AnonDroid

xunil154

May 6, 2012

Abstract

AnonDroid is a project designed to make Android users as anonymous to Google as possible. It will include proxying features, bogus data injections, Google Play download spoofing and much more.

Contents

1 Introduction

It appears that there is a large market of people who are begining to worry about how much information Google is able to obtain about their daily lives. Information such as, contacts, searches, favorite videos, emails, GPS locations, favorite foods, and the list continues. With the recent explosion of Android devices on the market it begs the questions of just how much information is Google now able to collect on a persons? New applications such as FourSquare are just downright creepy. This is where the AnonDroid software is going to come into play.

The primary goal of AnonDroid is to help increase the privacy of the end users by either proxying end users data through alternate sessions with Google, generating false information about the end users, or even going as far as to inject fake searches to throw off potential watchers. By doing so AnonDroid will effectively allow it's users to stay as anonymous as possible.

2 Limitations

Obviously it is impossible to create a fully anonymous Android device simply due to the nature of the data. The android devices will still have hardware identifiers which allow them to connect into the cellular networks. And we will not be able to spoof this ID because it will effectively make the Android device useless on any cellular network. The other issue arises with email. Since uses need to login to their email accounts, there is nothing we can do to break that link from that account to that end user. In the case of email, the only piece of information that we could possibly anonymize is the IP address used to login to Google.

3 Structure

AnonDroid will be a complex system consisting of several pieces of software. There will need to be an Android application that runs on the end device. This software will most likely will need to have root privileges. The second piece of software will be ran on a server which will perform the anonymization of the user data fed to it by the android software.

The Android software will tunnel all non-cellular network traffic to the centralized (for now) server. It will then be the responsibility of the server to anonymize as much of the traffic as possible before sending it onto Google.

4 Android software

4.1 Overview

This software will be ran on the end users devices. It will be responsible for thee main tasks.

- Creating a VPN/SSH tunnel back to the central server
- Intercepting all traffic leaving the phone
- Anonymizing GPS coordinates

As well as implementing these tasks, it will need to incorporate these main features as well

- Easily configurable
- Completely transparent to the user

- One click enable/disable functionality
- Generate reports for the user about what personal information is being leaked

4.1.1 VPN

There are several different VPN/SSH proxy application for the Android platform. Rather than reinventing the wheel so to speak, we will use a pre-existing platform for this. The only tricky part will most likely be properly configuring the routing inside the Android system in order to force all outgoing traffic to use the new VPN.

5 Server Software

The server is going to be the most complex portion of AnonDroid since it will need to perform several different tasks without incurring too much of a delay for the end user. The server will need to be:

- Secure
- Scalable
- Configurable
- Modular
- Able to provide many different tasks to provide anonymization.
  - Generate false information
  - Inject bogus queries
  - Filter personal information in a secure manner
  - (future) Virtual android devices

5.1 Security

Because there will be sensitive user data being sent through this server, it is *critical* that this server is heavily locked down and monitored. It is also vital that we develop this application from the ground up with security in mind. It is my suggestion that we use the following principles when designing the software:

- Randomize memory before releasing it back to the system
- Encrypt all data that gets written to disk including logs
- Ensure that *NO* user data ever touches the disk, or if it does encrypt it first

5.2 Scalability

Because it will be possible that the server(s) will be receiving a large quantity of data, it is vital that they will be able to scale incredibly well. There are several different models that can assist in this process. One of which will be to use asynchronous connections and limit the number of threads which are running. There is also a possibility of using a front proxy such as Nginx (nginx.org).

5.3 Distribution

For the proof of concept, everything will be processed on a single server. However, if the final product is on a single server then it will be easy to take down that server should Google decide to.

Instead what we will need is a cluster of computers in which contributing members will be able to volunteer their resources and contribute to the AnonDroid project. This will be the most difficult portion of the AnonDroid project because we will need to protect user data as it enters and exits these volunteered computers. There will also be the issue of releasing information about computers in the cluster to the Android devices as well has. As of right now a P2P database will probably be the best solution and we can adopt principles from the BitCoin system as well as utilize the torrent network.

5.4 Anonymization

For the proof of concept, it is suggested that we simply try to anonymize Google searches that are generated by the Android devices. To do this we can fork the GoogleSharing project (googlesharing.net) and customize it for our purposes and/or build a plugin for Android to use GoogleSharing. Doing the later would provide several benefits such as contributing back to the GoogleSharing project, as well as increase the availability of GoogleSharing to Android users.

We will also be able to make extensive use of the Tor network torproject.org. We can also contribute to this project in the future by requiring that all cluster nodes (see Distribution section) partake in the Tor network as routing nodes.

6 Future

It is expected that this project will continually grow and incorporate new technologies into it. Thus we will need to design the core system in such a way as to make it easy to take data and feed it into a new component which might be written in a different language.

6.1 Virtual devices

One idea that has been presented is the possibility of creating virtual Android devices which end users can then get remote desktop to. Afterward the virtual Android device can then be completely wiped and reused by another user. While this is a fantastic idea it appears that we are going to need to create our own remote desktop application for Android devices. After briefly looking, there are several remote desktop applications designed to connect to desktop machines, but none to connect to Android devices.