GoodApp.

Executive Summary

Nowadays, with malware crawling everywhere, users are very concerned about protecting and securing their devices. GoodApp is a web application that creates a community for Mobile Smartphone Application users and developers. This community establishes a web of trust between its members, where developers are assigned a rating based on their activity in this trust network. These ratings give users valuable information about which developers have an established and trusted reputation in the community, and which developers should be avoided.

Overview

App store communities are growing rapidly and thousands of new Applications are being uploaded every day. There are close to one million apps in the Apple Store and Google Play combined [1] [2]. Windows Market is slowly growing as well since its release early March 2012. So how can a user rest assured that an App found in any of these stores is not malicious? Right now, the only way to know would be install the App, run it, and find out if it is harmful. With the amount of private and sensitive data on a user's machine, this chain of events does not seem like an optimum solution. GoodApp is able to a provide a rich array of information regarding different Apps to give the user an idea of how dangerous a distributable may be, all without a user having to put their personal machines in danger.

GoodApp is a web application, where a user may register as a developer and be able to upload their own applications, view the results of virus scans of their applications, as well as have them be reviewed by other users. However, you do not have to be a developer to join GoodApp. Basic accounts exist as well that allow users to leave reviews and simply browse the site. GoodApp is a developer centric community, however, its interface is very easy to navigate and can be used by someone who is not very technology or security oriented. It's not just the users who can affect this web of trust. The Developers themselves have a big part in this: using their expanded knowledge of app building environment, they can provide deeper feedback for the community. This developer feedback is referred as endorsement and when a developer provides one, he or she is giving some of their trust to someone else, with the idea somewhat similar to LinkedIn recommendations. Once the web of trust community for developers is established, users can base their App download decisions not just based on the feedback that they might see on the App stores - but also on the reputation of the developer.

How our technology works

Ruby on Rails.

Ruby (version 1.9.2+,http://ruby-lang.org) is the language of choice to build GoodApp's web application.Ruby is simple to learn, elegant, and powerful in its dynamism. In addition, it lays the tracks (no pun intended) for our web framework of choice:
Ruby on Rails (http://rubyonrails.org). Additionally, Good App makes use of as many appropriate and established open source 'gems' (Ruby's current plug-in standard) as possible, effectively speeding the development process.

Ruby on Rails uses the Model, View, Controller pattern (MVC). Views are what the users sees and are determined by the following technologies: HAML, an engine for generating HTML with embedded Ruby that defines the markup of an HTTP response, CSS for defining the style of objects in the HTML, and JavaScript for additional flare, page dynamism, and AJAX (Asynchronous Javascript and XML) for asynchronous page updates. Views are the final block generated by the Ruby on Rails framework after fully processing an HTTP request.

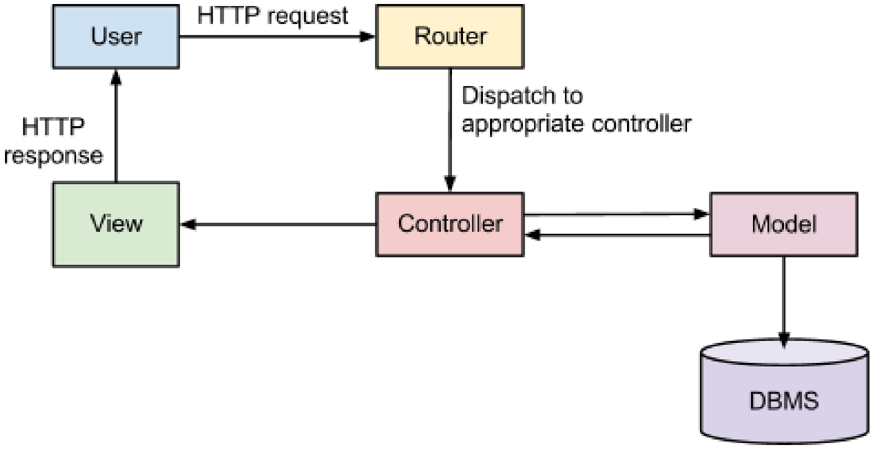


Figure 1: Ruby Framework

VirusTotal.

VirusTotal is a free online service that analyzes files and URLs identifying viruses, worms, trojans and other kinds of malicious content detected by antivirus engines and website scanners. At the same time, it may be used as a means to detect false positives, if it is noticed that similar results are detected as malicious by one or more scanners. GoodApp integrates VirusTotal and uses it to scan App binaries that developers will be uploading onto the site: this will allow users to see if an App has passed or failed multiple virus scans.

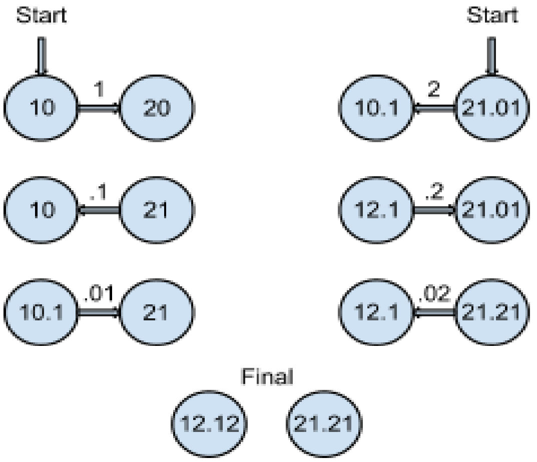
Endorsement Trust Algorithm (trust.cpp).

Initially, developers are given a base trust rating based on their work and the feedback they have received from the community. Once this base trust has been established, trust can be “shared” with other developers. If a developer's trust goes up through being endorsed by others, the trust given to others by this developer should also go up. GoodApp will use an algorithm which produces these desired effects. The algorithm recursively propagates trust given from new endorsements throughout the trust network to a point of termination.

1. *All developers (nodes) are established in a network, with links from endorser to endorsee.*
2. *Each developer gives some fraction of their trust to another developer they trust.*
3. *The trust which has just been given to each developer is again propagated through each trust link from that developer.*
4. *This continues until the fraction of trust being given is smaller than some tolerance, at which point the process starts at (2) again with a new developer.*
5. *Once each developer has been the starting node for propagation, the process ends.*

Each developer has a total allotted amount of trust which they can give: every endorsement given by the developer is a fraction of this total trust. The fraction is inversely proportional to the number of endorsements given by the developer. If some developers start trusting everyone, the trust they give is not as meaningful as trust from someone who is more careful about who they trust. The result of the process is that each developer gives a fraction of their *overall* trust through endorsement. It is also important to note that the order of nodes chosen to propagate has no effect on the end result. A small demonstration of the propagation between two nodes who trust one another is given below.

Ex: Developers with base trust of 10 and 20, fraction trust given = (.1)(base), tolerance = .01



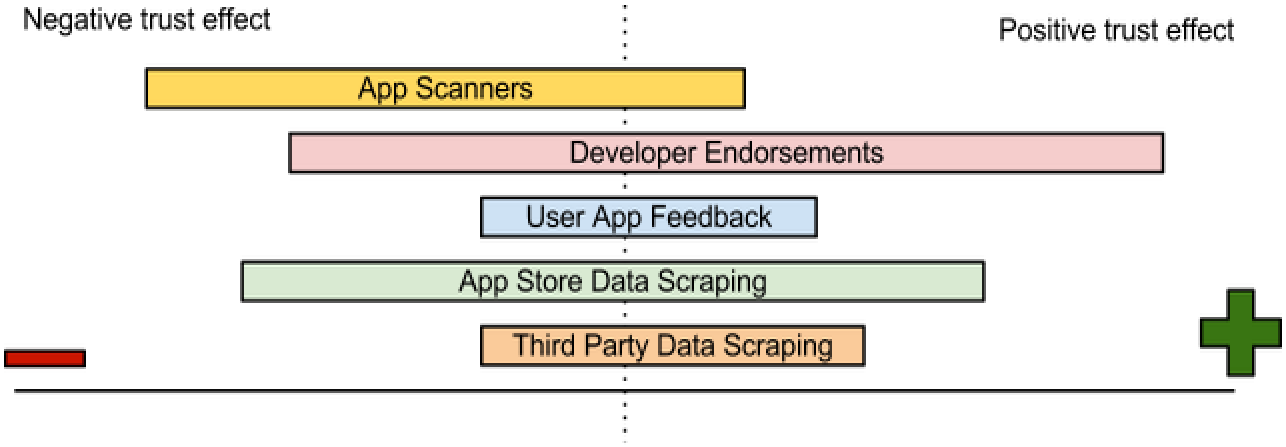
As can be seen from the final result, the node which started with trust 10 has gained 0.1 of the other nodes overall trust (10 + .1(21.21). The other node has gained 0.1 of the first's overall trust (20 + .1(12.12). The nodes do not gain trust less than the tolerance (in this case 0.01). Reversing the starting order does not change the outcome. This process can be applied to larger matrices with similar results.

Rating Formula.

Below are the categories and detailed description of what that make up a user's' trust.

- App Scanners
  - Assuming no false-positives, if an application does not pass App scanners, GoodApp will immediately generate a quite detrimental trust effect on the application's developers trust rating whereas passing the scanners with true colors simply will mean a small trust compensation.
- App Store Data Scraping
  - We expect to use this data in a similar manner to User App Feedback. However, since the data from the App store is more trusted we can give it more weight.
- Third Party Data Scraping
  - Similarly to App Store data scraping, any data scraping we do of 3rd parties will be trusted (not non-validated user feedback) and thus could contribute more significantly than User feedback on the positive side of the effect spectrum.
- User App Feedback
  - User feedback is not quite as crucial as developer endorsement, but it will contribute to the overall developer rating, just at a lesser scale.
- Developer Endorsements
  - This means developer A's rating will be dynamically calculated based on the rating of other developers who've endorsed A and vise versa.

The following diagram describes the range of trust affect each GoodApp component holds. The further a components spans to the left, the more negatively it could potentially affect a developer's rating, while the more it spans to the right, the more it could positively affect a developer's rating.



Trust Effect Spectrum

Depicting the range of trust effect various rating components of the system will have.

Figure 2: Rating Explained

Conclusion

The long term goal for GoodApp is that it will become a standard when it comes to trusting Applications. There has always been efforts to prevent and spread awareness of malicious Apps in many platforms, but there's never been a successful model that professionals and regular users could contribute to with the purpose of creating an important and useful service. Open reputation service is something that is not very familiar to the end user, so at first GoodApp reputation system may be something that is kept private within the stores themselves and would only be used for the stores benefit: for example, as a system to speed up the process of an App's release to the market. However, the aspiration is that in the future, GoodApp will not be isolated, but it will have a deep connection with all app stores. This symbiotic relationship would allow GoodApp to provide more effective web of trust and would facilitate the identification of malicious apps for Applications across all platforms. GoodApp will help users find the information they need to make educated decisions about their downloads, helping them avoid malicious applications and ensuring the safety of their data and their machines.

References:

[1] Apple iPhone 4s, http://www.apple.com/iphone/built-in-apps/app-store.html
[2] Number of available Android applications, http://www.appbrain.com/stats/number-of-android-apps
[3] VirusTotal, https://www.virustotal.com/about/