Reading Assignment 1

CMSI 4071

Owen Dewing, Kelly Tao, Juan Ballesteros, Joanna Estrada

12k+ Android apps contain master passwords, secret access keys, secret commands

https://www.zdnet.com/article/12k-android-apps-contain-master-passwords-secret-access-keys-secret-commands/

Researchers from Europe and the US published a study back in April 2020 where they discovered many hidden backdoor behaviors (master passwords, concealed access keys, secret commands, etc) in specific Android apps that could potentially be harmful to the users. They discovered these behaviors through the development of a tool called InputScope, a software that analyzes input form fields found in numerous Android apps. To test this software, the researchers utilized the top 100,000 Play Store apps, the top 20,000 third-party app store apps, and 30,000 other apps. Out of those around 130,000 apps, the researchers identified 12,706 of those apps to have these hidden backdoor access keys, master passwords, commands, etc.

These hidden behaviors could have a variety of effects on the users of these apps. Not all of the behaviors are harmful; for example, some apps have these functions as easter-eggs or mistakes that developers forgot to erase before they released their apps. Furthermore, InputScore incorrectly calculated bad word filters/blacklists as these backdoor behaviors, which are not harmful to the users (these errors accounted for 4,028 apps out of the 12,706). However, the researchers are very concerned that some of these functions/behaviors from these 12,000 + apps could allow attackers or hackers to gain access to users' accounts and their personal data.

Many real-life examples were given of these behaviors. The first example revolves around a remote control app with a master password, that when provided, can grant access to the device, even if it has been remotely locked by its owner after being lost. Another example focuses on an extremely popular screen locker app that when given a specific access key, can reset any user's password, allowing the screen to be unlocked and access to the system. A concerning example is from a live streaming app, that when supplied with an access key, can allow any user to enter the administrator interface. This is dangerous because an attacker could reconfigure the whole app and its functionality. Finally, a popular translation app holds a key that permits users to skip the payment process for certain features, including removing ads.

The researchers supplied visual evidence in the article through a tweet from Brendan Dolan-Gavitt back on March 31, 2020. This 13-second video displays a user going into the NBC Sports app, a very popular and widely used app, and tapping the version number 13 times. After the 13th tap, the user gets prompted with an "Enter Password" prompt. The user then enters the "Konami Code", and they get a secret debug menu. The research team reached out to all of the app developers where they discovered these secret backdoor behaviors; however, many of these developers did not wish to respond.

When app developers create secret access keys, they are opening a door to attackers and security risks. As described in the article, the InputScore tool found that a popular screen locker app uses an access key to reset arbitrary users' passwords to unlock the screen and enter the system. The **failure of systems of systems** relates to how these individual software components interact with the larger ecosystem they operate and how weaknesses in any component can compromise the entire system. Having a master password or access key gives permission to

control not only the app but also the devices' Android OS, device firmware, and network security protocols which leaves user information very vulnerable.

The article describes some hidden backdoor-like mechanisms, including accessibility to unlock personal devices, user's profile data, or even their payment data. The ethical dilemma revolves around transparency and the safety of the user. Leaving backdoors and master passwords is not just a technical oversight but a violation of user trust. Software engineers should be designing systems that are functional and equally prioritize user safety and privacy. To combat these issues there are a few different strategies that can be implemented to software discipline. For one, having regular code reviews and peer checks can help catch any hard-coded passwords or hidden features that shouldn't be there. Developers should stick to secure coding practices, making sure to get rid of hard-coded keys and use tools to scan for any sneaky vulnerabilities. It's also super important to document everything properly so all parts of the software are clear and accounted for, even the ones that are meant to be hidden. If these backdoor features were better documented they might have been removed before releasing the most updated version of software. On the other hand, if developers want to keep some of these features in their app, they should be transparent with the user and have either a popup or form that the user agrees to before they create an account or connect their personal data to the app. This way, the user knows that these features exist and can choose if they want to keep the app or delete it. Most importantly, teams should build a culture that puts user safety and transparency first, making sure to avoid shortcuts that might put users at risk.

Overall, while not all of these behaviors/functions are harmful to the user, it is certainly eye-opening that many of these extremely popular apps that are used daily contain features that cannot be seen by the user. These concerns highlight the importance of tools like InputScope

while calling for users to be informed about the apps that they decide to trust with their personal data.