A Study on CVE-2020-0022 Remote Code Execution Vulnerability in Android

First A. Author, *Dinesh Munasinghe*

***Abstract*—The population of Android devices is growing faster than ever to fulfill the current consumer requirements. 1.6 Billion users among 3.5 Billion total smartphone users are using Android devices as their primary device [1]. In modern days mobile applications, including android applications are playing a vital role in automating the traditional activities of day-to-day life by upgrading the existing solutions. The modern lifestyle is more and more dependent on mobile applications. Developers are mainly focusing on improving the user experience but not the security of the application. Cyber-attacks have become a major threat in this era. lack of knowledge about security and poor coding practices are leading to vulnerabilities of most applications. Remote Code Execution (RCE) attacks are one of the most prominent security threats for software systems, especially Java-based systems like Android [2]. This paper aims to describe what is Remote code execution and how it affects Android devices. Further, the existing CVE-2020-0022 vulnerability of the Android system is explained in detail. Finally, current patches for RCE are also described in this document.**

Index Terms— Remote Code Execution (RCE), Cyber Security, Software vulnerabilities, Arbitrary Code Execution (ACE)

I. INTRODUCTION

With the growing complex requirement of mobile Apps, the development process of Apps is more and more rapid. In this situation, developers are more focused on quick implementation of the logical functions of Apps and make them more user-friendly. Nevertheless, they spend little time on the security issues of Apps. Programming security bugs, called vulnerabilities, keep on being an important and most costly issue affected to all parts of our cyber world. Remote Code Execution (RCE) has been recognized as one of the most harmful vulnerabilities in Android applications [3].

RCE vulnerabilities allow malicious actors to execute any code of their choice on a remote machine over a network. RCE is a special kind of cross-site scripting attack which belongs to the broader class of arbitrary code execution (ACE) vulnerabilities [4]. According to global stats, 72.2% of mobile device users are using a mobile device with the Android operating system. The security issues of Apps are complex and challenging. Hence source code of this OS released as open-source formate will also lead to some security threats than OS like iOS.

1.1 Mobile OS Market Share

Android is on track to have fewer security vulnerabilities in 2021 than it did last year. In last year 696 vulnerabilities were published in Google Android OS [5].

Year	Vulnerabilities	Average Score
2021	175	6.73
2020	696	6.99
2019	491	7.11
2018	294	7.58

1.2 Vulnerabilities By Year

II. REMOTE CODE EXECUTION (RCE) EXPLAINED

With the internet becoming ubiquitous, though, RCE vulnerabilities' impact grows rapidly. In this scenario, an attacker tries to exploit an existing vulnerability and try to execute threats remotely. RCE can be affected in two different ways,

1. Dynamic Code Execution
2. Memory Safety

1. Dynamic Code Execution

This is the most common RCE attack vector. Most programming languages generate output at the runtime based

on user inputs. This on-the-spot code execution is very powerful when solving complex problems. But an attacker can valid malicious input code and attack the application on its dynamic code generation phase. Broadly speaking, dynamic code execution causes two major classes of RCE vulnerabilities, direct and indirect.In the case of direct dynamic code execution, the malicious actor is aware that their input would be used in code generation. In an indirect case, dynamic code generation might be a side effect and not the primary usage of the input [6].

2. Memory Safety

Memory safety means preventing code from accessing parts of memory that it did not initialize or get as an input. This lack of memory safety results in unauthorized data access. operating system and the underlying hardware use memory to store actual executable code. Metadata about code execution is also stored in memory. Getting access to this memory could result in ACE and possibly RCE. There can be several reasons for memory safety vulnerability. Such as,

- Software design flaws

Software design flaws are a type of memory safety vulnerability where there's a design error in some underlying component.

- Buffer overflow or Buffer overflow

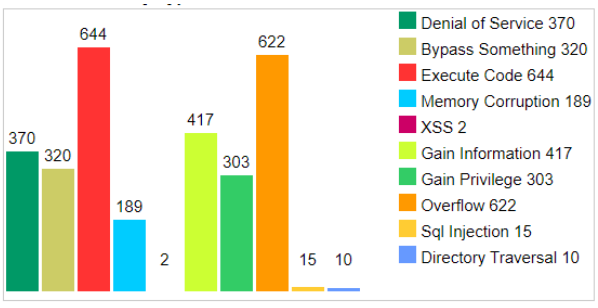
Buffer overflow (also known as buffer overread) is a fairly simple and well-known technique to violate memory safety. It exploits a design flaw or a bug to write to the memory cells that follow the actual end of a memory buffer

- Hardware design flaws.

Memory safety violations can occur because of hardware security design flaws as well. This is less common and harder to find, such vulnerabilities usually have an extremely high impact.

III. RCE IN ANDROID APPLICATIONS

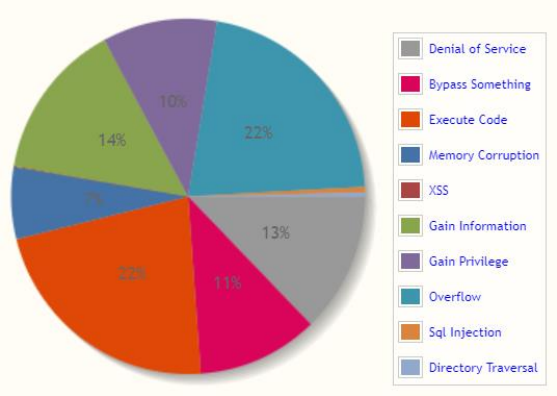
Multiple vulnerabilities have been discovered in the Google Android operating system (OS), the most severe of which could allow for remote code execution. Google has struggled with the spread of malware via Android apps being downloaded from the Google Play store and has made a huge effort in the last years. According to Google, the vulnerabilities pose a high risk for consumers as well as business and government institution users. However, the most critical of these found in the System component of Android could allow for remote code execution (RCE), depending on the existing privileges on the device, according to Google [7]. According to the below 3.1 figure, code execution vulnerabilities are the highest vulnerability type for the Android OS in last year [8].



3.1 Android Vulnerabilities By Type

Multiple vulnerabilities have been discovered in Google Android OS, the most severe of which could allow for remote code execution within the context of a privileged process. Details of these vulnerabilities are as follows:

- Multiple vulnerabilities in Framework that could allow for Escalation of Privileges (CVE-2021-0400, CVE-2021-0426, CVE-2021-0427, CVE-2021-0432, CVE-2021-0438, CVE-2021-0439, CVE-2021-0442)
- Multiple vulnerabilities in Framework that could allow for Information Disclosure (CVE-2021-0443, CVE-2021-0444)
- Multiple vulnerabilities in Media Framework that could allow for Escalation of Privilege (CVE-2021-0437)
- Multiple vulnerabilities in Media Framework that could allow for Information Disclosure (CVE-2021-0436, CVE-2021-0471)
- Multiple vulnerabilities in System that could allow for Remote Code Execution (CVE-2021-0430)
- Multiple vulnerabilities in System that could allow for Escalation of Privilege (CVE-2021-0429, CVE-2021-0433, CVE-2021-0445, CVE-2021-0446)
- Multiple vulnerabilities in System that could allow for Information Disclosure (CVE-2021-0428, CVE-2021-0431, CVE-2021-0435)
- Multiple vulnerabilities in Kernel Components that could allow for Escalation of Privilege (CVE-2020-15436)
- Multiple vulnerabilities in Kernel Components that could allow for Information Disclosure (CVE-2020-25705)
- Multiple high severity vulnerabilities in MediaTek components (CVE-2021-0468)
- Multiple critical severity vulnerabilities in Qualcomm components (CVE-2020-11210)
- Multiple high severity vulnerabilities in Qualcomm components (CVE-2020-11191, CVE-2020-11236, CVE-2020-11237, CVE-2020-11242, CVE-2020-11243, CVE-2020-11245, CVE-2020-11246, CVE-2020-11247, CVE-2020-11251, CVE-2020-11252, CVE-2020-11255).



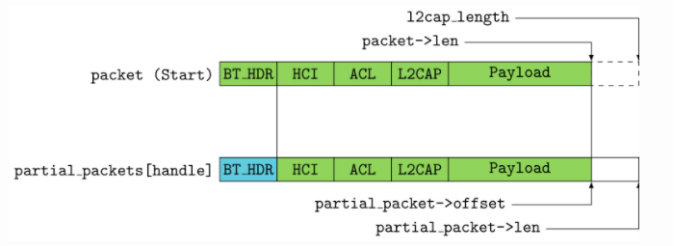
3.2 Percentages of Vulnerabilities By Type

Successful exploitation of the most severe of these vulnerabilities could allow for remote code execution within the context of a privileged process. Above mention, CVE examples of the above 22% of code execution vulnerability statistics. Google this week announced the release of patches for 37 vulnerabilities as part of the Android security updates for March 2021, including a fix for a critical flaw in the System component [9]. Google this week announced the release of patches for 37 vulnerabilities as part of the Android security updates for March 2021, including a fix for a critical flaw in the System component. All of these flaws were rated high severity, with their exploitation leading to remote code execution (three bugs), the elevation of privilege (five issues), and information disclosure (one vulnerability).

- One of the most significant fixes this month regards CVE-2021-0397, a critical vulnerability affecting Android 8.1, 9, 10, and 11 that could allow remote attackers to execute arbitrary code on a device using a specially crafted transmission.
- The company also addresses nine other high-severity flaws in the 2021-03-01 security patch level. An additional six security flaws were found in the System component and two affecting the Framework.
- The most severe vulnerability in the Framework section "could enable a local attacker with privileged access to gain access to sensitive data," Google said.
- The flaw in Android runtime tracked as CVE-2021-0395, only affects devices using Android 11 and could allow a "local attacker to execute arbitrary code within the context of a privileged process."
- The second security patch level of 2021-03-05 includes fixes for Kernel components, Qualcomm components, and Qualcomm closed-source components [7].

IV. CVE-2020-0022 VULNERABILITY

This is a critical vulnerability in Android which allows remote code execution with Bluetooth-enabled devices. This has already been fixed with the Android February 2020 update. This vulnerability allows an attacker to organize remote code execution by sending a specially crafted Bluetooth package on the Bluetooth stack [11].



4.1 L2CAP Packet Structure

The problem was classified as critical since this can be discreetly exploited by an attacker within Bluetooth range and that also this does not require interaction with its victim. It is possible that a vulnerability can create worms that chain neighboring devices. For an attacker, it is enough to know the MAC address of the victim's device. This does not require preliminary pairing, but Bluetooth must be activated on the device. On some devices, Bluetooth MAC address can be calculated based on Wi-Fi MAC address. If the vulnerability is successfully exploited, an attacker can execute threat code with the rights of a background process that coordinates the operation of Bluetooth on Android.

It is only known that the vulnerability is present in the package build code and it is caused by an incorrect calculation of the L2CAP packet size. In Android 8.0 to 9.0, a nearby attacker can silently execute arbitrary code with the privileges of the Bluetooth daemon as long as this communication medium is enabled. Older versions of Android are potentially prone to the problem.

To mitigate the flaw, Ruge recommends disabling Bluetooth and enable it only "if strictly necessary." If you need to activate Bluetooth, it is recommended to set the device non-discoverable for pairing with other devices. Importantly update the system with the 2020 February update will fix the issue.

V. CONCLUSION

Remote code execution vulnerability can be considered a critical vulnerability in Android applications. Patching RCE vulnerabilities are possible with OS and application updates but it doesn't completely ensure that no one can break the application. Vendors should release appropriate updates to their systems, immediately after appropriate testing. Users should only download applications from trusted vendors in the Play Store. Users should not visit un-trusted websites or follow links provided by unknown or un-trusted sources [10]. Inform and educate users regarding threats posed by hypertext links contained in emails or attachments, especially from un-trusted sources will also help to reduce the possibility of being a victim of an attack.

References

[1]S. O'Dea, "Android - Statistics & Facts", Statista, 2021. [Online]. Available: https://www.statista.com/topics/876/android/#dossierSummary . [Accessed: 12- May- 2021].

[2]S. Bier, B. Fajardo, O. Ezeadum, G. Guzman, K. Z. Sultana and V. Anu, "Mitigating Remote Code Execution Vulnerabilities: A Study on Tomcat and Android Security Updates," 2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS), 2021, pp. 1-6, doi: 10.1109/IEMTRONICS52119.2021.9422666.

[3]Y. Zheng and X. Zhang, “Path sensitive static analysis of web
applications for remote code execution vulnerability detection,” *2013
35th International Conference on Software Engineering (ICSE)*, San
Francisco, CA, USA, 2013, pp. 652-661

[4]T. Yovtchev, "Remote code execution (RCE), explained: what it is and how to prevent it", Sqreen Blog, 2021. [Online]. Available: https://blog.sqreen.com/remote-code-execution-rce-explained/#:~:text=Remote%20code%20execution%20(RCE)%20is,code%20execution%20(ACE)%20vulnerabilities. [Accessed: 10- May- 2021].

[5]"Google Android - Security Vulnerabilities in 2021", Stack.watch, 2021. [Online]. Available: https://stack.watch/product/google/android/. [Accessed: 15- May- 2021].

[6]T. Yovtchev, "Remote code execution (RCE), explained: what it is and how to prevent it", Sqreen Blog, 2021. [Online]. Available: https://blog.sqreen.com/remote-code-execution-rce-explained/#:~:text=Remote%20code%20execution%20(RCE)%20is,code%20execution%20(ACE)%20vulnerabilities. [Accessed: 10- May- 2021].

[7]A. Bizga, "Android Security Bulletin: Google Issues Fix for Critical Remote...", HOTforSecurity, 2021. [Online]. Available: https://hotforsecurity.bitdefender.com/blog/android-security-bulletin-google-issues-fix-for-critical-remote-code-execution-flaw-in-android-system-25415.html. [Accessed: 08- May- 2021].

[8]"Google Android : CVE security vulnerabilities, versions and detailed reports", Cvedetails.com, 2021. [Online]. Available: https://www.cvedetails.com/product/19997/Google-Android.html?vendor_id=1224. [Accessed: 13- May- 2021].

[9]I. Arghire, "Google Patches Critical Remote Code Execution Vulnerability in Android | SecurityWeek.Com", Securityweek.com, 2021. [Online]. Available: https://www.securityweek.com/google-patches-critical-remote-code-execution-vulnerability-android. [Accessed: 11- May- 2021].

[10]J. Qin, H. Zhang, J. Guo, S. Wang, Q. Wen and Y. Shi, "Vulnerability Detection on Android Apps–Inspired by Case Study on Vulnerability Related With Web Functions," in IEEE Access, vol. 8, pp. 106437-106451, 2020, doi: 10.1109/ACCESS.2020.2998043.

[11]J. Ruge, "Critical Bluetooth Vulnerability in Android", Ubunblog, 2021. [Online]. Available: https://ubunlog.com/en/una-vulnerabilidad-en-android-permite-la-ejecucion-remota-de-codigo-con-el-bluetooth-activado/. [Accessed: 11- May- 2021].

First A. Author Dinesh Munasinghe (MS19815756).

He is a Postgraduate student in Sri Lanka Institute of Information Technology (SLIIT). Currently doing MSc in information technology specializing in Cyber Security. Birth in the year 1990 in Anuradhapura, Sri Lanka. He completed his bachelor's degree at the University of Colombo, Sri Lanka in 2007. He had a degree in Bachelor of Information Technology(BIT).

He is currently working as an ICT executive at Emjay International (PVT) Ltd in Rajagiriya, Sri Lanka. He previously worked in Virtual Systems (PVT) Ltd in Anuradhapura, Sri Lanka as a System Administrator for 2 years of time.