. 2018.

[RCJ13] V. Rastogi, Y. Chen, X. Jiang. DroidChameleon:Evaluating Android Anti-malware against Transformation Attacks. In *ACM Symposium on Information, Computer and Communications Security*. Pp. 329–334. May 2013.

[ROS12] R. Ramachandran, T. Oh, W. Stackpole. Android Anti-Virus Analysis. In *Annual Symposium on Information Assurance & Secure Knowledge Management*. Pp. 35–40. June 2012.

[SES+13] M. Spreitzenbarth, F. Echtler, T. Schreck, F. C. Freling, J. Hoffmann. Mobile-Sandbox: Looking Deeper into Android Applications. In *International ACM Symposium on Applied Computing (SAC)*. Pp. 1808–1815. March 2013.

[SH78] S. Salas, E. Hille. *Calculus: One and Several Variable*. John Wiley and Sons, New York, 1978.

[SKL16] S. Song, B. Kim, S. Lee. The Effective Ransomware Prevention Technique Using Process Monitoring on Android Platform. *Mobile Information Systems* 2016, 2016.

[sma15] Dalvik opcodes. http://pallergabor.uw.hu/androidblog/dalvik_opcodes.html, last visit 26 January 2015.

[Sor11] I. Sorokin. Comparing files using structural entropy. *Journal of Computer Virology and Hacking Techniques* 7(4):259–265, November 2011.

[sym15] Remote Access Tool Takes Aim with Android APK Binder. http://www.symantec.com/connect/blogs/remote-access-tool-takes-aim-android-apk-binder, last visit 21 January 2015.

[USS+12] X. Ugarte-Pedrero, I. Santos, B. Sanz, C. Laorden, P. G. Bringas. Countering Entropy Measure Attacks on Packed Software Detection. In *The 9th Annual IEEE Consumer Communications and Networking Conference - Security and Content Protection*. Pp. 164–168. January 2012.

[WH14] W.-C. Wu, S.-H. Hung. DroidDolphin: a dynamic Android malware detection framework using big data and machine learning. In *Conference on Research in Adaptive and Convergent Systems*. Pp. 247–252. 2014.

[WKQ+08] X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, S. Y. Philip et al. Top 10 algorithms in data mining. *Knowledge and information systems* 14(1):1–37, 2008.

[XLQZ12] K. Xin, G. Li, Z. Qin, Q. Zhang. Malware Detection in Smartphones Using Hidden Markov Model. In *International Conference on Multimedia Information Networking and Security*. Pp. 857–860. 2012.

[XZM+17] X. Xiao, S. Zhang, F. Mercaldo, G. Hu, A. K. Sangaiah. Android malware detection based on system call sequences and LSTM. *Multimedia Tools and Applications*, pp. 1–21, 2017.

[YSMM13] S. Y. Yerima, S. Sezer, G. McWilliams, I. Muttik. A New Android Malware Detection Approach Using Bayesian Classification. In *International Conference on Advanced Information Networking and Applications*. Pp. 121–128. 2013.

[YY10] I. You, K. Yim. Malware Obfuscation Techniques: A Brief Survey. In *International Conference on Broadband, Wireless Computing, Communication and Applications*. Pp. 297–300. November 2010.

[YYQ+15] T. Yang, Y. Yang, K. Qian, D. C.-T. Lo, Y. Qian, L. Tao. Automated detection and analysis for android ransomware. In *IEEE 17th International Conference on High Performance Computing and Communications, IEEE 7th International Symposium on Cyberspace Safety and Security, IEEE 12th International Conference on Embedded Software and Systems*. Pp. 1338–1343. 2015.

[ZJ12a] Y. Zhou, X. Jiang. Dissecting android malware: Characterization and evolution. In *Security and Privacy (SP), 2012 IEEE Symposium on*. Pp. 95–109. 2012.

[ZJ12b] Y. Zhou, X. Jiang. Dissecting Android Malware: Characterization and Evolution. In *IEEE Symposium on Security and Privacy (SP)*. Pp. 95–109. May 2012.