Available online www.ejaet.com

European Journal of Advances in Engineering and Technology, 2021, 8(8):105-109

Research Article ISSN: 2394 - 658X

Exploring Identity Confusion Vulnerabilities in App-in-App Ecosystems

Naga Satya Praveen Kumar Yadati

DBS Bank Ltd
*praveenyadati@gmail.com

ABSTRACT

This paper delves into identity confusion vulnerabilities within app-in-app ecosystems, a critical yet understudied area in mobile security. These vulnerabilities stem from flawed identity checks in WebView, leading to severe consequences such as unauthorized access to sensitive APIs and user data manipulation. We investigate 47 popular super-apps and reveal that all are susceptible to at least one type of identity confusion, categorized into domain name, app ID, and capability confusions. Our findings underscore the necessity for robust identity verification mechanisms and provide insights into mitigating these vulnerabilities.

Key words: identity confusion, app-in-app ecosystems, WebView, mobile security, super-apps, domain name confusion, app ID confusion, capability confusion, Alipay, TikTok

---

INTRODUCTION

The proliferation of app-in-app ecosystems, where super-apps host various sub-apps, has introduced significant security challenges. Super-apps like Alipay and TikTok integrate numerous functionalities, including financial transactions and social interactions, making them attractive targets for adversaries. This study focuses on identity confusion vulnerabilities arising from inadequate identity verification processes in WebView-based environments.

BACKGROUND

2.1 WebView in Mobile Applications

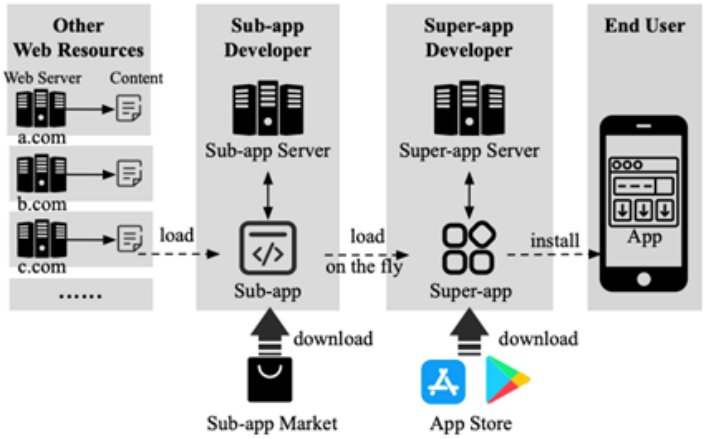
WebView is a crucial component in mobile apps, allowing the display of web content within the app itself. Despite its utility, WebView's integration has been plagued by security issues, particularly in app-in-app ecosystems where multiple sub-apps operate under a super-app's umbrella. These vulnerabilities often stem from improper handling of identity verification between the super-app and its sub-apps.

2.2 Identity Confusion in App-in-App Ecosystems

Identity confusion occurs when the system fails to accurately distinguish between different entities, leading to unauthorized access and operations. In app-in-app ecosystems, this can manifest as domain name confusion, app ID confusion, and capability confusion. These flaws enable attackers to manipulate the system, leading to severe security breaches.

METHODOLOGY

We conducted a comprehensive analysis of 47 popular super-apps to identify and categorize identity confusion vulnerabilities. Our methodology involved static and dynamic analysis of these apps, focusing on their WebView implementations and identity verification mechanisms.



3.1 Tools and Techniques

We utilized a combination of static analysis tools and dynamic testing environments to evaluate the security posture of the apps. Additionally, we developed custom scripts to automate the detection of identity confusion vulnerabilities.

FINDINGS

4.1 Identity Confusion Vulnerabilities in Alipay

Alipay, a widely used super-app, exhibits significant vulnerabilities due to identity confusion. Our analysis revealed two primary issues: domain name confusion and AppID confusion.

4.1.1 Domain Name Confusion

Alipay is vulnerable to domain name confusion due to race conditions in its customized WebView, UCWebView. This flaw allows adversaries to manipulate Alipay's backend servers by exploiting these race conditions.

4.1.2 AppID Confusion

Sub-apps of Alipay suffer from AppID confusion because of a flaw in Alipay's URL whitelist matching. Alipay uses regular expressions for string matching, but many sub-apps assume strict matching, adding domain names directly to their whitelist. This discrepancy allows attackers to exploit the URL whitelist and gain unauthorized access to privileged APIs.

4.1.3 Security Consequences

The security implications of Alipay's identity confusion are severe. Alipay only checks AppID for privileged API calls, allowing attackers to access any privileged API post-exploitation. Our analysis identified numerous undocumented but accessible APIs, including the privileged "rpc()" API, which can access Alipay's backend cloud server. This API, intended for Alipay's internal use, can be accessed by any sub-app, posing significant security risks.

4.2 Exploiting Alipay's Vulnerabilities

An attacker can exploit these vulnerabilities by crafting a phishing deep link, such as alipays://platformapi/startapp?appId=[1688]&url=malicious.com. When a user clicks this link, the 1688 sub-app executes malicious JavaScript from malicious.com, invoking the "rpc()" API to access Alipay's cloud servers and manipulate user data.

4.3 Identity Confusion Vulnerabilities in TikTok

TikTok, another widely used super-app, also exhibits identity confusion vulnerabilities. The app ID confusion stems from URL matching using endswith, while domain name confusion arises from a race condition in the customized WebView's onPageStarted handler.

4.3.1 Exploiting TikTok's Vulnerabilities

We reported these vulnerabilities to TikTok, which then deployed a patch updating its Chromium kernel. However, the patch remains vulnerable as attackers can use an error URL to delay webpage rendering and exploit the race condition. Specifically, attackers create a malicious webpage that abuses benign.com's identity by executing JavaScript with an unsupported scheme, triggering the race condition.

LESSONS LEARNED AND MITIGATION

5.1 Atomic Definition of Identity

The primary lesson from our research is that identity checks should follow the least privilege principle. An atomic definition of identity, combining domain name, sub-app ID, and capability, can mitigate identity confusion. This approach ensures clear coordination between developers of super-apps, sub-apps, and WebView.

5.2 Domain Synchronization

Mitigating identity confusions also benefits from domain synchronization between mobile and web layers of WebView. Tools like Draco provide a good example by modifying native WebView code to support domain synchronization, ensuring up-to-date domain information.

5.3 Developer Best Practices

Sub-app developers must prioritize security, particularly for sensitive interfaces like the launching webpage. Thorough understanding and implementation of super-app security checks, such as URL whitelisting, are essential.

ETHICS

We addressed ethical considerations by informing all 47 super-app developers of their vulnerabilities. Of these, 29 confirmed their vulnerabilities, and 19 have implemented fixes. We conducted all attacks on our own devices with test accounts to avoid harming real users or servers.

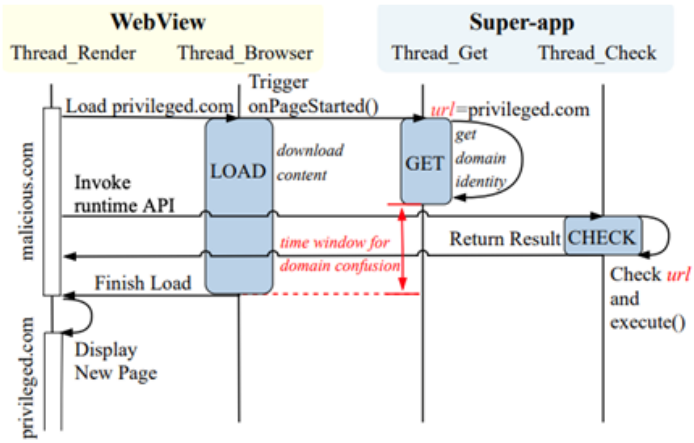
RELATED WORK

7.1 App-in-App Ecosystems

Recent studies have explored the design, prevalence, usage, and flaws of app-in-app ecosystems. However, our research uniquely focuses on identity confusion vulnerabilities, a threat model not previously studied.

7.2 WebView Security

Prior research highlights the security risks of WebView, particularly code injection and malicious ad hijacking. Our work extends these findings by examining identity confusion in WebView-based app-in-app ecosystems.



7.3 Identity Checks

Studies have investigated identity check flaws in mobile and web apps, emphasizing the importance of robust authorization mechanisms. Our research adds to this by addressing the unique challenges of identity verification in app-in-app ecosystems.

CONCLUSION

This paper provides a systematic study of identity confusions in app-in-app ecosystems, categorizing them into domain name, app ID, and capability confusions. Our findings reveal that all 47 super-apps studied are vulnerable to these issues, underscoring the need for improved identity verification processes. We also offer mitigation strategies to enhance security in these complex ecosystems.

Acknowledgements
We thank the anonymous reviewers for their insightful comments that improved the quality of this paper. This work was supported by the National Science Foundation (NSF) under grants CNS-20-46361 and CNS-18-54001, the National Natural Science Foundation of China, the Natural Science Foundation of Shanghai, and the China Postdoctoral Science Foundation. Yuan Zhang was supported by the Shanghai Rising-Star Program. The views and conclusions herein are those of the authors and do not necessarily represent the official policies or endorsements of NSF. Min Yang is the corresponding author and a faculty member at the Shanghai Institute of Intelligent Electronics & Systems, Shanghai Institute for Advanced Communication and Data Science, and the Engineering Research Center of Cyber Security Auditing and Monitoring, Ministry of Education, China.

REFERENCES

[1]. Yinzhi Cao, Zhichun Li, Vaibhav Rastogi, Yan Chen, and Xitao Wen. Virtual browser: a virtualized browser to sandbox third-party javascripts with enhanced security. In Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security, pages 8–9, 2012.

[2]. Yinzhi Cao, Vaibhav Rastogi, Zhichun Li, Yan Chen, and Alexander Moshchuk. Redefining web browser principals with a configurable origin policy. In 2013 43rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), pages 1–12. IEEE, 2013.

[3]. Yinzhi Cao, Yan Shoshitaishvili, Kevin Borgolte, Christopher Kruegel, Giovanni Vigna, and Yan Chen. Protecting web-based single sign-on protocols against relying party impersonation attacks through a dedicated bi-directional authenticated secure channel. In International Workshop on Recent Advances in Intrusion Detection, pages 276–298. Springer, 2014.

[4]. Yinzhi Cao, Vinod Yegneswaran, Phillip A Porras, and Yan Chen. Pathcutter: Severing the self-propagation path of xss javascript worms in social web networks. In NDSS, 2012.

[5]. Jianjun Chen, Jian Jiang, Haixin Duan, Tao Wan, Shuo Chen, Vern Paxson, and Min Yang. We still don't have secure cross-domain requests: an empirical study of CORS. In 27th USENIX Security Symposium (USENIX Security 18), pages 1079–1093, 2018.

[6]. Jianjun Chen, Vern Paxson, and Jian Jiang. Composition kills: A case study of email sender authentication. In 29th USENIX Security Symposium (USENIX Security 20), pages 2183–2199, 2020.

[7]. Erika Chin and David Wagner. Bifocals: Analyzing webview vulnerabilities in android applications. In International Workshop on Information Security Applications, pages 138–159. Springer, 2013.

[8]. Drew Davidson, Yaohui Chen, Franklin George, Long Lu, and Somesh Jha. Secure integration of web content and applications on commodity mobile operating systems. In Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, pages 652–665, 2017.

[9]. Martin Georgiev, Suman Jana, and Vitaly Shmatikov. Breaking and fixing origin-based access control in hybrid web/mobile application frameworks. In Proc. of the Network and Distributed System Security Symposium (NDSS'14), 2014.

[10]. Grant Hernandez, Dave Jing Tian, Anurag Swarnim Yadav, Byron J Williams, and Kevin RB Butler. Bigmac: Fine-grained policy analysis of android firmware. In 29th USENIX Security Symposium (USENIX Security 20), pages 271–287, 2020.

[11]. Xing Jin, Xuchao Hu, Kailiang Ying, Wenliang Du, Heng Yin, and Gautam Nagesh Peri. Code injection attacks on html5-based mobile apps: Characterization, detection and mitigation. In Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, pages 66–77, 2014.

[12]. Phi Tuong Lau. Static detection of event-driven races in html5-based mobile apps. In International Conference on Verification and Evaluation of Computer and Communication Systems, pages 32–46. Springer.

[13]. Jiyeon Lee, Hayeon Kim, Junghwan Park, Insik Shin, and Sooel Son. Pride and prejudice in progressive web apps: Abusing native app-like features in web applications. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, pages 1731–1746, 2018.

[14]. Song Li, Mingqing Kang, Jianwei Hou, and Yinzhi Cao. Detecting Node.Js Prototype Pollution Vulnerabilities via Object Lookup Analysis, page 268–279. Association for Computing Machinery, New York, NY, USA, 2021.

[15]. Tongxin Li, Xueqiang Wang, Mingming Zha, Kai Chen, XiaoFeng Wang, Luyi Xing, Xiaolong Bai, Nan Zhang, and Xinhui Han. Unleashing the walking dead: Understanding cross-app remote infections on mobile webviews. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, pages 829–844, 2017.

[16]. Haoran Lu, Luyi Xing, Yue Xiao, Yifan Zhang, Xiaojing Liao, XiaoFeng Wang, and Xueqiang Wang. Demystifying resource management risks in emerging mobile app-in-app ecosystems. In Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security, pages 569–585, 2020.

[17]. Tongbo Luo, Hao Hao, Wenliang Du, Yifei Wang, and Heng Yin. Attacks on webview in the android system. In Proceedings of the 27th Annual Computer Security Applications Conference, pages 343–352, 2011.

[18]. Tongbo Luo, Xing Jin, Ajai Ananthanarayanan, and Wenliang Du. Touchjacking attacks on web in android, ios, and windows phone. In International Symposium on Foundations and Practice of Security, pages 227–243. Springer, 2012.

[19]. Patrick Mutchler, Adam Doupé, John Mitchell, Chris Kruegel, and Giovanni Vigna. A large-scale study of mobile web app security. In Proceedings of the Mobile Security Technologies Workshop (MoST), page 50, 2015.

[20]. Phu H Phung, Abhinav Mohanty, Rahul Rachapalli, and Meera Sridhar. Hybridguard: A principal-based permission and fine-grained policy enforcement framework for web-based mobile applications. In 2017 IEEE Security and Privacy Workshops (SPW), pages 147–156. IEEE, 2017.

[21]. Vaibhav Rastogi, Rui Shao, Yan Chen, Xiang Pan, Shihong Zou, and Ryan Riley. Detecting hidden attacks through the mobile app-web interfaces. In Proc. of the Network and Distributed System Security Symposium (NDSS'16), 2016.

[22]. Claudio Rizzo, Lorenzo Cavallaro, and Johannes Kinder. Babelview: Evaluating the impact of code injection attacks in mobile webviews. In International Symposium on Research in Attacks, Intrusions, and Defenses, pages 25–46. Springer, 2018.

[23]. Kapil Singh. Practical context-aware permission control for hybrid mobile applications. In International Workshop on Recent Advances in Intrusion Detection, pages 307–327. Springer, 2013.

[24]. Stephen Smalley and Robert Craig. Security enhanced (se) android: Bringing flexible mac to android. In Proc. of the Network and Distributed System Security Symposium (NDSS'13), 2013.

[25]. Sooel Son, Daehyeok Kim, and Vitaly Shmatikov. What mobile ads know about mobile users. In Proc. of the Network and Distributed System Security Symposium (NDSS'16), 2016.

[26]. Wei Song, Qingqing Huang, and Jeff Huang. Understanding javascript vulnerabilities in large real-world android applications. IEEE Transactions on Dependable and Secure Computing, 17(5):1063–1078, 2018.

[27]. Yutian Tang, Yulei Sui, Haoyu Wang, Xiapu Luo, Hao Zhou, and Zhou Xu. All your app links are belong to us: understanding the threats of instant apps based attacks. In Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, pages 914–926, 2020.

[28]. Guliz Seray Tuncay, Soteris Demetriou, and Carl A Gunter. Draco: A system for uniform and fine-grained access control for web code on android. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, pages 104–115, 2016.

[29]. Rui Wang, Luyi Xing, XiaoFeng Wang, and Shuo Chen. Unauthorized origin crossing on mobile platforms: Threats and mitigation. In Proceedings of the 2013 ACM SIGSAC conference on Computer & Communications Security, pages 635–646, 2013.

[30]. Guangliang Yang and Jeff Huang. Automated generation of event-oriented exploits in android hybrid apps. In Proc. of the Network and Distributed System Security Symposium (NDSS'18), 2018.

[31]. GuangLiang Yang, Jeff Huang, and Guofei Gu. Iframes/popups are dangerous in mobile webview: studying and mitigating differential context vulnerabilities. In 28th USENIX Security Symposium (USENIX Security 19), pages 977–994, 2019.