**Article Review 2: Vulnerability Analysis of the Android Operating System**

Evan Jenkins

Old Dominion University

Professor Diwakar Yalpi

11/13/2024

**Vulnerability Analysis of the Android Operating System**

**Background**

The most commonly utilized mobile operating system in the world is that of the Android OS. Android OS and its compatible applications often rely on the native C/C++ code libraries. The C/C++ language does have some frequent issues with buffer overflow and other memory errors, which could be exploited by hackers to change data and execute malicious code (Sanna et al., 2024). The research team in the article worked to develop a risk scoring method of Android applications based on the presence of native C/C++ code libraries with known vulnerabilities. This would allow software developers and Android users to check applications that could be potential cybersecurity risks. The research article “proposes an alternative strategy for native code vulnerability identification that does not involve resource-heavy analyzes but leverages on public knowledge of known issues,” which would be quicker and more convenient than previous methods (Sanna et al., 2024).

**Hypothesis**

 The team hypothesized that an algorithm based on public domain knowledge of C/C++ native libraries could reliability identify vulnerable applications in a less computationally heavy way than competing vulnerability assessment models.

**Research Methodology**

First the team used common vulnerabilities exposure (CVE) databases to compile a list of over six thousand known and documented vulnerabilities in the Android OS. They then developed a tool to parse through a given APK, an Android compatible app distribution file, to glean the application’s native libraries, function names, and other information. Once this information is extracted its is then compared to the common vulnerability exposure database through a “whitelist approach” (Sanna et al., 2024). Additionally, they developed an algorithm to determine the risk factor of a given APK based on the whitelist results. The threat algorithm classifies an APK as critical (CVE present and exploitable), high (CVE present, possibly exploitable), medium (CVE possible, possibly exploitable), low (no CVE found), and none (no native library/no known vulnerability).

**Data Analysis**

To assess the effectiveness of their algorithm and tool they ran 100,000 APKs through the tool to detect a group of fifteen popular libraries with CVE. Roughly 40% of this sample set contained potentially troublesome native libraries. Of this subset 55% of the APKs had a risk of high or medium (Sanna et al., 2024).

 They also looked at a dataset of a different CVE recognition method to see if their own tool would achieve comparable results. The competing application, the Librarian, only determined whether an APK had a vulnerability or not and had no risk factor determination like the one of the research team in the article. They determined “the same results as the Librarian did for all applications, except for one, which we detected as MEDIUM risk, but the Librarian says it has no vulnerabilities” (Sanna et al., 2024).

 Overall, they found that their method could detect CVE at a similar rate to applications that required more computational bandwidth, while also being able to provide greater insight into the vulnerability.

**Social Science Principles**

 The article serves as an example of several social science principles in action. Given the widespread use of the Android OS across many different industries and fields then the research done by the team could have impact of many distinct parts of society. They also use parsimony to explain how their method is better than previous methods. They simply explain how their method is less complex without getting bogged down with a tangent on computational bandwidth issues. They use empiricism by setting clear parameters to determine if an APK has a vulnerability risk instead of operating on guesses or opinions. The research team also declared an intention and commitment to maintaining ethical neutrality. They also used skepticism to understand potential reasonings that could prevent their threat detection method from detecting a CVE, such as if the APK developer were to rename a native library. By doing this they opened their project up for further development and improvement in the future.

**Relation to Course Module Concepts**

 The article can also be tied into various module topics from throughout the semester. One is the concept of the psychological role of victims in cybersecurity incidents discussed in Module 5. This concepts discusses how optimism bias and hyperbolic discounting could lead to cybervictimization. For the developers of these APKs I wonder if these psychological roles factor into them ignoring or missing these known and documented vulnerabilities or does the inviting nature of the Android OS allow for amateur errors. Module one introduces social science research methods, including experiments, which involves a treatment group, a control group, and some measured variable. In the article the research team compared there method against a previously established threat detection method to gauge its effectiveness. In their experiment the rival method functioned as the control group, the novel threat detection system was the treatment group, and its ability to find vulnerabilities was measured. This module also details different areas of the cybersecurity field which included the design and developmental focus. This category aims to improve and develop the technology needed for stronger cybersecurity, and the article is an example of this. Module five discusses the varies theories on how cybercriminals think and operate, and one common behavioral theory is that risk versus reward analysis is a core part of cyber offending. From the perspective of a cybercriminal the article highlights how the Android OS could be a low risk and high reward target for a cyberattack. It is known and documented that APKs running on the Android OS often use native C/C++ libraries with known vulnerabilities that are often overlooked. Knowing that it is an easy target that has already been overlooked makes it low risk, while the popularity of the Android OS makes it a high reward target.

**Contribution to Marginalized Groups**

 Due to the open source nature of the Android operating system, it is commonly used in low cost affordable smartphone options. With a smartphone being a barrier to entry for many parts of modern society these more affordable Android devices are appealing to those struggling financially as opposed to the expensive iPhone competition. If the Android operating system is vulnerable to cyberattacks than those relying on the affordable phone option could face even greater financial struggle in the event of a cyberattack. With the research article working to find these vulnerabilities, then they can be patched and provide a safter cybersecurity landscape for those with already limited resources.

 Outside of the typical mobile devices, the Android OS has also appeared in various Internet of Things devices, such as smart appliances, connected lighting systems, thermostats, etc. ((BasuMallick, 2024). While these devices can be convenient for anyone, they can be essential for granting autonomy and self-reliance for those with physical disabilities. So, vulnerabilities in the operating system could potentially impact these devices which are a great aid to those with physical disabilities.

**Overall Contribution**

The Android operating system is the most widely used platform in the mobile device market across the globe and by better understanding the security flaws we can be better understand many issues in society. This would aid in supporting a stronger cyber security landscape, provide insight into mobile device related cybercrime, understanding which devices cybercriminals may be targeting and why, and provide a safer affordable smartphone option for society.

**References**

BasuMallick, C. (2024, March 19). *Android OS: history, features, versions, and Benefits | SpiceWorks*. Spiceworks Inc. https://www.spiceworks.com/tech/tech-general/articles/android-os/#:~:text=These%20devices%20can%20track%20users,specific%20business%20needs%20and%20requirements.

Lucia Sanna, S., Soi, D., Maiorca, D., Fumera, G., & Giacinto, G. (2024). A risk estimation study of native code vulnerabilities in Android Applications. *Journal of Cybersecurity*, *10*(1). https://doi.org/10.1093/cybsec/tyae015