www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

ENHANCED AUTHORIZATION RECOGNITION FOR MACHINE LEARNING-DRIVEN DETECTION OF MALICIOUS ANDRIOD APPLICATIONS

[1] Mr. A. Sudheer,[2] Sathuri Pranavi,[3] Raskas Swetha,[4] S Sujith Reddy,[5] Sayyad Khwaja
[1]Assistant Professor,[2345]B.Tech Students
Department Of Computer Science & Engineering
Sri Indu College Of Engineering & Technology,Sheriguda, Ibrahimpatnam

ABSTRACT

The alarming growth rate of malicious apps has become a serious issue that sets back the prosperous mobile ecosystem. A recent report indicates that a new malicious app for Android is introduced every 10s. To combat this serious malware campaign, we need a scalable malware detection approach that can effectively and efficiently identify malware apps. Numerous malware detection tools have been developed, including system-level and network-level approaches. However, scaling the detection for a large bundle of apps remains a challenging task. In this paper, we introduce Significant Permission Identification (SigPID), a malware detection system based on permission usage analysis to cope with the rapid increase in the number of Android malware. Instead of extracting and analyzing all Android permissions, we develop three levels of pruning by mining the permission data to identify the most significant permissions that can be effective in distinguishing between benign and malicious apps. SigPID then utilizes machine-learning-based classification methods to classify different families of malware and benign apps. Our evaluation finds that only 22 permissions are significant. We then compare the performance of our approach, using only 22 permissions, against a baseline approach that analyzes all permissions. The results indicate that when a support vector machine is used as the classifier, we can achieve over 90% of precision, recall, accuracy, and F-measure, which are about the same as those produced by the baseline approach while incurring the analysis times that are 4-32 times less than those of using all permissions. Compared against other state-of-the-art approaches, SigPID is more effective by detecting 93.62% of malware in the dataset and 91.4% unknown/new malware samples.

I. INTRODUCTION

Android is currently the most used smart-mobile device platform in the world, occupying 85% of market share [1]. As of now, there are nearly 3 million apps available for downloading from Google Play, and more than 65 billion downloads to date [2]. Unfortunately, the popularity of Android also spurs interests from cyber-criminals who create malicious apps that can steal sensitive information and compromise mobile systems. Unlike other competing smart-mobile device platforms, such as iOS, Android allows users to install applications from unverified sources such as third party app stores and file sharing websites. The malware infection issue has been so serious that a recent report indicates that 97% of all mobile malware target Android devices [3]. In 2016 alone, over 3.25 million new malicious Android apps have been uncovered. This roughly translates to an introduction of a new malicious Android app every 10 seconds [4]. These malicious apps are created to perform different types of attacks in the form of Trojans , worms, exploits, and viruses. Some notorious malicious apps have more than 50 variants, which makes it extremely challenging to detect them all [5]. To address these elevating security concerns, researchers and analysts have used various approaches to develop Android malware detection tools [6]–[19]. For example, RISKRANKER [6] utilizes static analysis to discover malicious behaviors in Android apps. However, static analysis approaches generally assume more behaviors are possible than actually would be, which may lead to a large number of false positives. To improve the analysis accuracy, researchers have also proposed various dynamic

Index in Cosmos
JUNE 2025, Volume 15, ISSUE 2
UGC Approved Journal

www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

analysis methods to capture real- time execution context. For example, TAINTDROID [8] dynamically tracks multiple sensitive data source simultaneously using tainting analysis. However, dynamic analysis approaches, in general, need adequate input suites to sufficiently exercise execution paths. As we can use cloud computing to satisfy those requirements, the privacy concerns then become potential problem [20]–[22]. Petra et al. [23],[24] show that there is another broad range of anti-analysis techniques can be employed by advanced malware to successfully evade dynamic analysis based malware detection. Recently, more efforts have been spent on analyzing apps' behavioral data both in the Android system and other online data process system [25]–[27]. The apps' requested permissions have been used to enforce least-privilege, which to some extent indicate the apps' functionalities as well as runtime behaviors. As a result, researchers use machine learning and data mining techniques to detect Android malware based on

permission usage. For example, DREBIN [9] combines static analysis and machine learning techniques to detect Android malware. The experimental result shows that DREBIN can achieve high detection accuracy by incorporating as many features as possible to aid detection. However, using more features leads to increased modeling complexity, which increases the computational overhead of their system. Considering the large amount of new malicious apps, we need a detection system that can operate efficiently to identify these apps. Google also identifies 24 permissions out of the total of more than 300 permissions as "dangerous" [28]. At first glance, the list of dangerous permissions can be used as a guideline to help identify malicious applications. Yet, as will be shown in this paper, using just this list to identify malware still yields suboptimal detection effectiveness. In this paper, we present SIGPID, an approach that extracts significant permissions from apps, and uses the extracted information to effectively detect malware using supervised learning algorithms. The design objective of SIGPID is to detect malware efficiently and accurately. As stated earlier, the number of newly introduced malware is growing at an alarming rate. As such, being able to detect malware efficiently would allow analysts to be more productive in identifying and analyzing them. Our approach analyzes permissions and then identifies only the ones that are significant in distinguishing between malicious and benign apps. Specifically, we propose a multi-level data pruning approach including permission ranking with negative rate, permission mining with association rules and support based permission ranking to extract significant permissions strategically. Then, machine learning based classification algorithms are used to classify different types of malware and benign apps. The results of our empirical evaluation show that SIGPID can drastically reduce the number of permissions that we need to analyze to just 22 out of 135 (84% reduction), while maintaining over 90% malware detection accuracy and measure when Support Vector Machine (SVM) is used as the classifier. We also find that the number of significant permissions identified by our approach (22) is lower than the number of "dangerous" permissions identified by Google (24). Moreover, only 8 permissions jointly appear on our list and their list. This is because, as a data- driven approach, SIGPID dynamically determines significant permissions based on actual usage by the applications instead of statically defining dangerous permissions based on their intended services. This fundamental difference allows our approach to detect more malware than the approach that uses the dangerous list alone. To show the generality of this approach, we also test SIGPID with 67 commonly used supervised algorithms and

find that it maintains very high accuracy with all these algorithms Furthermore, we compare the accuracy and running time performance of our approach against two state- of-threat approaches, DREBIN [9], PERMISSION-INDUCED RISK MALWARE DETECTION [29], and existing virus scanners Again, we find that our approach can

detect more malware samples than the other approaches with significantly less overhead. In summary, our paper makes the following contributions: 1) we develop SIGPID, an approach that identifies an essential subset of permissions (significant permissions) that can be used to effectively identify Android malware. By using our technique, the number ofpermissions that needs to be analyzed is reduced by 84%. 2) We evaluate the effectiveness of our approach using only a fifth of the total number of permissions in Android. We find that SigPID can achieve over 90% in precision, recall, accuracy, and F-measure. These results compare favorably with those achieved by an approach that uses all 135 permissions as well as just the dangerous permission list. Compare with other state-of-the-art malware detection approaches, we find that SIGPID is more effective by detecting 93.62% of malicious apps in the data set and 91.4% unknown malware. 3) To show that the approach can work generically with a wide range of supervised learning algorithms, we apply SIGPID with 67 commonly used supervised learning algorithms and a much larger dataset (5,494 malicious and 310,926 benign apps). We find that 55 out of 67 algorithms can achieve F-measure of at least 85%, while the average running time can be reduced by 85.6% compared with the baseline approach.

II. LITERATURE SURVEY

TITLE: Risk ranker: scalable and accurate zero-day android malware detection

AUTHORS: M. Grace, Y. Zhou, Q. Zhang, S. Zou, and X. Jiang

ABSTRACT: Smartphone sales have recently experienced explosive growth. Their popularity also encourages malware authors to penetrate various mobile marketplaces with malicious applications (or apps). These malicious apps hide in the sheer number of other normal apps, which makes their detection challenging. Existing mobile anti-virus software are inadequate in their reactive nature by relying on known malware samples for signature extraction. In this paper, we propose a proactive scheme to spot zero-day Android malware. Without relying on malware samples and their signatures, our scheme is motivated to assess potential security risks posed by these un trusted apps. Specifically, we have developed an automated system called Risk Ranker to scalable analyze whether a particular app exhibits dangerous behavior (e.g., launching a root exploit or sending background SMS messages). The output is then used to produce a prioritized list of reduced apps that merit further investigation. When applied to examine 118,318 total apps collected from various Android markets over September and October 2011, our system takes less than four days to process all of them and effectively reports 3281 risky apps. Among these reported apps, we successfully uncovered 718 malware samples (in 29 families) and 322 of them are zero-day (in 11 families). These results demonstrate the efficacy and scalability of Risk Ranker to police Android markets of all stripes.

TITLE: Droidminer: Automated mining and characterization of fine-grained malicious behaviors in android applications

AUTHORS: C. Yang, Z. Xu, G. Gu, V. Yegneswaran, and P. Porras

ABSTRACT: Most existing malicious Android app detection approaches rely on manually selected detection heuristics, features, and models. In this paper, we describe a new, complementary system, called Droid Miner, which uses static analysis to automatically mine malicious program logic from known Android malware, abstracts this logic into a sequence of threat modalities, and then seeks out these threat modality patterns in other unknown (or newly published) Android apps. We formalize a two- level behavioral graph representation used to capture Android app program logic, and design new techniques to identify and label elements of the graph that capture malicious behavioral patterns (or malicious modalities). After the automatic learning of these malicious behavioral models, Droid Miner can scan a new Android app to (i) determine whether it contains malicious modalities, (ii) diagnose the malware family to which it is most closely associated, (iii) and provide further evidence as to why the app is considered to be malicious by

including a concise description of identified malicious behaviors. We evaluate Droid miner using 2,466 malicious apps, identified from a corpus of over 67,000 third-party market Android apps, plus an additional set of over 10,000 official market Android apps. Using this set of real-world apps, we demonstrate that Droid Miner achieves a 95.3% detection rate, with only a 0.4% false positive rate. We further evaluate Droid Miner's ability to classify malicious apps under their proper family labels, and measure its label accuracy at 92%.

TITLE: App context: Differentiating malicious and benign mobile app behaviors using context

AUTHORS: W. Yang, X. Xiao, B. Andow, S. Li, T. Xie, and W. Enck

ABSTRACT: Mobile malware attempts to evade detection during app analysis by mimicking security- sensitive behaviors of benign apps that provide similar functionality (e.g., sending SMS messages), and suppressing their payload to reduce the chance of being observed (e.g., executing only its payload at night). Since current approaches focus their analyses on the types of security-sensitive resources being accessed (e.g., network), these evasive techniques in malware make differentiating between malicious and benign app behaviors a difficult task during app analysis. We propose that the malicious and benign behaviors within apps can be differentiated based on the contexts that trigger security- sensitive behaviors, i.e., the events and conditions that cause the security-sensitive behaviors to occur. In this work, we introduce App Context, an approach of static program analysis that extracts the contexts of security sensitive behaviors to assist app analysis in differentiating between malicious and benign behaviors. We implement a prototype of App Context and evaluate App context on 202 malicious apps from various malware datasets, and 633 benign apps from the Google Play Store. App Context correctly identifies 192 malicious apps with 87.7% precision and 95% recall. Our evaluation results suggest that the maliciousness of a security-sensitive behavior is more closely related to the intention of the behavior (reflected via contexts) than the type of the security-sensitive resources that the behavior accesses.

III. SYSTEM ANALYSIS & DESIGN

EXISTING SYSTEM

IBM QRadar is a Security Information and Event Management (SIEM) system that combines log management, threat detection, and incident response into a single platform. It collects and correlates data from various sources, including logs, network traffic, and endpoints. QRadar uses predefined rules, anomaly detection, and machine learning to identify potential security incidents and unauthorized access. The system provides real-time visibility into the organization's security posture and enables security teams to respond to threats effectively.

DISADVANTAGES

- False Positives and Negatives: Existing systems often struggle with balancing false positives (incorrectly flagging legitimate activities as malicious) and false negatives (failing to detect actual threats), which can lead to inefficient use of resources and potential security gaps.
- Resource Intensiveness: Implementing and maintaining machine learning-driven systems can be resource-intensive in terms of computational power, data requirements, and ongoing updates, making them costly and complex to manage.

Adversarial Vulnerabilities: These systems are susceptible to adversarial attacks, where malicious actors manipulate data to deceive the system. This vulnerability can undermine the system's effectiveness and reliability in real-world security scenarios.

PROPOSED SYSTEM:

A proposed system could be a deep learning-based access control system designed to enhance substantial authorization recognition. This system would use deep neural networks to analyze user behavior patterns and authorization requests. It would establish a baseline of normal behavior for users and entities within a network and continuously monitor for deviations from this baseline. When anomalies or suspicious access

www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

requests are detected, the system could trigger alerts or automatically enforce access controls to prevent unauthorized access. The advantage of a deep learning-based approach is its ability to adapt and learn from evolving threats, making it more effective in detecting novel and sophisticated malicious activities.

ADVANTAGES

The proposed deep learning-based access control system offers several advantages for substantial authorization recognition in machine learning-driven detection of malicious activities. It leverages deep neural networks to continuously analyze and adapt to user behavior patterns and authorization requests, providing a dynamic and context-aware approach to security. This adaptability enhances the system's ability to detect novel and sophisticated threats effectively, reducing false positives and false negatives. Furthermore, its deep learning foundation enables it to learn from evolving threat landscapes, improving over time without extensive manual rule management. This not only enhances the accuracy of threat detection but also reduces the burden on security teams, making it a more efficient and proactive solution for safeguarding critical assets and data

SYSTEM ARCHITECTURE

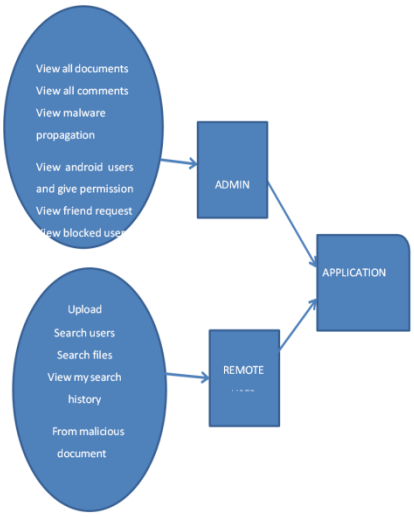

Fig. SYSTEM ARCHITECTURE

IV. IMPLEMENTATION

MODULES

- ADMIN
- REMOTE USER

MODULE DESCRIPTION

ADMIN

The Admin Module provides the following three tabs for viewing data.

- Master User List
- Auto-Created Employees User List
- Employee List

Master User List Tab

The Master User List includes all Users existing in the Clear View portal. You can search for Users, change your view from all users, to active, inactive, or deleted Users, and edit and delete Users. Also, Users are added to the Master User **List if the following occurs:**

- You create an Employee record in the Employee List and assign it to an active or inactive user or a newly-created User.
- A User in the Auto-Created Employees User List is edited and made active.

Auto-Created Employees User List Tab

If the system does not recognize an Employee code in an import file, a User is automatically assigned to the new Employee code. The User is then listed in the Auto- Created Employees User List tab. The Users in this tab are all inactive. You can create a User, and you can either edit a User and make it active, or assign it to a different Employee. Once the User is active, it is listed in the Master User List.

Employee List Tab

The Employee List tab lists all the Employees in the system along with their Retailers. The Users assigned to each Employee are also shown in the list. You can change your view from all to assigned or unassigned users. You can create an employee record and assign a user to it, and also edit and delete an employee record.

This module is used to manage users, roles, policies and directories. User Accounts

User accounts can be tied to directories (Active Directory/LDAP) or can be simply local accounts. Accounts that are tied to Active Directory/LDAP always authenticate against the Active Directory/LDAP server, so the password is always in sync.

www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

Managing Users

Users can be managed through the Admin > Manage Users page (Users & Roles > Manage Users for Password Server) by users with administrative permissions. The paragraphs below describe various functions that can be accessed through the Manage Users page.

Enable / Disable Users:

- A user account can be enabled or disabled through the by selecting the Enable User or Disable User option in the Actions drop-down list (the change will then be reflected in the Status column).
- A user who is disabled cannot log into the system, and the message The account is currently disabled is displayed if they try to do so. Exception: if an AD Guest account is enabled (not recommended), user authentication will be allowed if a securely encrypted connection can be established.

Force Re-Authentication: to force users to login / re-authenticate again, disable the user(s) and then re- enable. Once the user makes a request again they will be asked to login / re-authenticate, and will be brought back to their original page or section where they were working.

Set Password:

- The administrator can set a user's password without knowing the user's old password through the Set Password action in the Actions drop-down. The password requirements do not have to follow the User's policy in this location.
- The administrator can also set a user's roles (Set Roles), delete a user (Delete User), force a user to change their password upon the next login (Expire User Password) in the same drop-down.

Add Users:

- Users can be added using the actions above the users grid. The Add New User action adds a user whose information is specified and stored in the application's database. Import users from an active directory/LDAP server:
- Leads to the Manage Directories page, from where Import Users action in the Actions drop- down retrieves user information from a remote directory and then allows the administrator to choose which users to import into the application.

Update User From Directory:

- Directory users' information can be updated later on using the Update User from Directory action in the Actions drop-down on the Manage Users page. However, if all users in a directory need their information updated, it is faster to use the Update Users action in the Actions drop-down on the Manage Directories page.

View / Edit User Details:

- Also on the Manage Users page, a user's personal information, such as their display name, email address and phone number, can be viewed by clicking on their username and can be edited by clicking on the edit link beside the username. The edit link also allows setting the user's policy. Policies are described in detail in the Policy Administration section.

Unlock User:

- An administrator can perform an Unlock User action in the Actions drop-down on the Manage Users page: if lockouts are enabled and the user has been locked out. User lockouts are not enabled in the default policy, but can be changed in the manage policy section.

REMOTE USER: Remote User Module involves leveraging advanced authorization recognition techniques within machine learning-powered systems to identify potentially malicious activities conducted by remote users.

Remote users, accessing systems and data from external locations, pose unique security challenges. While legitimate remote access is essential for modern organizations, it also introduces risks such as unauthorized access, data breaches, and malicious activity.

V. SCREENSHOTS

Index in Cosmos
JUNE 2025, Volume 15, ISSUE 2
UGC Approved Journal

www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

FIG-1 Home page

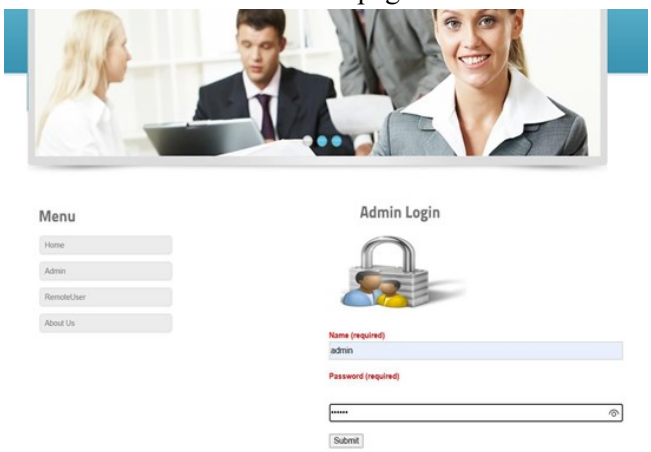

FIG-2 Admin login page

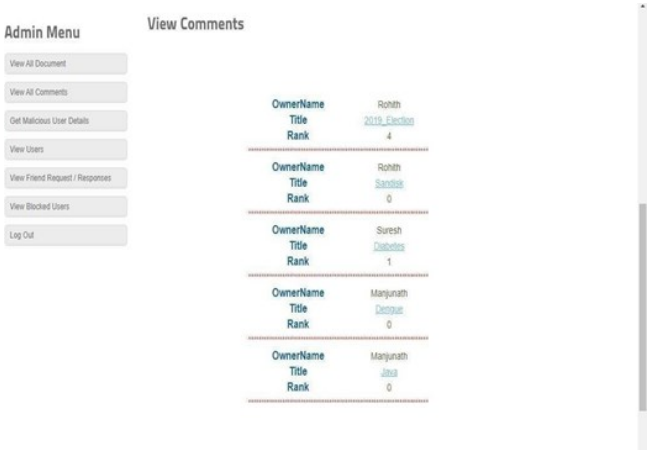

FIG-3 view comments

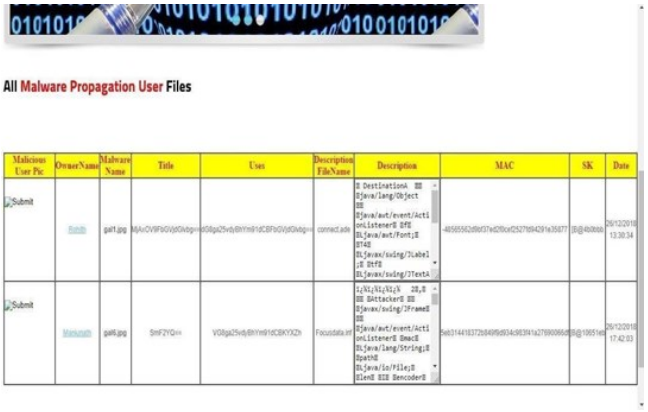

FIG-4 All malware propagation user files

FIG-5 Remote user menu search friends

FIG-6 View my search history

Index in Cosmos
JUNE 2025, Volume 15, ISSUE 2
UGC Approved Journal

www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

FIG-7 Malicious content

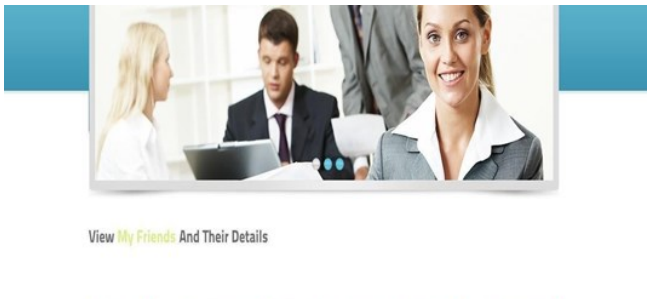

FIG-8 view my friend and their details

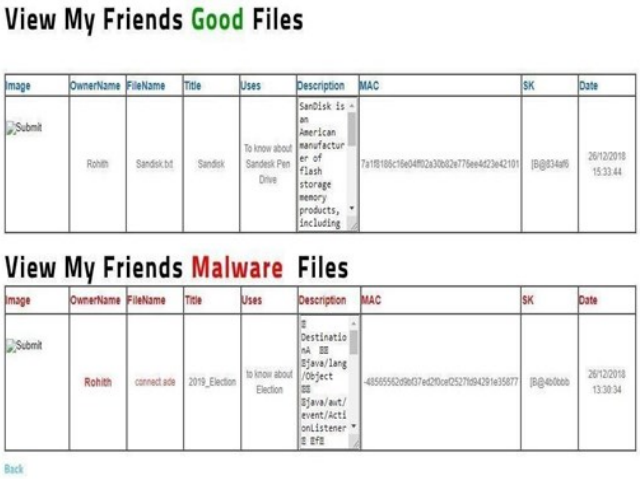

FIG-9 View my friend good files and malware files

VI. CONCLUSION

CONCLUSION

In this paper, we have shown that it is possible to reduce the number of permissions to be analyzed for mobile malware detection, while maintaining high effectiveness and accuracy. SIGPID has been designed to extract only significant permissions through a systematic, 3- level pruning approach. Based on our dataset, which includes over 2,000 malware, we only need to consider 22 out of 135 permissions to improve the runtime performance by 85.6% while achieving over 90% detection accuracy. The extracted significant permissions can also be used by other commonly used supervised learning algorithms to yield the F-measure of at least 85% in 55 out of 67 tested algorithms. SIGPID is highly effective, when compared to the state-of-the-art malware detection approaches as well as existing virus scanners. It can detect 93.62% of malware in the data set, and 91.4% unknown/new malware.

FUTURE SCOPE

Future research may focus on developing more sophisticated machine learning algorithms capable of recognizing complex authorization patterns and anomalies. Deep learning approaches, including deep neural networks and reinforcement learning, could improve the accuracy and efficiency of detecting malicious behavior in real-time.

As machine learning models become more complex, there is a growing need for explainable AI techniques that can provide insights into how decisions are made. Future systems may prioritize interpretability, enabling security analysts to understand the rationale behind the detection of malicious activities and the factors contributing to risk scores.

REFERENCES

1. IDC, “Smartphone os market share, 2017 q1.” [Online].
2. Available: https://www.idc.com/promo/smartphone-market-share/os
3. Statista, “Cumulative number of apps downloaded from the google play as of may 2016.” [Online]. Available: https://www.statista.com/ statistics/281106/number-of- android-appdownloads-from-google-play/
4. G. Kelly, “Report: 97% of mobile malware is on android. this is the easy way you stay

safe," in Forbes Tech, 2014. [4] G. DATA, "8,400 new android malware samples every day." [Online]. Available: https://www.gdatasoftware.com/blog/2017/04/29712-8-400- new-android- malware samples-every-day

5. Symantec, "Latest intelligence for march 2016," in Symantec Official Blog, 2016.
6. M. Grace, Y. Zhou, Q. Zhang, S. Zou, and X. Jiang, "Risk ranker: scalable and accurate zeroday android malware detection," in Proceedings of the 10th international conference on Mobile systems, applications, and services. ACM, 2012, pp. 281–294.
7. A. P. Felt, E. Chin, S. Hanna, D. Song, and D. Wagner, "Android permissions demystified," in Proceedings of the 18th ACM conference on Computer and communications security. ACM, 2011, pp. 627–638.
8. W. Enck, P. Gilbert, S. Han, V. Tendulkar, B.-G. Chun, L. P. Cox, J. Jung, P. McDaniel, and A. N. Sheth, "Taintdroid: an information flow tracking system for real time privacy monitoring on smart phones," ACM Transactions on Computer Systems (TOCS), vol. 32, no. 2, p. 5, 2014.
9. D. Arp, M. Spreitzenbarth, M. Hubner, H. Gascon, K. Rieck, and ¨ C. Siemens, "Drebin:Effective and explainable detection of android malware in your pocket," in Proceedings of the Annual Symposium on Network and Distributed System Security (NDSS), 2014.
10. C. Yang, Z. Xu, G. Gu, V. Yegneswaran, and P. Porras, "Droidminer: Automated mining and characterization of fine-grained malicious behaviors in android applications," in European Symposium on Research in Computer Security. Springer, 2014, pp. 163–182.
11. M. Lindorfer, M. Neugschwandtner, L. Weichselbaum, Y. Fratantonio, V. Van Der
12.
13. Veen, and C. Platzer, "Andrubis–1,000,000 apps later: A view on current android malware behaviors," in Building Analysis Datasets and Gathering Experience Returns for Security (BADGERS), 2014 Third International Workshop on. IEEE, 2014, pp. 3– 17.
14. W. Yang, X. Xiao, B. Andow, S. Li, T. Xie, and W. Enck, "Appcontext: Differentiating malicious and benign mobile app behaviors using context," in Software engineering (ICSE), 2015 IEEE/ACM 37th IEEE international conference on, vol. 1. IEEE, 2015, pp. 303–313.
15. S. Wang, Q. Yan, Z. Chen, B. Yang, C. Zhao, and M. Conti, "Textdroid: Semantics-based detection of mobile malware using network flows."
16. Z. Li, L. Sun, Q. Yan, W. Srisa-an, and Z. Chen, "Droidclassifier: Efficient adaptive mining of application-layer header for classifying android malware," in International Conference on Security and Privacy in Communication Systems. Springer, 2016, pp. 597– 616.
17. Z. Chen, Q. Yan, H. Han, S. Wang, L. Peng, L. Wang, and B. Yang, "Machine learning based mobile malware detection using highly imbalanced network traffic," Information Sciences, 2017.