International Journal of Emerging Research in Management &Technology
ISSN: 2278-9359 (Volume-6, Issue-7)

Research Article | **July 2017**

Android OS Fragmentation: Causes and Concerns

Manvindar Singh Chauhan*, Kulvinder Singh
Department of CSE, DIET Rishikesh, Dehradun,
Uttarakhand, India

Abstract—

Android is facing a serious security concern due to OS fragmentation, as a significant number of android users are using much older version of the mobile OS without being updated with latest security patches. The rush to release new devices without sufficient testing is inadvertently introducing security flaws.

Keywords— Android Security, Smartphone Security, Android Fragmentation, Android Customization, OHA

I. INTRODUCTION

Android is an open source OS developed by Google led Open Handset Alliance(OHA). This openness leads manufacturers to prefer android as Os for their new devices. As on today about 60% of android devices are running with Android 5.0 or lower version of the OS[1]. Having a large share of Smartphone market, Google with Android has brought a serious security concern along with it. One of the main reasons of this security threat is fragmented development of android. Android phone manufacturers are under the perpetual pressure to quickly on their new models, continuously customizing android to fit their hardware[2].

II. CAUSES OF OS FRAGMENTATION

The openness of android, allows vendors to alter it at their will like add their own widgets and thematic "skins" to and an OS, making arbitrary customizations to fit the OS to their hardware. The fact that newer versions of the android OS aren't available to certain devices keeps the manufacturers of those devices reliant on older versions of the OS. Which is the major cause of release of android devices with past versions even after release of latest version. Fast pace of release cycle of Android Open Source Project is also making the situation more complicated. 26* official versions of Android have be released till date, and every year at least one more major version is released. All of these versions are customized by the vendors and carriers which results thousands of customized branches of the Android in the world. This vast variety of Customized Android OS makes development and testing of new apps challenging across different phones. This also extends the security risks when vendors and carriers customize the functionalities without fully understanding the security implications of the changes they make.

III. CUSTOMIZATION EFFECTS

The fragmentation caused by the Android customization not only makes the development and testing of the new application difficult across different device models and hardware configurations a challenge, but also inevitably opens doors for security risks.

A. Compatibility and Portability Issues

Because of the differences between Android releases, hardware specifications and device variations the effort required to build applications that work seamlessly on all devices can be exhausting, which might intuitively introduce compatibility issues.

To ensure the consistent behaviour of Android application irrespective of the Android OS and hardware configuration, Google introduced Firebase Test Lab for Android, for testing of apps before release.

Keeping the benefits of Android community including users, developers and manufacturers, Google launched Compatibility program to address the fragmentation issue and ensure the compatibility across Android OS.

B. Android Update Issue

Google regularly advises security updates to be pushed on devices to fix known and detected security vulnerabilities and bugs. However, device manufacturers and carriers might be busy testing and optimizing the security updates for their own customized versions of Android. This produces huge delays in following Google's patch schedule and in pushing the advised updates to existing device models. Even worse, device vendors might opt to stop pushing further security updates to older models due to a lack of monetary incentives, a fact that can make several unpatched custom android devices under major security threats. For example, Samsung's Note 2.0 have not received any security updates since 4.4.2.

Study[4] had revealed that there is a significant variability in delivering the security updates across different device manufacturers and network operators. The study further reveals that 87.7% of the collected Android devices have major

security vulnerabilities and are exposed to at least 1 major threat, including udev exploid [5], Gingerbreak [6], Apk duplicate file names [7], Apk unchecked names [8], etc.

IV. TYPES OF SECURITY RISKS CAUSED BY CUSTOMIZATIONS

Customization process leads to the modification, addition or removal of feature, functionality or privileges originally provided in Android OS. Based on these types of changes in OS, security risks can be customized in three types: risks due to modifications, risks due to additions and risk due to removals.

A. Security Risks due to Addition

Addition of new app, component or framework or services, libraries and drivers may lead to such type of security risks if not carefully configured.

1) *Vulnerability Due to Weak Driver Configuration:* For adding device drivers, vendors have to specify the file system permissions some of which are security critical. Such devices could allow unauthorized app to access sensitive user data for example GPS location etc or system capabilities without requiring permissions from the user for example, redirecting driver to take picture or capture screenshot etc.
2) *Vulnerable System Apps:* If not carefully designed and implemented preloaded apps or components may contain security vulnerabilities. Indeed recent studies show that many pre-loaded apps on custom stock images are vulnerable, leaking system capabilities or sensitive user information to unauthorized parties[3]. It is found that vendor preloaded apps are over-privileged.
3) *Vulnerable Libraries and Framework Services:* System services are responsible for providing access to core Android functionalities, by enforcing several criteria including, proper access control based on caller's identity and permissions. Any type of compromise with the access control check, either weakening or removal at all might lead to put the corresponding operations of at the risk or might expose to unauthorized apps.

B. Security Risks Due to Modifications

Risks introduced due to modification of existing preloaded app's configuration and implementation, or modification in framework system services and libraries and modification in system configuration and device drivers fall in this category.

1) *Vulnerable System Apps:* Vendors might weaken the access control on existing components during customization of preloaded apps, by removing the security checks programmatically or by downgrading their protection within manifest. If the underlying component provides privileged capabilities, downgrading its protection would naturally lead to known Android vulnerabilities such as permission re-delegation attacks, and content leaks and pollution attacks.
2) *Weaker System wide Configuration:* During Android customization device manufacturers might decide to alter certain security-critical configuration used to protect privileged resources and capabilities on the system. If not carefully carried out, the new security configurations might be weaker than the original ones, thus breaking some of the assumptions made by other components.
3) *Breaking Relationships between Android Components:* Different Android components are connected together by Inter-Component Communication (ICC). Intents are the primary medium for ICC, and describe operations to be performed by the recipient. In the context of inter-app communication scenarios, individual apps often suffer from risky vulnerabilities such as Intent hijacking and spoofing, resulting in leaking sensitive user data. If ICC reference points(i.e. invocation of the content provider's authority) are not well guarded a malicious app might claim the ownership of the referenced entity and thus acquire some privileges associated with it.

C. Security Risks due to Removal

Removing a certain app or component that was originally provided by the Android during customization, can lead to breaking the intrinsic relationship that exists between them. When an attribute (e.g. a package name, authority, action, etc.) is used on a device but the party defining it has been removed, a malicious app can fill the gap to acquire critical system capabilities, by simply disguising as the owner of the attribute.

V. CONCLUSIONS

Due to OS fragmentation and vendor failure in providing the latest updated OS to all its customer base is holding Google back from advancing its mobile strategy. The most dangerous aspects of fragmentation is the challenges it presents for the application developers. In order to provide support for older devices developers need to make extra efforts. Google's Android Compatibility Program is an outcome to cope with the troubles of developers caused by OS fragmentation. Fragmentation also affects enterprise adoption of Android due the problem it causes for IT. The biggest challenge for an organization is to provide enough support to account for every device and OS version.

REFERENCES

[1] Android Developer website. [Online]. Available: https://developer.android.com/about/dashboards/index.html
[2] Indiana University website. [Online]. Available: http://www.cs.indiana.edu/~zhou/files/sp14_zhou.pdf
[3] Lei Wu, Michael Grace, Yajin Zhou, Chiachih Wu, and Xuxian Jiang. The impact of vendor customizations on android security. In Proceedings of the 2013 ACM SIGSAC conference on Computer communications security, CCS '13, pages 623–634, New York, NY, USA, 2013. ACM.

[4] D. R. Thomas, A. R. Beresford, and A. Rice, "Security metrics for the android ecosystem," in Proceedings of the 5th Annual ACM CCS Workshop on Security and Privacy in Smartphones and Mobile Devices, SPSM '15, (New York, NY, USA), pp. 87–98, ACM, 2015.
[5] "exploid udev." http://androidvulnerabilities.org/vulnerabilities/exploid_udev
[6] "Gingerbreak." http://androidvulnerabilities.org/vulnerabilities/Gingerbreak
[7] "Apk duplicate file." http://androidvulnerabilities.org/vulnerabilities/APK_duplicate_file.
[8] "Apk unchecked name." http://androidvulnerabilities.org/vulnerabilities/APK_unchecked_name.
[9] Yousra Aafer, "Systematic Discovery of Android Customization Hazards", August 2016

