ISSN 2278-3091

Volume 10, No.3, May - June 2021

International Journal of Advanced Trends in Computer Science and Engineering

Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse1271032021.pdf

https://doi.org/10.30534/ijatcse/2021/1281032021

Designing Threat Detection Model for Emerging Software Platform

Saima Shoro[1], Samina Rajper[2], Aisha Anwar[3], Sadia Anwar[4,] Abdul Basit Shaikh[5].

Department Of Computer Science,
[1]Shah Abdul Latif University Khairpur Mir;s, Pakistan, shorosaima@gmail.com
[2]Shah Abdul Latif University Khairpur Mir;s, Pakistan, Samina.rajper@salu.edu.pk
[3]Shah Abdul Latif University Khairpur Mir;s, Pakistan,Aishaanwar.27@yahoo.com
[4]Shah Abdul Latif University Khairpur Mir;s, Pakistan,Sadiaanwar.26@gmail.com
[5]Shah Abdul Latif University Khairpur Mir;s, Pakistan, Abasit.shaikh@salu.edu.pk

ABSTRACT

By the emergence of Smart Phones and Internet of Things platforms, new software systems are continually sophisticated, deployed in high dynamic situations and permitting interactions across heterogeneous domains. As a result analyzing their security risks critical issue necessitates high level of stability and scalability in order to handle the dynamic inherent in such predictable systems. This article aims to discuss several security threats in traditional systems and their safety to deal with the new aspects of today's software systems. It focuses on the security of Internet of Things and Android Platforms. The undertaken study reveals the emergent software, their security challenges, features of Internet of Things and Android platforms, to detect the venerable and dangerous interactions between applications which shares the common devices and proposed a Threat detection model to find out the security risks in emerging software platforms. It helps the scholars, academicians and software developers to cope with real world problems.

Key words: Emergent Software Platform, Internet of Things, Android, Security Threats and App interactions.

1.INTRODUCTION

Emergent software platforms have a huge impact on various parts of modern society, such as entertainment, human relationships, scientific advancement, economic growth and education. These modern applications are very dynamic and frequently interact over complex infrastructures. The growing market and widespread use of mobile and smart devices have ushered in a new era of emerging software.

1.1 Emergent Software Platforms

Traditional software relies on strict development chains, in which people, corporations, and, in some situations, entire communities develop and distribute software. In contrast to traditional development chains, emergent software systems involve a network of players who are responsible for the distribution of software and are more loosely tied. Furthermore, software development is growing more sophisticated these days [1], [2], as applications in emerging platforms move away from low-cost recreational apps and toward more business-oriented applications [1]. Indeed, emergent software has forced fundamental changes in how software is generated and consumed, as well as how users interact with mobile and smart devices, during the previous decade [3]. As a result, in compared to traditional software, contemporary software now has emergent properties. Unlike traditional apps, which are self-contained, emergent software platforms are developed from reusable components of software behavior. This trend encourages emerging software platforms to share features via inter-component communication on the Android platform [4], as well as IoT app interactions manifested through sensor and actuator coordination in smart homes [5]. On the other hand these platforms are also facing various security challenges, which may destroy the efficiencies of Android and IOT devices. Therefore this study highlighted various security risks faced by these platforms now a days. Several Internet of Things and Android Applications are emerged but the main focus of this study was the Smart Home platforms and Android Platforms. (Shown in Figure 1) Various researchers worked on the security risks of these platform which is illustrated in next section.

Figure 1 Emerging Software Platforms

1.1.1 Feature Interaction

In the emergent software era, interoperability refers to the capacity of applications to communicate and exchange data. Inter-app communication [6] and interactions between sensors and actuators in the trigger/action ecosystem [7] are examples of this interaction in the Android platform. The behavior of one feature (or a set of other features) is influenced by the existence of another feature (or a set of other features), which is known as Feature Interaction [8, 9, 10, 11]. As a result, the feature interaction concept adds value to the user's experience by providing value-added services. The feature interaction concept is discussed in the context of Android and smart home platforms in the next section.

A. Feature Interaction in the Android platform

Inter-component communication (ICC) or cross-app communication (Inter-app communication) are two terms used to describe how the Android platform facilitates interaction between components within an app. Despite the fact that IAC improves user experience and reduces development burden, it can be used to launch collusive assaults [12]. Reflection and Dynamic Class Loading (DCL) are two dynamic Java programming capabilities that can be used to hide this attack. Dynamic programming features are justified because their use is likely to increase in the amplified era [2]. For backward compatibility, accessing hidden/internal application program interface (API), offering external library support, and reinforcing app security, the Java reflection technique is widely utilized in Android apps [13, 14].

B. Feature Interaction in Smart Home Platform

In a smart home, many IoT platforms can control the same collection of sensors and actuators (i.e. Smart Things Groovy and Smart Things IFTTT). As a result, the race to configure, control, and monitor these devices may begin [15]. Users can install third-party software programs to automate their home's equipment on these platforms. Software programs deployed by users may interact in both physical and cyberspaces by controlling physical equipment in a system. This allows for sophisticated and varied automation. While providing a variety of alternatives for automating one's house improves the user's experience, it also increases the attack surface for safety and security risks. Interaction between smart home applications and gadgets has the potential to alter both cyberspace and physical space, posing serious safety and security risks. As a result, detecting potentially dangerous interactions is critical.

C. Cross Architecture Malware

The development of cross architecture apps and firmware [16, 17], which support diverse CPU architectures of Internet of Things devices, underpins interoperability in the emergent software era. This entails the ability to program Internet of Things devices of various architectures with a single compiler [18], allowing for heterogeneous firmware upgrades rather than monolithic binary updates. Furthermore, encouraging cross-architecture means that the code base will be built with diverse compilers and configurations (e.g., different optimization levels). Mirai virus was created for infecting various designs of Internet of Things devices in the Internet of Things malware sector. Due to its cross-architecture capabilities, Mirai was able to infect a huge number of Internet of Things devices for a few hours, causing a massive Internet service outage [19]. As a result, cyber-attackers are more interested in Internet of Things devices.

2. LITERATURE REVIEW

In order to construct a threat detection model for Android and Internet of Things, the major task was to analyze the threats in these platforms. Therefore, several articles and reports from literature have been reviewed which are based on the features of the emergent software's and their security. Some of which are highlighted in the table given below. (Table 1)

Table 1 Literature Review

LITERATURE REVIEW				
Article	**Year**	**Android Inter App Communication**	**Smart Home Safety & Security**	**Internet of Things Malware**
[20][21]	2020	✓		
[22][23]	2019		✓	
[24][25] [26][27]	2018	✓		
[28][29] [30][31] [32]	2018		✓	
[33]	2017	✓		
[34][35]	2017		✓	
[36]	2017			✓
[37]	2016	✓		
[38][39]	2016		✓	
[40][41] [42][43]	2015	✓		
[44]	2015		✓	
[45]	2015			✓

3. RESEARCH METHODOLOGY

In order to construct the threat detection model several steps, analysis and calculation were required Figure 2 discussed the core milestones of this research analysis.

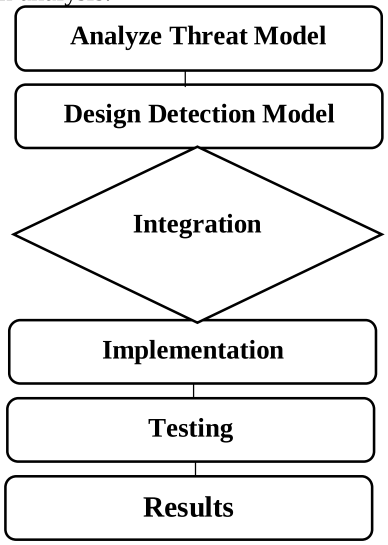

Figure 2 Research Methodology

4. ANALYZING THREAT MODEL

Information leakage, intent spoofing, and Android component activation are three severe dangers that are outlined as follows:

4.1 Information Leakage

When a receiver app exfiltrates sensitive data collected through IAC communications from other applications and sends it to an external location, this is known as information leakage.

4.2 Intent spoofing

Intent spoofing is a security exploit in which the sender app creates fake Intents in order to deceive receiving apps [46].

4.3 Android component activation

Because the Intent is not effectively secured by permission constraints, an Android component activation occurs when a malicious app intercepts an implicit Intent by defining an Intent filter matching the Intent [46]. The threat Model is depicted in Figure 3

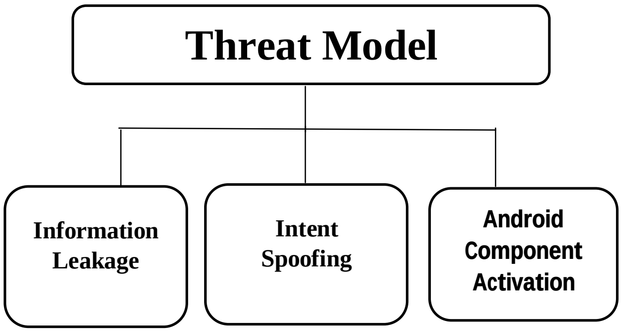

Figure 3 Threat Model

A harmful component is one that makes use of Intent sending/receiving APIs to aid transport malicious Intents that include sensitive data for data leakage, are faked for Intent spoofing, or are received in an illegal way. A malicious component is the source of the data leaks. When the sender component is malicious, intent spoofing involves a path between two components, whereas unauthorized intent receipt involves a way between two components when the receiver component is malicious.

5. PROPOSED MODEL

After analyzing the various threats in emergent software, the next step was to design the model which helps to detect the security risks in Android Applications and Internet of Things. The designed model is illustrated in Figure 4.

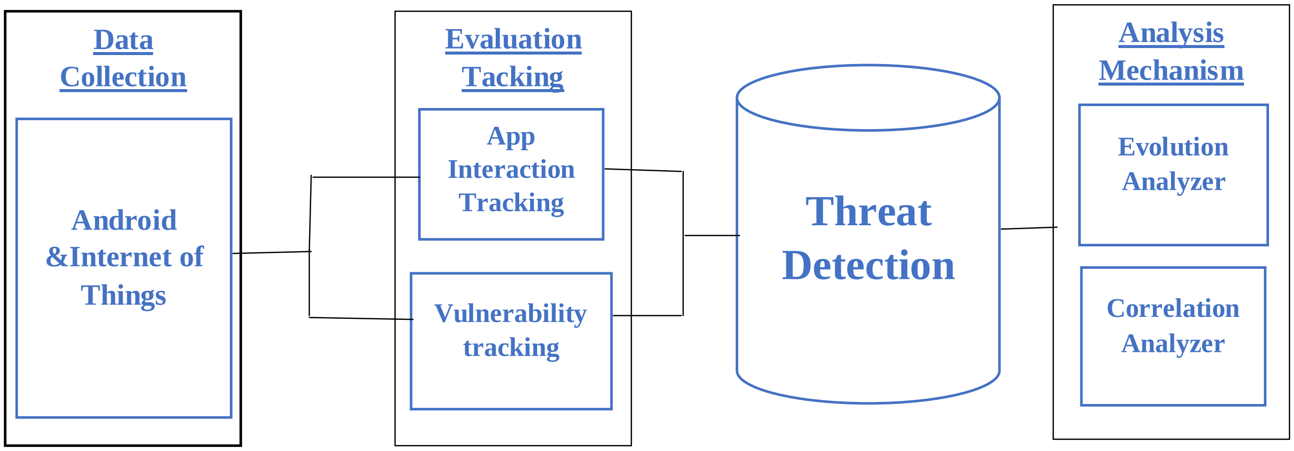

Figure 4 Threat Detection Model Android and Internet of Things

5.1 Data Collection

In this phase the collection of the data about the events of applications takes place. This data will be analyzed in the next phases to filter out the emerging software. Whereas for testing the proposed model the identification of data is based on the principals utilized in [47].

5.2 Evolution Tracking

App Interaction and vulnerability tracking are two essential milestones in this phase. The app interaction event and vulnerabilities tracking takes place in this phase.

5.3 Analysis mechanism

In this phase the analytical aspects, such as the evolution analysis and correlation analysis carried out. The evolution analyzer detects the app interaction events as well as their inspection, survival and elimination. On the other hand the correlation analyzer determines the relationships between app interaction event and vulnerability.

6. EXPERIMENTS AND RESULTS

For testing the proposed model, four bundles of android applications and Internet of Things applications were examined. Each bundle comprises of five to six applications based on similar features and compositions. At initial stage the data collection takes place. The data collection of each bundle was based on the criteria used in [47]. After data collection the data was analyzed by filtering them through all phases of detection model. The tracking phase in which the tracking of app interaction and vulnerability takes place identifies the threat. After that the last phase identifies the evolution and correlation among interactions. The accuracy rate of the detection model is based on the accurate identification of threat using particular data of applications. From the experiment it is found that the model gives 92% accuracy rate for threat detection. (Shown in Figure 5) Which shows that the proposed model will be helpful to detect threat and security risks and provide the efficiency of applications by removing the threat emerging events.

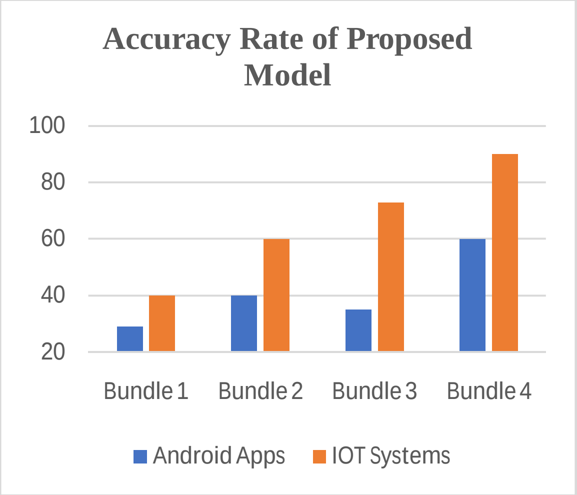

Figure 5 Accuracy Rate of Proposed Model

7. DISCUSSION AND CONCLUSION

The undertaken study tend to propose a threat detection model to find out various threat occurs in android applications and Internet of Things. Therefore several applications selected to evaluate the designed model. The threat detection is totally based on the app interaction events and vulnerability events. For experimental approach four bundles of applications were selected. For testing particular data was collected which was the actions performed by the applications, timing to complete the tasks, app interaction, security and safety of the applications. This data was passed through all the phases of model to detect threat and to detect the event from where the threat was initialized. The results shows that the proposed model provides 92% accurate results, therefore this model makes the threat detection easy. By detecting and removing the threats and events the security risks will be reduced and efficiency of the application will be increased. The future direction of the research will be to compare several threat detection models and to suggest a robust one among all.

REFERENCES

1. A. I. Wasserman, “Software engineering issues for mobile application development,” in Proceedings of the FSE/SDP Workshop on Future of Software Engineering Research, FoSER ’10, (New York, NY, USA), pp. 397–400, ACM, 2010.
2. A. Taivalsaari and T. Mikkonen, “On the development of iot systems,” in 2018 Third International Conference on Fog and Mobile Edge Computing (FMEC), pp. 13–19, April 2018.
3. Y. Acar, M. Backes, S. Bugiel, S. Fahl, P. McDaniel, and M. Smith, “Sok: Lessons learned from android security research for appified software plat- forms,” in 2016 IEEE Symposium on Security and Privacy (SP), pp. 433–451, May 2016.
4. D. Octeau, P. McDaniel, S. Jha, A. Bartel, E. Bodden, J. Klein, and Y. L. Traon, “Effective Inter-Component Communication Mapping in Android with Epicc: An Essential Step Towards Holistic Security Analysis,” in Proc. of USENIX Security, 2013.
5. Z. B. Celik, P. McDaniel, and G. Tan, “Soteria: Automated iot safety and security analysis,” 2018.
6. W. Enck, M. Ongtang, and P. McDaniel, “On lightweight mobile phone ap- plication certification,” in Proceedings of the 16th ACM Conference on Computer and Communications Security, CCS ’09, pp. 235–245, 2009.
7. M. Nakamura, H. Igaki, and K.-i. Matsumoto, “Feature interactions in integrated services of networked home appliances: An object-oriented ap- proach.,” pp. 236–251, 01 2005.
8. S. Apel, J. M. Atlee, L. Baresi, and P. Zave, “Feature interactions: the next generation (dagstuhl seminar 14281),” in Dagstuhl Reports, vol. 4, Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2014.
9. S. Nguyen, “Feature-interaction aware configuration prioritization,” in Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, ESEC/FSE 2018, (New York, NY, USA), pp. 974–976, ACM, 2018.
10. T. Pedersen, T. Le Guilly, A. P. Ravn, and A. Skou, “A method for model checking feature interactions,” in 2015 10th International Joint Conference on Software Technologies (ICSOFT), vol. 1, pp. 1–10, July 2015.
11. S. Apel, S. Kolesnikov, N. Siegmund, C. Kästner, and B. Garvin, “Exploring feature interactions in the wild: The new feature-interaction challenge,” in Proceedings of the

5th International Workshop on Feature-Oriented Software Development, FOSD ’13, (New York, NY, USA), pp. 1–8, ACM, 2013.

12. W. Enck, M. Ongtang, and P. McDaniel, “Understanding android security,” IEEE Security Privacy, vol. 7, pp. 50–57, Jan 2009.
13. L. Li, T. F. Bissyandé, D. Octeau, and J. Klein, “DroidRA: Taming reflection to support whole-program analysis of android apps,” in Proc. of ISSTA, pp. 318–329, 2016.
14. Y. Zhauniarovich, M. Ahmad, O. Gadyatskaya, B. Crispo, and F. Massacci, “Stadyna: Addressing the problem of dynamic code updates in the security analysis of android applications,” in Proc. of CODASPY ’15, pp. 37–48, 2015.
15. D. T. Nguyen, C. Song, Z. Qian, S. V. Krishnamurthy, E. J. M. Colbert, and P. McDaniel, “Iotsan: Fortifying the safety of iot systems,” in Proceedings of the 14th International Conference on Emerging Networking EXperiments and Technologies, CoNEXT ’18, (New York, NY, USA), pp. 191–203, ACM, 2018.
16. M. Abomhara and G. M. Kien, “Cyber Security and the Internet of Things: Vulnerabilities, Threats, Intruders and Attacks,” Journal of Cyber Security and Mobility, vol. 4, no. 1, pp. 65–88, 2015.
17. S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Porisini, “Security, privacy and trust in internet of things: The road ahead,” Computer Networks, vol. 76, pp. 146–164, 2015.
18. D. Navarro, F. Mieyeville, W. Du, M. Galos, and I. O’Connor, “Towards a design framework for heterogeneous wireless sensor networks,” in 2011 1st International Symposium on Access Spaces (ISAS), pp. 83–88, June 2011.
19. B. Herzberg, D. Bekerman, and I. Zeifman, “Breaking Down Mi- rai: An IoT DDoS Botnet Analysis.” https://www.incapsula.com/blog/ malware-analysis-mirai-ddos-botnet.html, Accessed at Feb 21, 2017.
20. Van Duong, Lai, Tisenko Victor Nikolaevich, Nguyen Quang Dam Do Hoang Long, and Nguyen Quoc Hoang. "Detecting malicious applications on Android is based on static analysis using Deep Learning algorithm." *International Journal* 9, no. 3 (2020).
21. Nasri, Nuren Natasha Maulat, Mohd Faizal Ab Razak, RD Rohmat Saedudin, Salwana Mohamad, and Ahmad Firdaus Asmara. "Android Malware Detection System using Machine Learning." *International Journal* 9, no. 1.5 (2020).
22. C. Davidson, T. Rezwana, and M. A. Hoque, “Smart home security applica- tion enabled by iot:,” in Smart Grid and Internet of Things (A.-S. K. Pathan, Z. M. Fadlullah, and M. Guerroumi, eds.), (Cham), pp. 46–56, Springer International Publishing, 2019.
23. M. Balliu, M. Merro, and M. Pasqua, “Securing cross-app interactions in iot platforms,” in IEEE Computer Security Foundations Symposium, 2019.
24. H. Bagheri, J. Wang, J. Aerts, and S. Malek, “Efficient, evolutionary security analysis of interacting android apps,” in Proceedings of the 34th IEEE Interna- tional Conference on Software Maintenance and Evolution (ICSME), pp. 357–368, 2018.
25. Z. B. Celik, L. Babun, A. K. Sikder, H. Aksu, G. Tan, P. McDaniel, and A. S. Uluagac, “Sensitive information tracking in commodity iot,” in Proceedings of the 27th USENIX Conference on Security Symposium, SEC’18, (Berkeley, CA, USA), pp. 1687–1704, USENIX Association, 2018.
26. Q. Wang, W. U. Hassan, A. M. Bates, and C. A. Gunter, “Fear and logging in the internet of things,” in 25th Annual Network and Distributed System Security Symposium, NDSS 2018, San Diego, California, USA, February 18-21, 2018, 2018.
27. I. Bastys, M. Balliu, and A. Sabelfeld, “If this then what?: Controlling flows in iot apps,” in Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, CCS ’18, (New York, NY, USA), pp. 1102–1119, ACM, 2018.
28. H. Chi, Q. Zeng, X. Du, and J. Yu, “Cross-app interference threats in smart homes: Categorization, detection and handling,” CoRR, vol. abs/1808.02125, 2018.
29. A. Rahmati, E. Fernandes, K. Eykholt, and A. Prakash, “Tyche: A risk-based permission model for smart homes,” in 2018

IEEE Cybersecurity Development (SecDev), pp. 29–36, IEEE, 2018.

30. A. Acar, H. Fereidooni, T. Abera, A. K. Sikder, M. Miettinen, H. Aksu, M. Conti, A.-R. Sadeghi, and A. S. Uluagac, "Peek-a-boo: I see your smart home activities, even encrypted!," arXiv preprint arXiv:1808.02741, 2018.
31. B. Ali and A. Awad, "Cyber and physical security vulnerability assessment for iot-based smart homes," Sensors, vol. 18, no. 3, p. 817, 2018.
32. A. Goudbeek, K. R. Choo, and N. Le-Khac, "A forensic investigation frame- work for smart home environment," in 201817thIEEEInternationalConference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/Big-DataSE), pp. 1446–1451, Aug 2018.
33. A. Sadeghi, H. Bagheri, J. Garcia, and S. Malek, "A taxonomy and qualita- tive comparison of program analysis techniques for security assessment of android software," IEEE Trans. Software Eng., vol. 43, no. 6, pp. 492–530, 2017.
34. Y. Tian, N. Zhang, Y.-H. Lin, X. Wang, B. Ur, X. Guo, and P. Tague, "Smar- tauth: User-centered authorization for the internet of things," in Proceedings of the 26th USENIX Conference on Security Symposium, SEC'17, (Berkeley, CA, USA), pp. 361–378, USENIX Association, 2017.
35. N. Apthorpe, D. Reisman, and N. Feamster, "Closing the blinds: Four strategies for protecting smart home privacy from network observers," arXiv preprint arXiv:1705.06809, 2017.
36. A. Costin, A. Zarras, and A. Francillon, "Towards Automated Classification of Firmware Images and Identification of Embedded Devices," in 32nd International Conference on ICT Systems Security and Privacy Protection (IFIP SEC), May 2017.
37. B. Schmerl, J. Gennari, A. Sadeghi, H. Bagheri, S. Malek, J. Camara, and D. Garlan, "Architecture Modeling and Analysis of Security in Android Systems," in Software Architecture, pp. 274–290, Nov. 2016.
38. E. Fernandes, J. Jung, and A. Prakash, "Security analysis of emerging smart home applications," in 2016 IEEE Symposium on Security and Privacy (SP), pp. 636–654, IEEE, 2016.
39. A. Tekeoglu and A. Tosun, "A testbed for security and privacy analysis of iot devices," in 2016 IEEE 13th International Conference on Mobile Ad Hoc and Sensor Systems (MASS), pp. 343–348, Oct 2016.
40. H. Bagheri, A. Sadeghi, J. Garcia, and S. Malek, "Covert: Compositional analysis of android inter-app permission leakage," IEEE Transactions on Software Engineering, vol. 41, no. 9, pp. 866–886, 2015.
41. L.Li, A.Bartel, T.F.Bissyandé, J.Klein, andY.L.Traon, "Apkcombiner: Com- bining multiple android apps to support inter-app analysis," in Proceedings of the IFIP TC 11 International Conference, pp. 513–527, 2015.
42. H. Bagheri, A. Sadeghi, R. Jabbarvand, and S. Malek, "Automated dynamic enforcement of synthesized security policies in android," tech. rep., Depart- ment of Computer Science, George Mason University, 2015.
43. P. Barros, R. Just, S. Millstein, P. Vines, W. Dietl, M. dAmorim, and M. D. Ernst, "Static analysis of implicit control flow: Resolving java reflection and android intents (t)," in 2015 30th IEEE/ACM International Conference on Automated Software Engineering (ASE), pp. 669–679, Nov 2015.
44. C. Busold, S. Heuser, J. Rios, A.-R. Sadeghi, and N. Asokan, "Smart and secure cross-device apps for the internet of advanced things," in International Conference on Financial Cryptography and Data Security, pp. 272–290, Springer, 2015.
45. M. Alazab, "Profiling and classifying the behavior of malicious codes," Jour- nal of Systems and Software, vol. 100, pp. 91–102, 2015.
46. E. Chin, A. P. Felt, K. Greenwood, and D. Wagner, "Analyzing inter- application communication in android," in Proc. of MobiSys'11, pp. 239–252, 2011.
47. U. A. Mannan, I. Ahmed, R. A. M. Almurshed, D. Dig, and C. Jensen, "Understanding code smells in android applications," in Proceedings of the International Conference on Mobile Software Engineering and Systems, MOBILESoft '16, (New York, NY, USA), pp. 225–234, ACM, 2016.