Researcher aims to develop system to detect app clones on Android markets

October 31 2014

Peng Liu, a professor at Penn State's College of Information Sciences and Technology (IST), is part of a team that is developing a system to accurately and efficiently detect app clones on Android markets.

Mobile apps have exploded in popularity in recent years, as studies have

reported that smartphone owners are spending more time on their apps versus the mobile web. However, users also face increased risks from attackers that clone the codes from legitimate Android apps and repackage them with malicious code. Peng Liu, a professor at Penn State's College of Information Sciences and Technology (IST), is part of a team that is developing technology that would enable users to accurately and efficiently gauge app clones from their legitimate counterparts.

"This tool enables the app market to quickly analyze a very large number of apps within hours," said Liu, director of the Center for Cyber-Security, Information Privacy, and Trust (LIONS Center).

Liu and his collaborators designed and implemented the app clone detection system and evaluated it on five Android markets. The system is described in the paper, "Achieving Accuracy and Scalability Simultaneously in Detecting Application Clones on Android Markets," which was written by Liu; Kai Chen, a researcher at the State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, China; and Yingjun Zhang, a researcher at the Institute of Software, Chinese Academy of Sciences. Chen recently presented the paper at the International Conference on Software Engineering (ICSE) in Hyderabad, South India.

"Three types of people will benefit from our research," Chen said. "App users, software developers and app market managers will all benefit."

According to Liu, the rapidly increasing use of smartphones has enabled some cyber attackers to benefit by cloning popular smartphone apps while embedding malicious code into the clones. Chen said that millions of app users are currently using app clones. Specifically, some attackers go to app stores like Google Play, where they search for popular apps, clone the code, and assemble them with "purpose-added" functionalities

or modifications after reverse-engineering those apps. After cloning an app, attackers would upload it to the same market (e.g. Google Play) or other markets. Clones are frequently used as carriers for smartphone malware, Liu said, as studies have reported that about 86 percent of malware samples are app clones with malicious payloads. While Google Play inspects all new apps for possible clones, sophisticated attackers can easily evade that screening process.

App clones bring a number of severe problems for users and software developers, Liu and Chen said. The most common types of malware are aimed at aggressive advertisements and stealing private information such as Social Security numbers, passwords, and credit card data. In addition, legitimate developers lose their revenue and users to app clones. According to a recent study, the researchers reported in their paper, 14 percent of the advertising revenue and 10 percent of the user base for a developer are diverted to app clones on average.

"It is a very difficult problem for people to deal with," Liu said. "We are taking one step towards solving this problem."

App clones, according to the researchers, meet two essential criteria. First of all, a large portion of the core functionalities of one app are cloned in another app. Secondly, the apps are developed by different authors/ companies. Since two app clones could appear on different markets, cross-market analysis is necessary when trying to detect clones. Since most developers do not release the source code, the researchers use the number of opcodes (bytecode statements) to measure the amount of code to analyze. Traditional clone detection is conducted inside a large software project (e.g. Apache) to identify similar code fragments, not similar apps.

Although considerable research has been conducted on clone detection, Liu et. Al. wrote in the paper, existing techniques are only capable of

detecting app clones on Android markets. The researchers' approach, Liu and Chen said, is more advanced in the sense that it is both accurate and scalable. Current app clone detection methods use Program Dependence Graph (PDG)-based approaches that capture the control flow and data dependencies between the code statements inside code fragments. While they can effectively detect certain types of clones, PDG comparison is not scalable and cannot handle billions of opcodes in multiple markets.

"We wanted to do this research in both accuracy and scalability," Chen said.

According to the researchers, scalable cross-market clone analysis requires scalable pairwise comparison between all the methods (i.e. code fragments in multiple markets). For any two methods in the Android market, they can form a method pair (mp), which is either cloned or not-cloned. The goal of method-level comparison is to separate cloned pairs from non-cloned pairs. Each method is represented by a graph. Liu and his colleagues developed an encoding approach that encodes a control flow graph (CFG)— a representation, using graph notation, of all paths that might be traversed through a program during its execution. After the CFGs are encoded, expensive graph isomorphism checks can be avoided in comparing two methods.

The researchers use a geometry characteristic called centroid of CFG to encode a CFG. A centroid, or geometric center of a two-dimensional region, is the arithmetic mean ("average") position of all the points in the shape. They discovered that centroids have a remarkable capability to distinguish cloned from not-cloned pairs. If two methods in a pair have the same centroid, the mp is almost certain to be cloned. Alternatively, if two methods in an mp have different centroids, the mp is 99 percent to be not-cloned. After implementing a prototype and systemically evaluating it on five Android markets (including 150,145 apps, 203 million methods and 26 billion opcodes), the researchers discovered that

it takes less than one hour to perform cross-market app clone detection after generating centroids only once; and for a given method, it takes less than 0.1 second to find the method clones from the 203 million methods.

"This surprising and intriguing 'centroid effect' enables us to achieve high accuracy without sacrificing scalability when detecting cloned methods," the researchers wrote.

In addition to the centroid effect, Chen said, the researchers discovered that centroid has another "very special property"—monotonicity (or consistency)—that contributes to high scalability. They found that when a method changes a little, its centroid will not change a lot. That property enables the researchers to simultaneously achieve accuracy and scalability when performing clone detection analysis.

"The observed 'centroid effect' and the inherent 'monotonicity' property enable our approach to achieve both high accuracy and scalability," the researchers wrote. "It takes less than one hour to perform cross-market app clone detection."

Liu and Chen said that they plan to make the app clone detection system available to the public through a website they are developing that will have an interface on which users can upload their apps and get quick feedback on whether the app is authentic or a clone. The researchers hope to have the website up and running by the end of the year, Chen said.

"Through accurate and scalable app clone detection, the Android app ecosystem would become healthier and less tolerant to app clones," Liu said. "App users and developers can certainly benefit from a heathier ecosystem."

Provided by Pennsylvania State University