Intercepting and Analysing Data Packet from Android Applications to Gain Perspective on Unauthorized Dissemination of Location Information

Muhamad Iqbal bin Basir
Faculty of Electrical Engineering
Universiti Teknologi Mara, Shah Alam, Selangor

***Abstract*— The mobile phone is a modern marvel to humankind since its main role has extended far from just making a simple call. Other than making calls or sending messages, the mobile phone now offers high speed processing power, high RAM capacity and built-in Global Positioning System (GPS) sensor just to name a few. The Android operating system is a major player in the mobile phone industry. Its operating system provide programming interfaces and platforms to fully utilize the mobile phone resource capability. No overhead cost and third party application support from Google Play, has made the operating system (OS) the discernible choice for mobile phones. As the mobile phone becomes more advanced, it is disturbing to know that it is also capable of silently spying on our privacy to the point of disseminating user location information stored in the android device to unauthorized parties. This study aims to identify how users' location information are being transmitted to public domain through analyzing data packet transmitted through our mobile phone. This work also proposes a feasible method for users to closely monitor their location privacy when using an android device. The result of data collection and intensive observation of more than 8000 (0.3% of total application available in Google Play Store) sample applications, it was found that more than 3000 of them had requested android location information.**

Keywords*— Android Applications, Information Privacy, Location-based Applications, GPS

I. INTRODUCTION

From the last century, the apparatus used by human to speak to their counterpart from a distance away are constantly and rapidly evolving. The telephone, at its early stage, required a single pair of wire to establish the communication. Through technology adoption and dedication to providing a better way of communication, inventors have transformed wired telephone into mobile and wireless telephone in 1970.

The desire to communicate efficiently regardless of location is the main motivation that bring us to this achievement. Along with the importance of real time information in this information critical era, all of these help push the communication boundaries that was set by the earlier telephone system. In modern telecommunications, the means of communication using this device are not bound by voice communication boundary. Cell phones nowa days have become more personal and sensitive than just transmitting and receiving voice.

Compared with their predecessor, modern cell phones are equipped with many sensors, high processing capabilities and large memory storage. Moreover, the rich user interface of Android OS and user service experience application, literally, give users access to a device that have computer size capability but with palm size screen. Unfortunately, these modern features could lead to many possibility of privacy and security issues. Unnoticed by most of cell phone user, many information were stored inside our cell phone and some of it were transmitted out without user consent. Although in Android systems, user permissions for certain information access were sought prior to application installation, most users take the issue for granted and at other times they were not left with much choice.

One of the main concerns regarding sensitive information is location information. Almost all modern cell phone have GPS receiver embedded on board. Together with Android OS, the platform could offer the location knowledge service to end user. This piece of information is very sensitive yet are required by some application installed in our cell phone and we used it every day. Location information is very sensitive because it directly represent the cell phone user's last location. This is a privacy threat as when the user moves along with their mobile phone, the user could be tracked using their mobile phone as a beacon.

This paper is to identify if any location information breach by any mean that could compromise user privacy. There are application that genuinely require GPS service and location information for them to function properly. But ones they have knowledge on user location, how they utilize the data would be another story. It is very crucial to analyze each establish communication requested by any running application in the cell phone. By explicitly monitor, capture and analyze each packet transmitted out from cell phone. It is also important that user have very distinctly control privilege upon permission access by application. This approach might cause application not function properly, but this is the best way to retain user privacy.

II. Background and Related Works

A. Location Tracking

There are several ways user location could be determined using a cell phone.

One is through Mobile Subscriber Tracking, where, as the cell phone is powered on and connect with their subscription network, the user location is determined using the commonly called triangulation method. Location is determined by continuously monitoring the signal strength between cell phone and service provider cell tower. By calculating the signal strength from multiple cell towers, the service provider could determine user location to the nearest predetermined tower ID.

Another way is through the Global Positioning System (GPS), where, the GPS receiver provide many service that could benefit user. This is because, it provide a mean of determine its own location. This features can be used in conjunction with map application to show user position on the map and could have better navigation. But there are many application available in Google Play Store that also require location information. There are some possibilities that these application will transmit out our location information through internet which in turn provide a mean to other to track our location. Developers might not have been motivated by the desire to track users when they develop the application, but they might still end up with the ability to do that, due to poor programming practice and effort. By the end of the day they might end up revealing location information about their users to governments or hackers.

B. Past Works on Location Privacy

Location information privacy awareness is nothing new in mobile privacy study. There are a few study related with this issue. Through previous study and report, most of them are acknowledging and addressing the present of the activity of location information request by application.

But none of them present of evidence of the location information were transmitted out to public domain through cell phone connection. Many application that require location information to serve their purpose. But before we can accuse that application are threat to user privacy, a study and evidence must be shown before we put an effort into how we can address and mitigate this issue.

On approach presented by Hazim Almuhimedi and his peers [1] exposed the number of times that user location data were being shared. But the authors did not show how this information were transmitted out. Their study showed qualitative and quantitative evidence that their approaches were complementary and can each play a significant role in empowering users to more effectively control their privacy without presenting any evidence of location information leakage. Their study only focus into user control for their privacy which is the mitigation step. For instance, their mention participants benefited from nudges showing them how often some of their sensitive data was being accessed by applications, with most of participants reassessing their permissions, and half of them further restricting some of their permissions.

From another study did by Bin Liu et.al. [2], their report that most users are uncomfortable with the permissions requested by their mobile apps which indicate user awareness of their location privacy. But the overwhelming number of permissions required caused many users to become incapable of adequately managing their permission settings. Their paper presented a methodology for building personalized privacy assistants to recommend permission pre-settings to users. Through their study, following interactions with the assistant, participants were motivated to further review and modify their settings with daily privacy nudges.

C. Gibler et.al proposed AndroidLeaks [3], a tool for detecting potential privacy leaks on Android system including location information. This tool uses the existing analysis framework of Java programming language on top of the android applications by translating android dex into a Java Archive (JAR) file. However, the translation into jar from dex does not always produce the genuine source. This incorrect translation could lead to an incorrect result.

M. Grace et.al [4] also propose an accurate zero-day android malware analysis framework known as RiskRanger. This tool filter applications from their behaviours. But its results may be inaccurate because it only uses the reachable analysis without taking the taint analysis into consideration.

III. Android Framework

The main problem with the previous study is there is no study that prove the application that use the location information leak out the location information to another party. Most of them skip this crucial step and just straight into the mitigation step. Because location information can be very useful to user, there are many application that genuinely required location information to help user to achieve specific task such as navigation. But there also the possibility that those application abuse the information for other reason.

The Android OS was developed by Google based on the Linux kernel under an open source license. Its user interface is based on direct manipulation such as touchscreen gestures. There have been many versions of the android platform that has evolved since 2008. Among them, only few are still being supported in the market today. The list can be seen in Table 1.

Regardless any android version installed in the cell phone, they are all running on top of the Linux kernel. Hence we can utilize Linux features to monitor and capture all the device process and connection either in real time or through history log.

TABLE I. ANDROID VERSIONS AVAILABLE PRESENTLY

Version	Codename	API	Distribution
2.3.3 - 2.3.7	Gingerbread	10	1.0%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	0.8%
4.1.X - 4.3	Jelly Bean	16, 17, 18	9.1%
4.4	KitKat	19	18.8%
5.0-5.1	Lolipop	21 & 22	32%
6.0	Marshmallow	23	31.2%
7.0	Nougat	24 & 25	7.1%

A. Android Location Services

The Android platform framework run on top of several components. When an application runs, the running code will be compiled and executed at Android Runtime (ART) level. When the application run, they will execute all their main function as well as support library required in order for them to function properly. This is where the location information will be requested by application. Fig.1 shows the Android framework for location services.

Before any application is permitted to access location information, they must request into that particular permission. There are two types of permissions regarding location permission:

- android.permission.ACCESS_COARSE_LOCATION.
- android.permission.ACCESS_FINE_LOCATION.

The difference between this two permission is the latter permission is required if the application need an accurate location information through GPS information. This of course will affect battery life span. While the former permission using Mobile Subscriber Tracking. Both permission must be declared in their application manifest file. From this permission request, we can identify which application that constantly request location update. Either during runtime or silently embedded process not relevant to application main activity.

In the first stage, we should collect some sample and identify how many from that sample that require location information. By identifying the application, we can narrow down our specimen to identify which application that maliciously transmit user location information.

Android application is able to request location information as soon as they start running. There are a service known as Google Play services location APIs. Application developer can request the last known location of the user's device. In most cases, application interested in the user's current location, which is the same as the user's last known location. Specifically, program will call a Fused Location Provider as shown in the software stack of Fig. 2 to retrieve the device's last known location. The fused location provider is one of the location APIs built in Google Play services. Fused Location provider is a feature where location information from various sources were called upon by a single object. It performs the underlying location technology and manages a simple API to help the developer to retrieve location as they only need to specify requirements at a high level, like high accuracy or conserve power. It also optimizes the device's battery power.

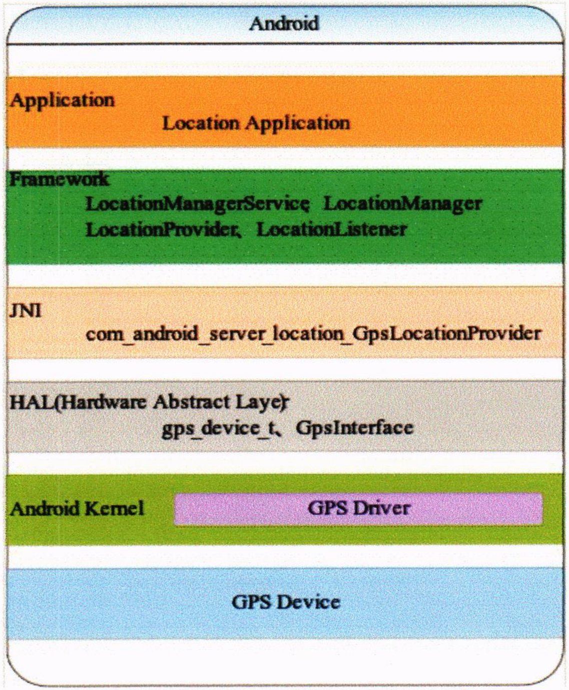

Android
Application
Location Application
Framework
LocationManagerService, LocationManager
LocationProvider, LocationListener
JNI
com_android_server_location_GpsLocationProvider
HAL(Hardware Abstract Layer)
gps_device_t, GpsInterface
Android Kernel
GPS Driver
GPS Device

Fig. 1. Android Framework for Location Services (http://blog.sciencenet.cn/blog-1060307-727660.html)

Application will receive location information in the form of Latitude and Longitude. Besides that, Fused Location Provider also could provide bearing, speed, activity recognition and altitude of device.

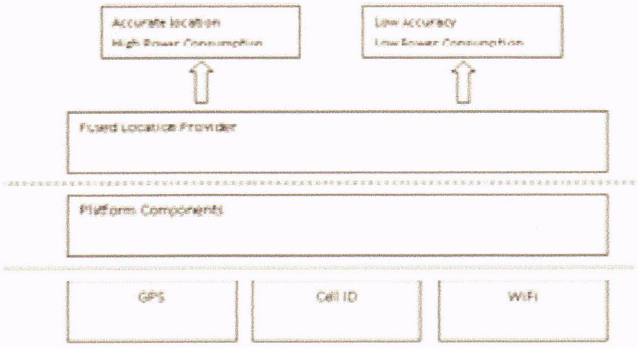


Fig. 2. Android Fused Location Service

B. Packet Analysis

To really conclude that a certain application can really do harm to user privacy, we must prove that after application have knowledge of user location information, that application will transmit out location information to another party and disclose user location with unknown implications. To prove that the android application has disclosed user location information to other parties, we must investigate the packet that was transmitted by the application.

C. Mitigation

After we have analyzed and verified that the location information had been compromised, then mitigation plans should be proposed in order to prevent such information to be transmitted out. The proposed mitigation plan is to use permission control that will block application when they request location information. During installation, users would be asked what permission required by that application. While certain applications really do require location information to function properly, some applications which have no proper intention for location information might also request location permission. Most of the time, unwary users would just ignore the privacy issues and simply install the application without even reviewing the permissions required during installation. Once the application is installed, the permissions have been granted and users have no normal way to disable that permission through android setting.

IV. METHODOLOGY

A. Methodology

Figure 3 show methodology of this study. The idea is to monitor all the application activities through Linux platform installed inside the cell phone. The subject of investigating are to monitor application process activity. This is to identify, which application that will request location information when they running.

Then we will investigate if that application is trying to establish a network connection with other parties. If they were to establish the connection, we would then intercept the connection and capture the packet to investigate, what information if any were transmitted out.

B. Identify Application

There are 2.8 million application available at Google Store. From that huge of number, we need to sample a fraction from that. From that sample, we should identify how many of application required location permission access. This can be achieve by thoroughly review each application permission. Prior to installation, user can view the application required permission.

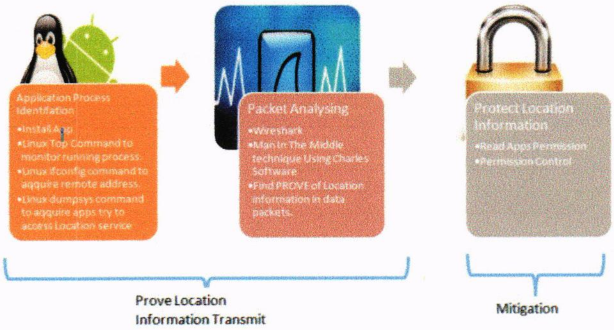

Application Process Identifation
•Install App
•Linux Top Command to monitor running process.
•Linux ifconfig command to aqquire remote address.
•Linux dumpsys command to aqquire apps try to access Location service
Packet Analysing
•Wireshark
•Man In The Middle technique Using Charles Software
•Find PROVE of Location information in data packets.
Protect Location Information
•Read Apps Permission
•Permission Control
Prove Location Information Transmit
Mitigation

Fig. 3. Methodology of Proposed Study

C. Process Monitoring

The dumpsys tool runs on the device and provides information about the status of system services [5]. This tool able to capture application and location related activity log. From the log, we can identify what application try to request location service, when they try to request that and currently active request location information update within certain interval.

D. Packet Analysis

For packet analysis, TCPDump application was installed into the test cell phone. This application will intercept and collect packet and dump it into dump file. These dump file can be view and analyse using packet analyser such as Wireshark. This is true if that application establish the plain connection with any encryption. But there are possibility that information were transmitted using secure Transport Layer Security (TLS)/Secure Sockets Layer (SSL). If this is the case, is will hard to decrypt since we didn't possess the private key of the encryption. To achieve this, the first approach is to use Packet Analyser software. This software will intercept the data stream between mobile phone and remote server that establish connection with the cell phone. After we intercept the network and dump the packet into a dump file, we can investigate the protocol and content of the packet. We are looking for evidence that location information is present inside the transmitted packet.

But first method are not feasible when the application employ secure Transport Layer Security (TLS) and its predecessor, Secure Sockets Layer (SSL) during packet transmission. It is because without private key, it is impossible to decrypt secure packet using mere packet analyzer. The private key is usually stored at remote server and used to decrypt secure packets. To overcome this issue, a second method is employed which is Man-In-The-Middle Technique (MITM) technique. The approach of this technique is to put a server in a middle of connection between mobile phone and the main server. The middle server will trick both ends by telling each of them that they are communicating with legitimate machine. The MITM server then will intercept the communication. MITM server will dynamically generates a certificate for the server and signs it with its own root certificate. Then MITM server will receives the servers certificate, while cell phone will use MITM server's generated certificate.

E. Permission Control

For the applications that were identified to be maliciously transmitting user location information to other parties, a permission control scheme can be employed rather than disabling or uninstalling the application.

V. Results

Based on 8872 application samples that were analyzed from Google Play Store, there were 3809 applications that require location permission. This constitutes a considerable proportion of 43.8% which require location permission. It would be very suspicious when an application that not provide any location awareness services requests location permission. Throughout the 22 android application categories of Fig 4, it was found that almost all categories require location permission.

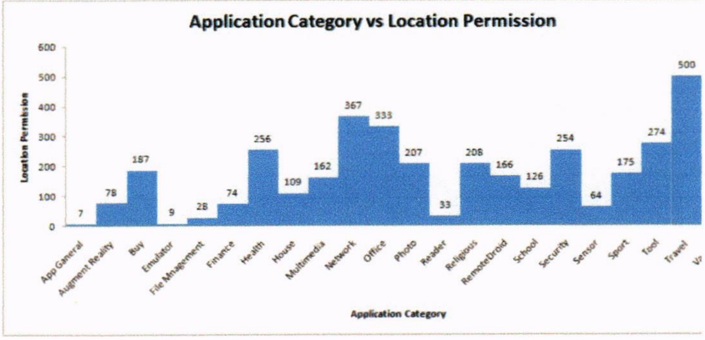
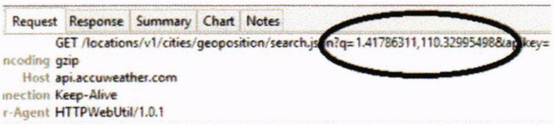


Fig. 4. Application Category vs. Location Permission (source???)

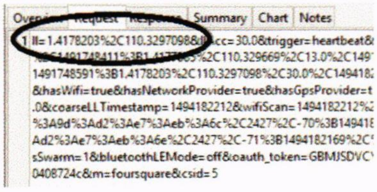
The Install Application process can be monitored using Linux Top command. The important information that can be observed is process ID, user, start time and process status. Every application running have their own ID and Process ID. Using the command *netstat -tp* we were able to view the established connection between device and foreign server. Other important information are service name and their Process ID number. From continuous packet capturing and analysis, it was found that almost all packets were transferred using TLS/SSL port 443. One system application was observed to have transmitted plain Hypertext Transfer Protocol (HTTP) as shown in Figure 5. It uses the GET method to request location information from the server and the particular transmissions from the same application were observed to be consistently periodical whenever location update occurs.

Another application's transmitted packet was observed to have a location term in its query but it was only a portion of unread chunk since it was encrypted as per Figure 6 and Figure 7. The exact location extracted from the transmitted packet was identified as the location of the mobile user as shown by the pin on Fig. 8, which was the location where this test was carried out.



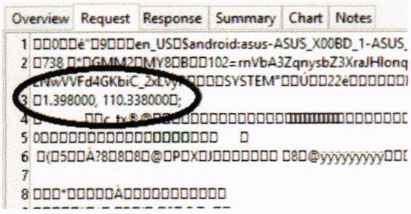
```
        GET /locations/v1/cities/geoposition/search.json?q=1.41786311,110.32995498&apikey=
ncoding gzip
   Host api.accuweather.com
inection Keep-Alive
r-Agent HTTPWebUtil/1.0.1
```

Fig. 5. Location Information in Plain HTTP



```
ll=1.4178203%2C110.3297098&ll...cc=30.0&trigger=heartbeat&
%2C1491748411%3B1.4177...%2C110.329669%2C13.0%2C149
1491748591%3B1.4178203%2C110.3297098%2C30.0%2C149418
&hasWifi=true&hasNetworkProvider=true&hasGpsProvider=t
.0&coarseLLTimestamp=1494182212&wifiScan=1494182212%2
%3A9d%3Ad2%3Ae7%3Aeb%3A6c%2C2427%2C-70%3B149418
Ad2%3Ae7%3Aeb%3A6e%2C2427%2C-71%3B1494182169%2C
sSwarm=1&bluetoothLEMode=off&oauth_token=GBMJSDVC
0408724c&m=foursquare&csid=5
```

Fig. 6. Location Information Chunk in Encrypted Packet



```
□□0□□é"□9□□□en_US□$android:asus-ASUS_X00BD_1-ASUS_
□738 □*□GMM2□MY8□B□□102=mVbA3ZqnysbZ3XraJHlonq
LNwVVFd4GKbiC_2xLvy...□□□□SYSTEM*□□Ú□□22e□□□□□□
□1.398000, 110.338000□;
□        □□c tx®@□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
0□□□□□□□□□□□□□□□□□□□□    □
 □(□5□□À?8□8□8□@□P□X□J□□□□□□□ □8□@ÿÿÿÿÿÿÿÿ□□□

□□□*□□□□□À□□□□□□□□□□□□
```

Fig. 7. Location Information Send By System Application

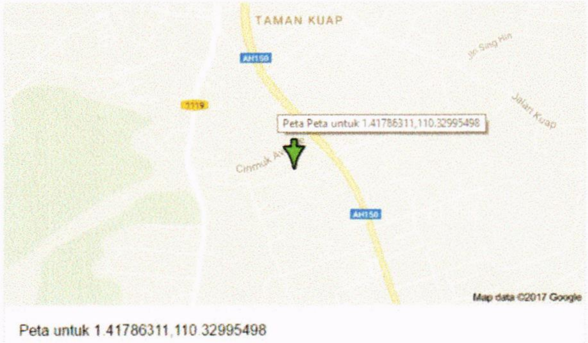


Fig. 8. Exact Location on Map

We employ a Filter Permission application on the applications that we suspected were using location information beyond their intended purpose. But to employ this feature, an Android phone needs to be rooted first because it requires super user ability. There were mix behaviours from the application when location permission was not allowed. Some application would not function properly whereas some would function normally.

VI. CONCLUSIONS

There were about 43.8% of 8872 sampled android applications which were found to request location permission. The tasks of capturing and analyzing each packet were tedious because most of them used TLS/SSL protocol. But from these, there were a few of them that could be decrypted and location information was found embedded in them. From the sampled collection, it was found that some installed application from Google Store were requesting location information for various purposes. There were applications that were requesting location information in order to deliver genuine location-based services to users such as Google Map, social applications, weather forecast applications and shopping suggestion service. When further investigations were conducted, it was found that some of the applications were transmitting out the location information to other parties. This activity could be considered to be beyond their intended purposes and leaked location information can be manipulated in many ways by unknown parties.

Through packet analysis it was found that the information were transmitted in plain or encrypted form. This information were transmitted in web GET or POST method embedded in their query. The location information can be found as the latitude and longitude information. This activity has obviously disclosed the device and user location without consent.

Either way, another issue that was discovered concerns the data that could easily be intercepted and recovered in the middle of the transmission. When this happens through plain HTTP protocol, the location information were easily captured and read by another party. This situation can thus escalate from a privacy issue to a security issue. It is a very dangerous situation as for another specific requirement, this threat could come from a personal tracking tool normally used criminal or warfare purposes.

The Filter Permission application was introduced to the device to control the permission on the application. This method was proposed by a previous study as an alternative to other methods such as Firewall filtering. But permission filtering can be considered a more effective method. When this application was applied, it was observed that the effected application could no longer receive any location information. Some of the applications crashed while others continue to function normally.

ACKNOWLEDGMENT

I would like to thank my supervisor Assoc. Prof. Dr. Habibah Hashim for her guidance. I also would like to extend the gratitude to my wife, family and friend for their continuous support for completing this project.

REFERENCES

[1] Hazim Almuhimedi, Florian Schaub, Norman Sadeh, Idris Adjerid, Alessandro Acquisti, Joshua Gluck, Lorrie Cranor and Yuvraj Agarwal, Your Location has been Shared 5,398 Times!A Field Study on Mobile App Privacy Nudging, School of Computer Science Carnegie Mellon University Pittsburgh, 2014.

[2] Bin Liu, Mads Schaarup Andersen, Florian Schaub, Hazim Almuhimedi, Norman Sadeh, Yuvraj Agarwal and Alessandro Acquisti,To Deny, or Not to Deny: A Personalized Privacy Assistant for Mobile App Permissions Carnegie Mellon University Pittsburgh, PA, USA, 2016

[3] Clint Gibler, Jonathan Crussell1, Jeremy Erickson1 and Hao ChenAndroidLeaks: Automatically Detecting Potential Privacy Leaks In Android Applications on a Large Scale University of California, Davis,2012

[4] M. Grace, Y. Zhou, Q. Zhang, S. Zou, and X. Jiang, Riskranker: scalable and accurate zero-day android malware detection 2012

[5] Dumpsys System Diagnostics https://source.android.com/devices/tech/debug/dumpsys

[6] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

Intercepting and Analysing Data Packet from Android Applications to Gain Perspective on Unauthorized Dissemination of Location Information

ORIGINALITY REPORT

%10	%10	%5	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.synergylabs.org Internet Source	%1
2	www.santman.org Internet Source	%1
3	www.jammer-store.com Internet Source	%1
4	www.idc.com Internet Source	%1
5	developer.android.com Internet Source	%1
6	homepages.cwi.nl Internet Source	%1
7	www.telekoplus.com Internet Source	%1
8	www.googletransparencyproject.org Internet Source	<%1

9	www.interconnective.co.uk Internet Source	<%1
10	www.qub.ac.uk Internet Source	<%1
11	www.citizendia.org Internet Source	<%1
12	ajadjkt.blogspot.co.id Internet Source	<%1
13	www.tabletoid.com Internet Source	<%1
14	www.mecs-press.org Internet Source	<%1
15	source.android.com Internet Source	<%1
16	siela.tu-sofia.bg Internet Source	<%1
17	www.jegandsons.com Internet Source	<%1
18	en.wikipedia.org Internet Source	<%1
19	emmasolutions.net Internet Source	<%1
20	child-gps-tracking-cheap.co.cc Internet Source	