How malware infects apps

June 22 2017, by Peter Hannay

Credit: AI-generated image (disclaimer)

Malicious software on popular mobile platforms such as iOS and Android is at best a nuisance and at worst a security threat to individuals and businesses.

Known as malware, some perpetrators use it to infect apps and get inside your smartphone. Why do they do it? Money, mostly.

The recent Judy malware, for example, was reportedly found in 41 apps in the Google Play store. It seems to have made money for its creators by repeatedly auto-clicking on advertisements. Other mechanisms for mobile malware monetisation include covert sending of premium rate SMS messages, financial fraud and credential theft.

With millions of apps out there, it's a growing problem.

Pervasiveness of mobile app malware

It's difficult to get a firm idea of the size of the malware problem when it comes to apps.

A study conducted as part of a project called ANDRUBIS, published in 2014, examined one million Android applications. These were gathered directly from the Google Play app marketplace, as well as from unofficial marketplaces and services offering pirated apps.

The study found 1.6% of apps sampled from Google Play to be malicious in nature at the time, but other studies have shown different ratios. There is a lack of comprehensive data available concerning malware on the Apple App Store, although there are some known examples.

How apps get infected

The most obvious way malware makes its way into marketplaces is through developers intentionally releasing malicious apps. However, this avenue of attack requires a developer who is willing to produce an application, market it, gain a following and then activate the hostile routines within the application.

It is far more common for malware to be inserted into already existing applications. There are a number of different mechanisms through which criminals achieve this feat:

- **Application republishing:** Apps are automatically downloaded, infected with malware, then republished to app stores, both official and unofficial. Attackers making use of this strategy may publish under the original app name or one that is slightly different. An example of republishing malware was seen recently with the MilkyDoor malware, which allows attackers to bypass firewalls.
- **Malvertising:** Advertisers provide packages of code to allow developers to incorporate ads into their apps. There have been instances in which attackers have managed to purchase advertisements that perform malicious actions through an otherwise benign app. An example of this was the Svpeng malware, which was installed via Google AdSense ads targeting Google Chrome for Android users in Russia. The users did not have to click the ad – simply opening a page and displaying the ad was enough.
- **Application acquisition:** Some developers may wish to sell their apps outright. There is potential for the new owners to release malicious updates that will be automatically installed. While there are no documented cases of this occurring on mobile platforms, developers of browser extensions have spoken out about this issue. In some cases, it is possible to purchase applications with hundreds of thousands of users for a few hundred dollars.
- **Infected development tools:** In one (documented) case, it was reported that infected app development tools were being distributed to app authors. A version of XCode, the primary tool used by iOS developers, would insert malicious functionality into applications that it built and prepared for distribution. Apple told

> Reuters at the time it was working with the developers to ensure "they're using the proper version of Xcode to rebuild their apps."

How malware evades safeguards

Of course, the maintainers of official app marketplaces like Apple and Google have an interest in keeping malware off their platforms.

There are a number of schemes aimed at addressing this: Apple has its app review process, and Google has recently launched its Play Protect feature. Among other programs, these efforts make use of a mix of automated and manual examination of apps in an effort to determine whether they are safe or not.

Malware authors attempt to defeat these processes by concealing the true functionality of their code. There are many ways in which this is accomplished: an attacker may have the application download the hostile portion of the code at a later date after installation, rely on time delays or instruct apps to wait for an external signal before launching their malicious payload.

In fact, similar approaches were reportedly used by Uber developers to show a different version of their app to Apple's engineers, based on their location.

What are the solutions?

Unfortunately, there isn't a single solution to these issues.

End users can ensure they only install applications from reputable developers, app marketplaces can continue to improve detection mechanisms and operating system developers can continue to improve

security.

Nevertheless, malware authors will not be far behind in improving their strategies and devising new ways to compromise devices.

This article was originally published on The Conversation. Read the original article.

Provided by The Conversation