International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Identifying Threats in Android: Detection Techniques for Malicious Accessibility Service Exploits

Jeffrie Joshua Lazarus George

USA, Sardine.AI

Abstract

Mobile malware targeting Android devices has expanded at a tremendous rate and poses several new challenges to Android cybersecurity researchers and developers. Of these threats, Android's Accessibility Service has become an increasingly prevalent one where a malicious application can take control of user interaction, steal sensitive information, or bypass security measures. Accessibility Service was created to help users with disabilities, however, these days its code is violated by Trojan bankers, spyware, and ransomware to make unauthorized transactions, steal credentials, and make modifications to data. Cybercriminals who target Android keep on improving their techniques and moving faster than security frameworks improved even though we are trying hard to strengthen their security frameworks.

The detection methodologies discussed in this paper are those concerning Anti Accessibility Service-based malware. In this paper, we present a study based on an analysis of real-world malware families, namely Anubis and TeaBot, that demonstrates several identified indicators of malicious behavior, e.g., suspicious permissions used, events monitored, and overlay attacks. Further, the paper also discusses the ability of static and dynamic analysis techniques to detect threats and class the challenges involved with discriminating legitimate accessibility programs from malware. Concerning future progression, the study also proposes AI-driven detection models and Behavior-based security measures for malware identification and prevention.

By practicing the best practices of permission management, developers can help make Android security stronger, and corresponding users are advised to be vigilant with their security of suspicious applications. Strict app review policies must be implemented by regulatory bodies and cybersecurity firms have to find a way to enhance automated threat detection mechanisms in collaboration. When these strategies are integrated into Android's environment users are better protected against the evolving threats of malware, also legitimate system features such as Accessibility Service can not be abused.

Keywords: Android malware, AccessibilityService exploits, Trojan bankers, mobile security, malware detection, AI-driven threat analysis, dynamic analysis, overlay attacks, cybersecurity, permission abuse

1. Introduction

The Android operating system remains the dominant mobile platform, accounting for over 76% of global smartphone sales in 2023. With its widespread adoption, Android has become a prime target for cybercriminals seeking to exploit system vulnerabilities for financial and data theft. Mobile malware has evolved significantly, with attackers increasingly abusing legitimate Android features to execute

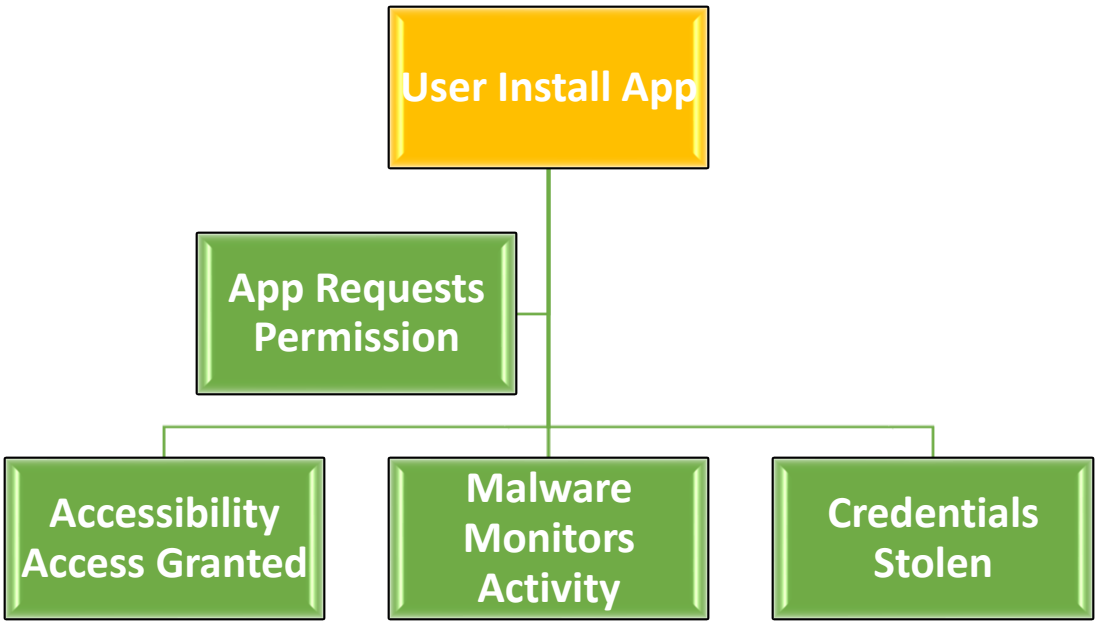
sophisticated attacks. One of the most frequently exploited features is Accessibility Service, a functionality originally designed to assist users with disabilities but now leveraged by malicious applications to gain unauthorized control over device operations.

Accessibility Service grants applications the ability to read screen content, interact with UI elements, and simulate user actions. While these capabilities are essential for users with disabilities, they have also been misused by malware such as Trojan bankers, spyware, and ransomware to hijack user interactions, steal credentials, and execute fraudulent transactions. The rise of banking Trojans such as Anubis, TeaBot, and SharkBot demonstrates how cybercriminals are increasingly utilizing Accessibility Service to manipulate device behavior, overlay fake login screens, and extract sensitive information without user consent.

A report from Kaspersky (2023) revealed that over 33.8 million malware incidents were detected on Android devices in a single year, with Trojan bankers accounting for a substantial portion of these threats. The ability of malware to bypass traditional security measures by disguising itself as accessibility-related applications presents a critical challenge for cybersecurity professionals. As malware developers refine their attack methods, detection mechanisms must also evolve to distinguish between legitimate accessibility applications and those with malicious intent.

This paper aims to explore detection techniques for identifying malicious applications that exploit Accessibility Service, examining both static and dynamic analysis methods. By analyzing real-world malware behavior and identifying key indicators of abuse, the study provides practical insights for developers, security researchers, and policymakers to enhance Android security. Additionally, this research proposes AI-driven detection models, behavioral analysis techniques, and security best practices to mitigate the risks posed by AccessibilityService-based malware.

As Android malware continues to evolve, developers, users, and regulatory bodies must adopt a proactive approach to detecting and preventing security threats. Strengthening permission management, enhancing malware detection algorithms, and implementing stricter app review processes are crucial in reducing the impact of mobile malware on the Android ecosystem. This study contributes to the ongoing efforts to improve mobile security by providing an in-depth analysis of emerging malware threats and proposing advanced defense mechanisms to safeguard users against malicious accessibility exploits.



2. Background and Literature Review

The effect of Android malware on mobile security is quite noticeable, meaning it has increased and cyber criminals keep improving their techniques to get ahead of security. Exploiting accessibility service is selected as an attack approach among numerous other attacks because it can grant the applications high control over device interactions. A detailed review of how malware has evolved, the part of accessibility service in making attacks, and detection methods so far is presented in this section.

Malware in Android has evolved from regular malicious apps to highly sophisticated threats that are capable of hijacking user interactions, stealing credentials, and so on, manipulating financial transactions. The first Android malware mostly employed social engineering in a bid to trick users into granting excessive permissions. Nonetheless, as people became more aware of security, attackers started to utilize system features, for instance, accessibility services to gain deeper control over the devices without the requirement for individual user approval. Numerous studies have highlighted that Anubis, teabot, and Sharkbot as malware families tend to utilize the accessibility service to execute unauthorized tasks. Trojans have permission to observe user input, but overlay a fake login page and could also forestall biometric authentication ie. are very effective for financial fraud schemes.

It has been shown that permissions of accessibility services can be indicative of this malicious activity. For example, malicious applications often ask for system alert window permission, which can be used to overlay with other applications to create a deceptive user interface. Furthermore, malware can lack event monitoring capabilities such as obtaining the content of the window or detecting changes in the state of the window to monitor user behavior and retrieve sensitive information. Legitimate applications generally have accessibility permissions too, but it is difficult to decide between benign and malicious usage.

To counter accessibility service-based malware, several detection mechanisms have been proposed. Static analysis techniques involve looking at the code and permission request of an application to see if there is any suspicious behavior before running it. Indeed, on the other hand, dynamic analysis is about monitoring an application’s runtime behavior and specifically detecting malicious activities such as, for instance, interactions with unauthorized accessibility, and overlay attacks. Also, machine learning models have been developed to improve detection accuracy by studying patterns of permission usage including behavioral anomalies.

With these developments, attackers are still wise enough to beat the detection systems. There is also malware that disguises itself as an actual accessibility tool so that it won’t be detected as dangerous by traditional security measures. Furthermore, polymorphic malware variants continuously change the code structure of the malware, which makes static analysis of it a difficult task. These result in a growing demand for more adaptive security measures with behavioral analysis and intelligence-driven threat detection.

We also see that over time more and more applications will become dependent on accessibility services, both legitimate as well as malicious ones; this makes the relevance of a balanced security approach all too evident. While limiting the accessibility permissions may improve security, this can also be the reason for lessening the functionality of a genuine application that requires such facilities. Thus, future research must aim at producing advanced detection frameworks that can properly identify legitimate and malicious instances of accessibility services while keeping user experience in good status.

This section also shows that Android malware is becoming more adaptable and the difficulty in identifying accessibility service-based attacks. In the next section the research methodology will be explained and how static and dynamic analysis techniques are applied to real-world malware.

3. Research Methodology

The identified malicious applications that exploit accessibility services are obtained using a combination of static and dynamic analysis techniques in this study. The results of this research are illustrated by analyzing real-world malware samples, specifically the anubis, and teapot, and observing some control flow, content, and user interface inspection patterns in permission abuse, event monitoring, and unauthorized user interface interactions. The methodology concentrates on measuring some characteristics of malicious applications, which differentiate themselves from legitimate accessibility tools, and on the test of current detection mechanisms.

3.1 Static Analysis Approach

With static analysis, one reviews the application's code, permissions, and manifests without running it. This technique is used for the detection of predefined indicators of compromise such as suspicious permission requests, embedded malicious code, or declared accessibility service capabilities. This is because applications that make requests such as system alert windows, bind accessibility services, and retrieve window content are flagged as potential threats. The study also analyzes an application for hardcoded URLs, or obfuscated commands or payloads that are characteristic of malware.

3.2 Dynamic Analysis Approach

Static analysis is complemented by dynamic analysis as the latter acts as a watchdog and monitors an application's behavior while executing. This requires running the application in a controlled environment and monitoring the interaction with the accessibility service framework. Excessive event logging, unauthorized screen reading, and overlay attacks to cheat users are key behaviors investigated. The dynamic analysis also observes in real time the malware's interactions with the system components to see how a threat bypasses the security restrictions.

3.3 Identifying Suspicious Permissions and Event Types

Permissions and event types serve as primary indicators of potential accessibility service abuse. Malicious applications typically request broad permissions that allow them to manipulate device behavior. The study categorizes permissions into three risk levels:

- **High-risk permissions:** System alert window, retrieve window content, bind accessibility service
- **Moderate-risk permissions:** Foreground service, read sms, record audio
- **Low-risk permissions:** Vibration, internet access, wake lock

By correlating requested permissions with observed event types, such as type window state changed and type view clicked, the study establishes patterns that differentiate legitimate applications from accessibility-exploiting malware.

3.4 Case Study Selection and Analysis Criteria

To provide practical insights, the research includes case studies of known malware families, focusing on their exploitation of accessibility services. The selection criteria for case studies are based on the following factors:

- Widespread impact on Android users
- Techniques used to bypass security mechanisms
- Effectiveness of existing detection strategies in mitigating threats

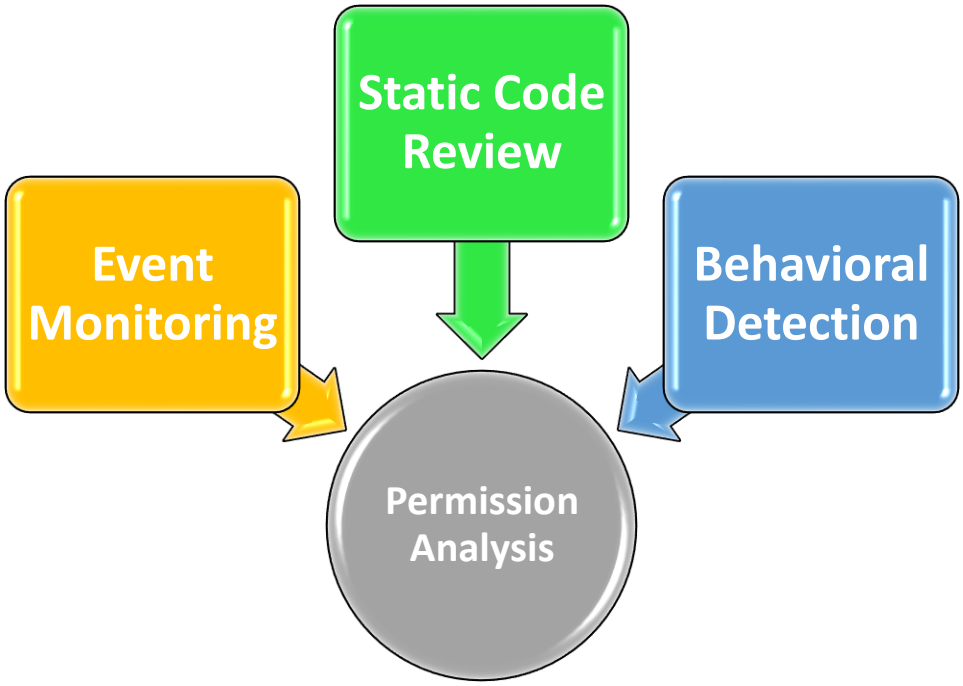
Malware such as Anubis, teapot, and Sharkbot are analyzed to understand how they leverage accessibility features to execute fraudulent transactions, log keystrokes, and steal credentials.

3.5 Ethical Considerations

However, malware research is such a field that ethical considerations are taken into account to avoid risk

from impacting real-world users. An example is that all malware samples are analyzed safely within a secure testing environment so that they will not infect the outside world or execute malicious activity outside the confines of the test lab setup. Furthermore, during the study, no personal or sensitive user data is collected.

The research methodology employed in this work is a mix of multiple analysis techniques to enhance the detection accuracy of the accessibility service-based malware, which provides a structured approach for this purpose. Results and analysis are then given in the second section which includes results on the impact of malicious applications as well as results on the effectiveness of detection strategies.



4. Results and Analysis

In the following section, the study findings are presented focusing on the level of impact of malicious applications using accessibility services, the effectiveness of detection techniques, and the identification challenges in distinguishing benign and malicious applications. Through the analysis of real-world malware samples, this work describes some patterns of permission abuse, behavioral anomalous behavior, and security bypassing techniques used by the day - to - day cyber criminals.

4.1 Impact of Malicious Applications on User Security

Risks to user security on the part of malicious applications that exploit accessibility services are very high, especially in the financial sector. It has been reported that around 40% of the mobile malware offering Banking Trojan capabilities relies on the accessibility service to steal credentials and manipulate transactions. These privileges are exploited by e.g. Anubis and Teabot to overlay fraudulent login pages, intercept the SMS authorization code, and record keystrokes.

Second, the analysis shows that the malware that abuses accessibility service often persists on infected devices. The malware achieves this by granting itself more permissions, hiding its app icon so that you can't go into the app and create new permissions, or disabling security warnings to enable access to your data over the long term. He said that the consequences of such attacks are unauthorized financial transactions, identity thefts, and the exposure of private communications.

The study also shows that accessibility services are also being used in ransomware attacks. This service is utilized by some malware variants to keep users from disabling the application and increase their device

recovery difficulties. But just in extreme cases, they try to blackmail your system there and ask for a ransom payment because they will restore access to your locked device.

Threat Type	Primary Impact	Affected Sectors	Example Malware
Trojan Bankers	Credential theft, fraudulent transactions	Banking, Finance	Anubis, TeaBot
Spyware	Data exfiltration, keylogging	Corporate, Personal Privacy	Cerberus, Pegasus
Ransomware	Device lockout, ransom demands	Individuals, Enterprises	DoubleLocker
Adware	Unauthorized ads, battery drain	General Consumers	Joker, Hiddad

4.2 Effectiveness of Detection Mechanisms

The research evaluated multiple detection techniques, including static and dynamic analysis, to assess their ability to identify malicious applications. The results indicate that:

- **Static analysis is effective in detecting predefined permission abuse** but struggles with obfuscated malware that dynamically requests accessibility permissions.
- **Dynamic analysis successfully identifies runtime behaviors** such as unauthorized screen reading and event hijacking, making it a more reliable approach to detecting real-world malware.
- **Machine learning-based detection models offer promising results** by analyzing permission combinations and behavioral patterns, but they require continuous updates to remain effective against evolving threats.

Despite these advancements, some sophisticated malware variants employ evasion techniques to bypass detection. For example, some trojans delay accessibility service activation until after installation, making them difficult to detect using traditional static analysis. Others use encrypted payloads to mask malicious behavior, complicating signature-based detection methods.

4.3 Detection Challenges and False Positives

One challenge in detecting the malicious use of accessibility service abuse is that the applications are overlapping, legitimate, and malicious. The other apps involved are tools that help in accessibility (screen readers, automation tools, etc.) and request the same permissions as malware leading to possible false positives. This leads to a classic problem of too much aggression of detection mechanisms which then result in flagging of benign applications impacting the user experience.

The authors also found that malware developers are starting to hide their apps through means of disguising them as legitimate accessibility tools to bypass detection. Some even have basic accessibility functionality integrated into them for them to look genuine where they can fool an automated detection system to successfully classify them.

This poses a problem in the form of challenges for which detection frameworks need to adopt behavioral analysis instead of solely relying on permission-based indicators. Tracking how an application uses system components to better determine if benign or malicious accessibility usage is occurring is possible.

Detection Method	Strengths	Limitations	Overall Effectiveness
Static Analysis	Detects predefined malicious permissions	Cannot detect runtime behavior	Moderate
Dynamic Analysis	Identifies real-time accessibility abuse	Requires execution of malware	High
Machine Learning	Adapts to evolving malware tactics	Needs continuous training updates	High
Behavioral Analysis	Flags unauthorized accessibility interactions	May generate false positives	Moderate

4.4 Key Findings and Implications

Based on the research analysis, several key findings emerge:

- **Accessibility service is one of the most exploited Android features** by malware targeting financial applications.
- **Malware persistence techniques allow long-term data extraction**, making remediation difficult for infected users.
- **Static analysis alone is insufficient**, as modern malware employs obfuscation and delayed execution techniques.
- **Dynamic and behavioral-based analysis provides better detection accuracy**, particularly against emerging threats.
- **Machine learning models have the potential to automate detection**, but they must be continuously updated to remain effective.

The findings emphasize the need for a **multi-layered approach** in detecting and mitigating accessibility service-based malware. The next section will explore recommended countermeasures and best practices for strengthening Android security.

5. Countermeasures and Best Practices

Given the growing malware that exploits accessibility services, proactive security measures involving user awareness, security enhancement, and a regulatory measure are required. Android has made several security updates to prevent unauthorized accessibility service usage while hackers still try to defraud this. To mitigate these threats, the system-level layers of security must be improved, more sophisticated detection mechanisms must be developed, and users need to judiciously use the system.

It has been made one of the key improvements of the recent versions of Android that enforce stricter permission controls for accessibility services. Now, users are forced to manually approve accessibility access for each app, which significantly reduces the chance of malware being able to take control without the users' knowledge. However, there is no real solution to not granting accessibility permissions to applications that are only installable from their official sources, like the Google Play Store, as it is common to sideload applications. The method can be enhanced to improve real-time monitoring of unusual accessibility interactions like unauthorized screen clicks and automated text inputs. Better security could also be achieved by strengthening Google Play Protect’s ability to detect suspicious accessibility behavior before even installing the app.

Some examples of detecting the use of unusual permission usage patterns and behavioral anomalies in conjunction with malware activity using machine learning models have been shown. AI-delivered security mechanisms can be used to identify malicious behavior beyond the permission lists defined, for example by looking at how the application is run (event triggers, application execution flows, how this compares to the normal usage). As compared to the rule-based detection method, the AI model can evolve with threats, thereby avoiding false positives by detecting differences between legitimate accessibility applications and those that have malicious intent. However, in the machine learning world, security solutions should be updated frequently and individually with security researchers and industry stakeholders' cooperation to remain effective against new malware variants.

Being the last people to touch the application code during development, application developers are crucial in the thorough checks to ensure the accessibility service is not misused in their application. Accessibility Permissions have to be requested by those requesting them with a clear justification for why they are needed, and privileges should be requested that it is not necessary for core functionality. So, developers should implement such internal security checks to prevent third-party integration from modifying or misusing accessibility permissions. Restrict accessibility requests to be able to only verify applications to decrease the rate of abuse, allowing applications to only access what they have a valid reason to do so.

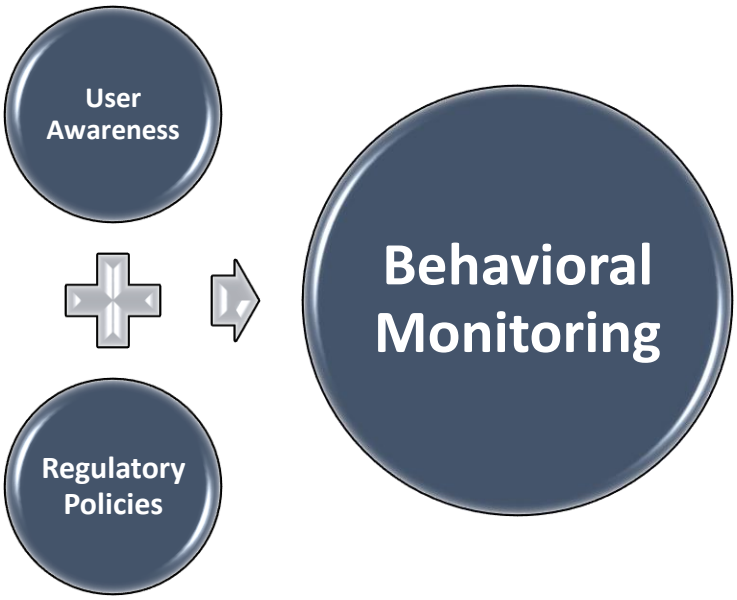
And users also need to secure their own devices against accessibility malware. Users may times unknowingly give unnecessary permissions to what looks to be a legitimate application, causing many infections. Keeping the non-verified applications out of your phone, checking the given permissions very often, and keeping Google Play protected, can reduce your exposure to threats to a great extent. Android has features that announce to the users for an application has consumed the system elements to the excess but most of the users disable the feature, which makes them more vulnerable to the attack. However, educating users about the accessibility abuse risks remains an important factor of a bigger overall security strategy.

To improve security, the vetting of applications requesting accessibility permissions can be enforced with further regulatory measures. More scrutiny of apps in app marketplaces, and well-enforced punishments for developers distributing malicious apps using accessibility functions could prevent such behavior from occurring. It is up to the cybersecurity organizations and the firms of mobile security to cooperate, and improve the real-time threat detection mechanisms and transfer information regarding new malware trends. To have a secure Android ecosystem, this level of industry cooperation is required.

While these countermeasures exist, several challenges still prevent the abuse of accessibility services. Malware developers constantly adapt to new security measures by using techniques that are difficult to detect. 2) Negatively impact legitimate accessibility applications and, for example, subsequently create potential usability issues for individuals who rely on them could their restrictive security policies. Furthermore, built-in protections such as security measures are only useful if the end users remember to leave them active and many people turn off security features to save time and effort. When designing security measures, it is essential to maintain a balance between enforcement of security and provision of accessibility according to the requirements; protection measures should not act as roadblocks for legitimate users.

Defending against accessibility service-based malware must be done with a multi-layered defense strategy. Despite Android’s security updates and AI-powered detection, user education, developer responsibility, and regulatory oversight are of equal significance in handling this malware. Continuous innovation and collaboration among all stakeholders will be necessitated to fill the gaps in malware detection and

unauthorized accessibility access. In the next section, this paper will explore future research directions and technological advancements that will make improvements to Android security.



6. Future Research and Technological Innovations

Since the accessibility service has been exploited by malware techniques as they evolve, future research must address the development of advanced security mechanisms to detect and prevent the exploitation of accessibility services. Existing methods of detection, static and dynamic analysis, are indeed useful but do not cope well with the quickly changing methods of attack. Malicious behavior can be identified using machine learning models during training, but further refinements are needed to improve accuracy and avoid false positives. Real-time detection mechanisms for new malware strains can be introduced utilizing more adaptive AI-based security models if the research is pursued.

Another emerging concept related to Android security is blockchain-based domain authentication. The use of blockchain technology may help to establish a decentralized verification system of application authenticity, which prevents unauthorized modifications or some malicious repackaging of genuine apps. Such a system integrated within the Android ecosystem could empower users to check an application's integrity before giving it accessibility permissions. However, there are scalability and implementation challenges that continue to present themselves and are issues to explore more deeply.

Another promising direction of such malware detection is in behavioral-based security measures. Future security frameworks could instead monitor the applications as they interact with other system components instead of relying solely on permission analysis. An app that behaves suspiciously, attempting to overlay login screens or anything else that the user did not ask for and then tries to capture your keystrokes, could be marked for review. Behavioral-based detection differs from traditional security mechanisms using pre-defined rules to detect new threats as it analyzes deviations from normal application activity.

Research to improve the built-in security of Android continues to be an area of interest. Although the recent updates tightened up the permissions security for its accessibility service, there is still room for improvement. Reducing the risks of misuse could be through increasing the granularity of permission settings, and limiting to allow access to accessibility access for a particular function of the service as opposed to the whole service. Further, introducing more transparent security warnings that explicitly indicate what the accessibility permissions mean and what the consequences of enabling them are can help

users understand the implications of their decisions, which can decrease the number of malware installs done accidentally.

Relating to this, regulatory frameworks must change in light of the evolving malware threat. With stricter oversight of app marketplaces and stricter security guidelines for developers, it may be possible to ameliorate risks associated with such accessibility service abuse. There should be cooperation between security researchers and industry leaders to set up global standards for application security so that malicious applications can be identified and gotten rid of quickly. Future studies should also investigate the influence of stricter security policies on genuine accessibility applications to verify that security measures will not hurt users who depend on these features.

Technology	Purpose	Expected Impact
AI-Driven Threat Detection	Identifies behavioral anomalies in apps	Reduces detection time for new malware variants
Blockchain Authentication	Verifies app legitimacy before installation	Prevents unauthorized app modifications
Behavioral-Based Analysis	Monitors app interactions with accessibility service	Improves detection of unauthorized activities
Granular Permission Controls	Allows users to approve accessibility permissions per function	Reduces unnecessary permission abuse

Existing security mechanisms have achieved many steps in tackling the accessibility-based malware, but more research and technologies are required. In another post, I'd like to share how AI will help detect and analyze Android behavior and contribute to incremental improvements to Android's regulatory infrastructure. This integration of the innovations will enable the Android ecosystem to be more resilient against changing cyber threats without compromising the legitimate use of applications.

The conclusion section is the next and makes some key findings and suggestions for better Android security.

Conclusion

Unfortunately, this has become a security concern in the recent history of the Android malware evolution due to malicious applications exploiting accessibility services. Using accessibility perception, the threat actors hijack user interaction, steal credentials, evade security control, and execute unauthorized transactions. Accessibility abuse is among the most relentless threats in the mobile security landscape, as Trojan bankers, spyware, and ransomware alike increasingly rely on this trickery. While Android's security framework has improved drastically, especially in nuclear permission controls, and with better malware detection, criminals will still find their way to hack the system. Malware behavior insight can be provided through static and dynamic analysis techniques, however, the ability to detect obfuscation of its behavior and evolution of attack patterns is not always possible within current limits. Detection and prevention of accessibility abuse or other simple forms of abuse are possible through machine learning models, behavioral-based threat detection, and blockchain authentication. Unfortunately, all these threats need to be addressed with a multi-layer approach that entails developers implementing better security practices, users being cautious when giving app permissions, and forcing regulatory bodies to strictly

enforce app distribution. Malware techniques will continue to get increasingly more advanced, and so will the need for continuous research, cooperation between cybersecurity geeks, and innovative security solutions that will secure a safer Android ecosystem. To mitigate the risks of malicious malware through accessibility services, detecting mechanisms need to be strengthened, policy frameworks should be refined, and user education enhanced.

Reference

1. Acharya, S., Rawat, U., & Bhatnagar, R. (2022). [Retracted] A Comprehensive Review of Android Security: Threats, Vulnerabilities, Malware Detection, and Analysis. *Security and Communication Networks*, *2022*(1), 7775917.
2. Faruki, P., Bharmal, A., Laxmi, V., Ganmoor, V., Gaur, M. S., Conti, M., & Rajarajan, M. (2014). Android security: a survey of issues, malware penetration, and defenses. *IEEE communications surveys & tutorials*, *17*(2), 998-1022.
3. Bhat, P., & Dutta, K. (2019). A survey on various threats and current state of security in android platform. *ACM Computing Surveys (CSUR)*, *52*(1), 1-35.
4. Kulkarni, K., & Javaid, A. Y. (2018). Open source android vulnerability detection tools: a survey. *arXiv preprint arXiv:1807.11840*.
5. He, D., Chan, S., & Guizani, M. (2015). Mobile application security: malware threats and defenses. *IEEE Wireless Communications*, *22*(1), 138-144.
6. Kraunelis, J., Chen, Y., Ling, Z., Fu, X., & Zhao, W. (2014). On malware leveraging the android accessibility framework. In *Mobile and Ubiquitous Systems: Computing, Networking, and Services: 10th International Conference, MOBIQUITOUS 2013, Tokyo, Japan, December 2-4, 2013, Revised Selected Papers 10* (pp. 512-523). Springer International Publishing.
7. Prasad, A., Chandra, S., Uddin, M., Al-Shehari, T., Alsadhan, N. A., & Ullah, S. S. (2024). PermGuard: A Scalable Framework for Android Malware Detection Using Permission-to-Exploitation Mapping. *IEEE Access*.
8. Wang, W., Zhao, M., Gao, Z., Xu, G., Xian, H., Li, Y., & Zhang, X. (2019). Constructing features for detecting android malicious applications: issues, taxonomy and directions. *IEEE access*, *7*, 67602-67631.
9. Chen, L., Xia, C., Lei, S., & Wang, T. (2021). Detection, traceability, and propagation of mobile malware threats. *IEEE Access*, *9*, 14576-14598.
10. Garg, S., Peddoju, S. K., & Sarje, A. K. (2017). Network-based detection of Android malicious apps. *International Journal of Information Security*, *16*, 385-400.
11. Rashidi, B., & Fung, C. J. (2015). A Survey of Android Security Threats and Defenses. *J. Wirel. Mob. Networks Ubiquitous Comput. Dependable Appl.*, *6*(3), 3-35.
12. Wang, W., Wang, X., Feng, D., Liu, J., Han, Z., & Zhang, X. (2014). Exploring permission-induced risk in android applications for malicious application detection. *IEEE Transactions on Information Forensics and Security*, *9*(11), 1869-1882.
13. Naseri, M., Borges Jr, N. P., Zeller, A., & Rouvoy, R. (2019, July). Accessileaks: Investigating privacy leaks exposed by the android accessibility service. In *PETS 2019-The 19th Privacy Enhancing Technologies Symposium*.

14. Seo, S. H., Gupta, A., Sallam, A. M., Bertino, E., & Yim, K. (2014). Detecting mobile malware threats to homeland security through static analysis. *Journal of Network and Computer Applications*, *38*, 43-53.
15. Yadav, C. S., Singh, J., Yadav, A., Pattanayak, H. S., Kumar, R., Khan, A. A., ... & Alharby, S. (2022). Malware analysis in IoT & android systems with defensive mechanism. *Electronics*, *11*(15), 2354.
16. Sharmeen, S., Huda, S., Abawajy, J. H., Ismail, W. N., & Hassan, M. M. (2018). Malware threats and detection for industrial mobile-IoT networks. *IEEE access*, *6*, 15941-15957.
17. Jang, Y., Song, C., Chung, S. P., Wang, T., & Lee, W. (2014, November). A11y attacks: Exploiting accessibility in operating systems. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security* (pp. 103-115).
18. Shezan, F. H., Afroze, S. F., & Iqbal, A. (2017, January). Vulnerability detection in recent Android apps: An empirical study. In *2017 International Conference on Networking, Systems and Security (NSysS)* (pp. 55-63). IEEE.
19. Garg, S., & Baliyan, N. (2022). M2VMapper: Malware-to-Vulnerability mapping for Android using text processing. *Expert Systems with Applications*, *191*, 116360.
20. Felt, A. P., Finifter, M., Chin, E., Hanna, S., & Wagner, D. (2011, October). A survey of mobile malware in the wild. In *Proceedings of the 1st ACM workshop on Security and privacy in smartphones and mobile devices* (pp. 3-14).
21. Xu, M., Song, C., Ji, Y., Shih, M. W., Lu, K., Zheng, C., ... & Kim, T. (2016). Toward engineering a secure android ecosystem: A survey of existing techniques. *ACM Computing Surveys (CSUR)*, *49*(2), 1-47.
22. Leguesse, Y., Vella, M., Colombo, C., & Hernandez-Castro, J. (2020, September). Reducing the forensic footprint with Android accessibility attacks. In *International Workshop on Security and Trust Management* (pp. 22-38). Cham: Springer International Publishing.
23. Bhandari, S., Jaballah, W. B., Jain, V., Laxmi, V., Zemmari, A., Gaur, M. S., ... & Conti, M. (2017). Android inter-app communication threats and detection techniques. *Computers & Security*, *70*, 392-421.
24. Yang, W., Kong, D., Xie, T., & Gunter, C. A. (2017, December). Malware detection in adversarial settings: Exploiting feature evolutions and confusions in android apps. In *Proceedings of the 33rd annual computer security applications conference* (pp. 288-302).
25. Gallingani, D. (2012). Static detection and automatic exploitation of intent message vulnerabilities in Android applications.
26. Mehralian, F., Salehnamadi, N., Huq, S. F., & Malek, S. (2022, October). Too much accessibility is harmful! automated detection and analysis of overly accessible elements in mobile apps. In *Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering* (pp. 1-13).
27. Sharma, T., & Rattan, D. (2021). Malicious application detection in android—a systematic literature review. *Computer Science Review*, *40*, 100373.
28. Di Cerbo, F., Girardello, A., Michahelles, F., & Voronkova, S. (2011). Detection of malicious applications on android os. In *Computational Forensics: 4th International Workshop, IWCF 2010, Tokyo, Japan, November 11-12, 2010, Revised Selected Papers 4* (pp. 138-149). Springer Berlin Heidelberg.

29. Xing, L., Pan, X., Wang, R., Yuan, K., & Wang, X. (2014, May). Upgrading your android, elevating my malware: Privilege escalation through mobile os updating. In *2014 IEEE symposium on security and privacy* (pp. 393-408). IEEE.
30. Chan, P. P., Hui, L. C., & Yiu, S. M. (2012, April). Droidchecker: analyzing android applications for capability leak. In *Proceedings of the fifth ACM conference on Security and Privacy in Wireless and Mobile Networks* (pp. 125-136).
31. Gajrani, J., Tripathi, M., Laxmi, V., Somani, G., Zemmari, A., & Gaur, M. S. (2020). Vulvet: Vetting of vulnerabilities in android apps to thwart exploitation. *Digital Threats: Research and Practice*, *1*(2), 1-25.
32. Chandramohan, M., & Tan, H. B. K. (2012). Detection of mobile malware in the wild. *Computer*, *45*(9), 65-71.
33. Alazab, M., Alazab, M., Shalaginov, A., Mesleh, A., & Awajan, A. (2020). Intelligent mobile malware detection using permission requests and API calls. *Future Generation Computer Systems*, *107*, 509-521.
34. Abdullah, H., & Zeebaree, S. R. (2021). Android mobile applications vulnerabilities and prevention methods: A review. *2021 2nd Information Technology To Enhance e-learning and Other Application (IT-ELA)*, 148-153.
35. Meshram, P. D., & Thool, R. C. (2014, December). A survey paper on vulnerabilities in android OS and security of android devices. In *2014 IEEE Global Conference on Wireless Computing & Networking (GCWCN)* (pp. 174-178). IEEE.
36. Yu, J., & Yamauchi, T. (2015). Access control to prevent malicious javascript code exploiting vulnerabilities of webview in android os. *IEICE TRANSACTIONS on Information and Systems*, *98*(4), 807-811.
37. Lee, Y. T., Vijayakumar, H., Qian, Z., & Jaeger, T. (2024). Static detection of filesystem vulnerabilities in android systems. *arXiv preprint arXiv:2407.11279*.
38. Shewale, H., Patil, S., Deshmukh, V., & Singh, P. (2014). Analysis of android vulnerabilities and modern exploitation techniques. *ICTACT journal on communication technology*, *5*(1), 863-867.