Privacy Leaks in Mobile Phone Internet Access

Collin Mulliner
Security in Telecommunications
Technische Universität Berlin / Deutsche Telekom Labs
Berlin D-10587, Germany
collin@sec.t-labs.tu-berlin.de

***Abstract*—Accessing the Internet and specifically the World Wide Web from a mobile phone is common today. Especially since the usage fees for packet-data access dropped to a point where anybody who can can afford a mobile phone can afford mobile Internet access. Almost every mobile phone today comes with an integrated web browser that can display HTML web pages and execute JavaScript. Almost all major web sites such as news sites, social networks, and shopping sites run websites that are optimized for small displays of mobile phones. Due to the broad use of mobile web access we investigated possible privacy problems of mobile phone web access. We conducted a study where we monitor all HTTP headers sent from mobile phones to our web server. We analyzed the logged data for privacy problems. Through this study we determined that a world wide privacy problem exists when accessing the world wide web from a mobile phone. We show what kind of data is leaked and who leaks it.**

Keywords: Web Security, Mobile Phone Internet, Privacy, Data Collection.

I. Introduction

Accessing the Internet and specifically the World Wide Web from a mobile phone is common today. In the past mobile web access was purely in the domain of high-end smart phones since only those devices had the processing power to render and display HTML content. The cheaper feature phones were mostly bound to the Wireless Application Protocol (WAP) [6] part of the Internet. Further, packet-data was expensive in the past but today packet-data is relatively cheap all around the world (even without a special packet-data plan). Today most feature phones posses the processing power to display HTML pages. Therefore, current feature phones come with real web browsers that even support JavaScript. The combination of both, cheap packet-data and cheap phones with web browsing capabilities enable mobile web access all around the world. Especially users in developing countries seem to profit from these developments.

Due to the broader use of this technology we decided to investigate the privacy implications that arise through mobile phone web access. Our investigation specifically targets the so-called feature phones and low price range smart phones. We specifically do not investigate high-end smart phones such as the iPhone and Android-based devices, since those platforms are study quite well [5]. Our interest in the topic was further sparked since we stumbled across multiple online discussion forums[1] where this topic was discussed. The main issue was that those discussions did not come to any meaningful result since only rumors and very simple studies where presented by the individuals. Therefore, we decided to investigate for ourselfs. The only bit of information that seemed coherent is that some private information is leaked through HTTP [3] headers.

In this paper we presented a study on if and how private data is leaked through mobile phone-based web access. The study concentrates on data that is leaked through HTTP headers. One important point is that in order to access the leaked data communication does not need to be intercepted in any way. The data is delivered to any web server visited from a mobile phone. The main study was conducted over the period of 12 month, overall we spent 18 month on this project since we only slowly learned what kind of data exists and how we can find it. The work is separated in four parts. First, data acquisition. Second, identifying interesting HTTP header fields. Third, data analysis. Forth, determine how and why data is added to the HTTP connections in the first place. Relatively early in the project we decided to mainly look for leaked mobile phone numbers, but we briefly discuss the other kind of information we found. At the end of our study we created our mobile privacy checker where privacy concerned users can go and test if their private data is leaked.

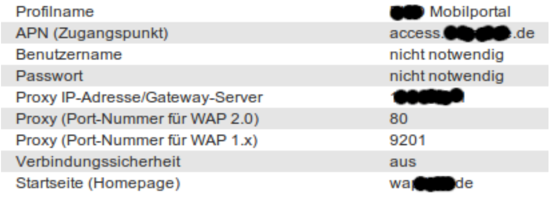
The main contributions of this paper are:

- We show that private data is leaked by many mobile operators around the world.
- We show that anybody owning a website that gets viewed from a mobile phone has the ability to collect personal information about his visitors.
- We show why this leak hasn't gotten any attention so far, this is because until now nobody knew what to look for exactly.

The rest of this paper is structured in the following way. In Section II we briefly describe how mobile phone Internet access works. In Section III we explain how we acquired

[1] http://www.php-resource.de/forum/showthread/t-95785.html
http://discussion.forum.nokia.com/forum/archive/index.php/t-5719.html
http://www.smart-techie.com/blog/2007/01/your-gprs-mobile-is-spying-on-you/

978-1-4244-7445-5/10/$26.00 ©2010 IEEE



Profilname	[redacted] Mobilportal
APN (Zugangspunkt)	access.[redacted].de
Benutzername	nicht notwendig
Passwort	nicht notwendig
Proxy IP-Adresse/Gateway-Server	[redacted]
Proxy (Port-Nummer für WAP 2.0)	80
Proxy (Port-Nummer für WAP 1.x)	9201
Verbindungssicherheit	aus
Startseite (Homepage)	wa[redacted]de

Fig. 1. Typical Internet settings that are provided by a mobile operator.

our data. In Section IV we describe in great detail how we analyzed our data. Section V shows where and why the data is injected into the HTTP connection. In Section VI we present our findings. Section VII discusses potential cases for abuse of such data, and in Section VIII we draw our conclusions.

II. Mobile phone Internet Access

Mobile phone Internet access depends on a few settings in the mobile phone. The Access Point Name (APN), user name and password (mostly unused in consumer contracts), and HTTP and WAP proxy address and ports. These settings are either pre-configured in the mobile phone or are configured into the mobile phone by either the mobile operator through Over-The-Air (OTA) Provisioning or the user by going to the phones settings and manually entering each configuration parameter. A typical set of configuration options for mobile phone Internet access is shown in Figure 1. The information was obtained from a mobile network operators website, we blurred out any information that are operator specific since this is not relevant.

III. Data Acquisition

Data acquisition for such a project is complicated for a very simple reason. How to get enough people to visit your website using their mobile phone. Luckily the author of this paper owns a website that falls in this criteria. A few years ago he developed some games for the J2ME[2] (Java 2 Micro Edition) platform and thus people visits his website[3] to download the games. Furthermore a big mobile gaming website embeds images (screen shots of these games) from the authors website, therefore, every time a visitor loads the relevant page at the gaming website a request is sent to our web server. Due to this circumstances we get quite a lot of relevant traffic.

The logging itself was done in two parts. First we added a rewrite rule to the configuration of our Apache[4] web server. The rewrite rule redirected all request that requested an image file to a script. *The script logged all HTTP headers send by the client* and returned the request image. The second part of the logging was done by adding a few lines of PHP[1] to the code that generates the web pages for the authors website. Therefore, every access to the authors website could be logged just as with the image requests. A typical data set is shown in Figure 2.

Further we acquired pre-paid SIM cards from all mobile operators in Germany in order to do some minimal testing ourselfs. We did not want to purely rely on external data. The hope was that one of the operators or the reseller of their pre-paid product would cause a data leakage that we could investigate further and analyze. Luckily one of our test cards caused such a data leakage. The complete log entry containing all HTTP headers is shown in Figure 2. The logged phone number is actually the phone number of the author's test SIM card, therefore, we are confident that the leaked data is a real privacy leak. *We want to emphasis that we do not want to discredit this operator. We just tested local operators and show the – for us – lucky result.*

IV. Data Analysis

We analyzed the data in two steps. In the first step we sort all HTTP headers by occurrence so that the header that was seen the least number of times is on top of the list. We did this since we were sure that only very few of the log entries would contain interesting headers, and, therefore, a low number of occurrences indicate interesting content. From this list we manually extracted interesting header names. We discuss what makes a header interesting further below. In the second step we filter for log entries that contain these interesting headers. Finally we build a tool to analyze the list of log entries that contain the interesting headers. In this step we extract information such as the MSISDN[5] (the mobile phone number) and the IMSI[6] (the unique ID of a SIM[7] card).

A. Identifying Interesting HTTP Headers

In order to be able to identify interesting HTTP header names one has to be familiar with some parts of the mobile phone service terminology. Words and abbreviations such as MSISDN, IMSI, IMEI[8], line ID, APN, `roaming`, and `subscriber` to name just a few. Armed with a list of words one can start hunting for header names. Once you get a feeling for it you constantly find more and more headers that look interesting. Our primary goal was to find headers that contain the users mobile phone number (MSISDN) so we started looking for fields that contain only numbers.

We had multiple strangely interesting finds during the search. For example the COOKIE header that contained a list of other HTTP headers including a header named MSISDN. Further we found that some of the headers contain the MSISDN in hex encoded form, this is shown in the first example in Figure 4.

V. The data leakage

Now that we found a substantial amount of private data in HTTP headers sent to our web server we investigated the

[2] http://java.sun.com/javame/
[3] http://mulliner.org/wj
[4] http://httpd.apache.org/
[5] Mobile Subscriber Integrated Services Digital Network Number
[6] Integrated Mobile Subscriber Identity
[7] Subscriber Identity Module
[8] International Mobile Equipment Identity

```
Header Name             Content
====================    ==============================================================
HOST                    mulliner.org
USER-AGENT              Mozilla/5.0 (X11; U; Linux armv7l; en-US; rv:1.9.2a1pre)
                          Gecko/20090928 Firefox/3.5 Maemo Browser 1.4.1.15 RX-51 N900
ACCEPT                  image/png,image/*;q=0.8,*/*;q=0.5
ACCEPT-LANGUAGE         en
ACCEPT-ENCODING         *
ACCEPT-CHARSET          ISO-8859-1,utf-8;q=0.7,*;q=0.7
REFERER                 http://mulliner.org/blog/
X-UP-SUBNO              1233936xxx-346677xxx
X-UP-FORWARDED_FOR      10.248.240.209
X-FORWARDED_FOR         10.248.240.209
X-UP-CALLING-LINE-ID    491522852xxxx
X-UP-SUBSCRIBER-COS     System,UMTS,SX-LIVPRT,A02-MADRID-1BILD-VF-DE,Vodafone,Prepaid,
                          Rot
MAX-FORWARDS            10
VIA                     1.1 rn2wwpsv161-ncl-0.wwp.vodafone.de
CONNECTION              close
REMOTE_ADDR             139.7.146.41
```

Fig. 2. Header information leaked by BILDmobil. The log was produced by the author using his own SIM card.

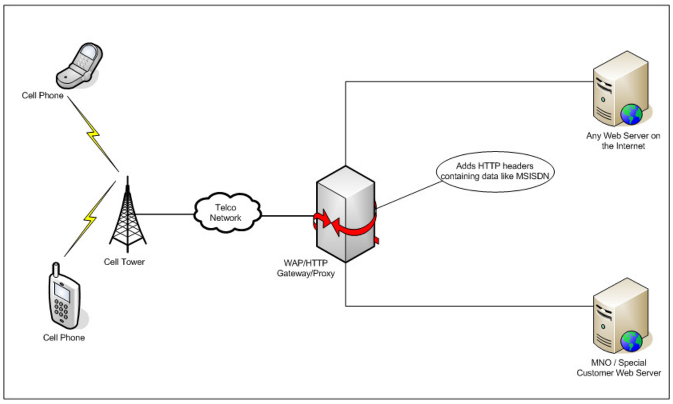

Cell Phone
Cell Tower
Telco Network
WAP/HTTP Gateway/Proxy
Adds HTTP headers containing data like MSISDN
Any Web Server on the Internet
Cell Phone
MNO / Special Customer Web Server

Fig. 3. The proxy adding private data to the HTTP connection.

cause for this.

We determine that the HTTP headers are added by the HTTP and/or WAP proxy on the mobile operators network. Multiple facts add to this conclusion. First, we know that some operators do this in order to provide their third party application developers with the means to identify users for billing purposes [2]. Second, a mobile phone does not store all the data that shows up in the various headers. Data that is not present on the phone can not be sent out. A good example is the subscriber number (X-UP-SUBNO shown in Figure 2) that we found in the log entry that we produced with our own SIM card. Third, we do not have any log entries from smart phones that normally do not use an HTTP proxy by default such as the iPhone or Android-based devices. Therefore, we conclude that the HTTP/WAP proxy is adding the relevant HTTP headers. Figure 3 shows a simplified picture of this setup.

VI. Findings

We split our findings in to two parts. In part one we discuss the technical part of what HTTP headers leak what kind of data. In part two we discuss our quantitative results.

A. *HTTP Headers*

Since we concentrated on identifying HTTP headers that contain the MSISDN we spent the most time on this subject. Figure 5 shows a list of all the HTTP headers *we identified* to contain the MSISDN in some form. The fact that twenty (20) different headers can contain the mobile phone number adds to the fact that nobody before us was able to point out the huge privacy problem of this leakage. Also notable is the COOKIE header since this header is normally used for another purpose, and, therefore, is clearly abused. Further we found headers such as X-FH-SUBSCRIBER-INFO and X-NETWORK-INFO that contain multiple pieces of information related to the users connectivity (including the MSISDN). Figure 4 shows some example of these combined headers.

B. *Quantitative Results*

In the previous section we discussed what HTTP headers contain what data. In this section we want to give a brief overview of how many individual mobile phone numbers (MSISDNs) we actually logged. We further show where the phone numbers belong to, meaning from which country.

We used the international country code number prefix to identify the country an individual phone number is registered at. The country codes are standardized by the International Telecommunications Union (ITU) and can be found in [4]. In addition we used the IP address of each data set (REMOTE-ADDR) to verify the name of the operator and country. This works well since all operators that serve multiple countries

```
COOKIE: User-Identity-Forward-msisdn=32363737313639xxxxxxxx;Bearer-Type=
 w-TCP;wtls-security-level=none;network-access-type=GPRS;Called-station-id=wap.mascom

COOKIE: User-Identity-Forward-msisdn=91943831xxxx;Bearer-Type=w-TCP;
 wtls-security-level=none;network-access-type=GPRS;roaming-information=no_info

COOKIE: X-SDP-MSISDN=4072404xxxx; Bearer-Type=w-TCP; wtls-security-level=none;
 network-access-type=GPRS

X-WAP-FH-SUBSCRIBER-INFO: IP=10.191.142.165,MSISDN=6013749xxxx,APN=postpaid.celcom3g

X-WAP-FH-SUBSCRIBER-INFO: IP=10.178.98.99,MSISDN=6013799xxxx,APN=prepaid.celcom3g

X-NETWORK-INFO: 3G,10.45.28.146,44798017xxxx,194.33.27.146,unsecured
```

Fig. 4. HTTP headers that contain multiple items.

Header Name	Count
X-UP-CALLING-LINE-ID	324
X-NOKIA-MSISDN	238
X-MSISDN	203
X-H3G-MSISDN	125
MSISDN	106
COOKIE	67
_RAPMIN	41
X-WAP-FH-SUBSCRIBER-INFO	16
X-FH-MSISDN	16
X-HTS-CLID	16
X-MSP-CLID	15
X-UP-LSID	12
X-JINNY-CID	7
X-NETWORK-INFO	5
X-MSP-MSISDN	4
X-NX-CLID	4
X-WAP-MSISDN	3
IGCLI	2
X-WSB-CLI	2
X-ORANGE-CLI	2

Fig. 5. HTTP headers that contain the MSISDN.

seem to have separate IP ranges for their mobile customers in each country. Therefore, the IP address can used to verify the country of origin of a individual data set. In some rare cases this was not possible here we spent more time verifying the origin.

Figure 6 shows the count of *unique* mobile phone numbers per country in our log file. We only list countries where we at least have two unique phone numbers. We have 13 additional countries where we only logged one individual number. This means we have collected 1183 phone numbers of 67 countries. Note, adding the numbers of Figure 5 will show a higher number this is the case since some log entries actually contain up to three headers that contain the MSISDN.

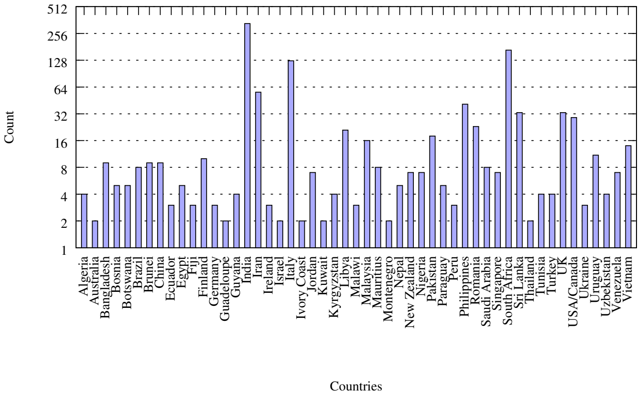


Fig. 6. Countries we collected more then one MSISDN from.

C. Other interesting data

Besides hunting for HTTP headers that contain the MSISDN we further analyzed our data set. We stumbled upon multiple interesting kinds of headers. These headers contain the IMSI (the unique SIM card ID), Bearer type (the type of communication channel), the roaming status, and the access point name (APN).

Figure 7 shows a list of HTTP headers that contain bearer information. Through looking at this header one can easily determine how a particular mobile phone is connected to the Internet. What is unique about the bearer header is that it is inserted by the mobile phone itself.

Figure 8 shows a list of HTTP headers that indicate of the subscriber is currently roaming (out side of his home network).

Some of the examples in Figure 4 also contain the APN. Also additional headers exist that just contain the APN.

VII. Abuse

The main abuse of this information leak is user tracking. In some countries one can do a reverse lookup of phone numbers. This means for those countries one can take a mobile phone number and actually determine the name (first and last) and

Header Name	Observed Content (separated by comma)
X-NOKIA-MUSICSHOP-BEARER	WLAN,GPRS/3G
X-NOKIA-BEARER	GPRS,UMTS-p,GPRS or 3G,CSD,UMTS,258,2G,4,3G
X-UP-BEAR-TYPE	GPRS,WCDMA,gprs,GPRS/EDGE
BEARER-INDICATION	11,0,gprs,gsm_gprs_ipv4
X-UP-BEARER-TYPE	GPRS,128,0
X-BEARER-TYPE	GPRS,UMTS
NOKIA-BEARER	GPRS
BEARER	GPRS
NEW-BEARER-HEADER	GPRS

Fig. 7. HTTP headers that contain bearer information.

Header Name	Observed Content (separated by comma)
X-ORANGE-ROAMING	YES,NO
X-ROAMING	True,Yes,False,NO
X-SDP-ROAMDING	True,False
X-NOKIA-ROAMING	0

Fig. 8. HTTP headers that contain roaming information.

maybe even the address of the visitors of his website. *Das Telefonbuch*[9] offers such a backward search for Germany.

User tracking is very likely since the phone number is internationally unique. Further the phone number will stay even if the user changes to a new mobile phone (e.g. after an upgrade or warranty replace). This enables reliable longterm tracking possibilities.

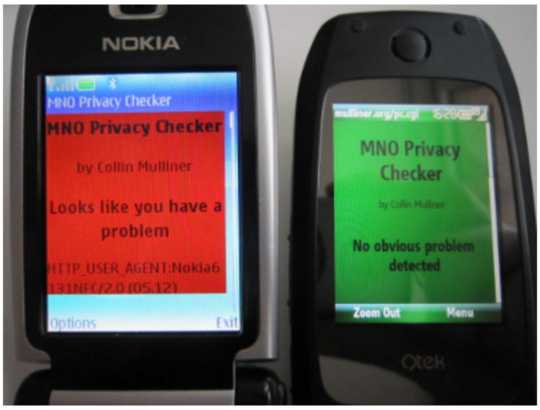
Large commercial and social networking sites that operate a mobile version of their site can collect this information to enrich the profile data that they already have about their users.

A. Counter Measures

The true solution is to change the configuration in the HTTP proxies operated by the mobile network operators and the external companies that run those proxies for the operators. In the best case the data injection is completely turned off. If the data is actually used by either the operator himself or by authorized third parties, measures must be taken so that the data is only injected into connections that terminate inside the operator network or at hosts belonging to authorized third parties.

The user has actually no way of disabling the data leakage himself besides disabling the HTTP proxy in the configuration of his mobile phone. This may not be possible since the users mobile phone contract conditions may require using this specific proxy.

In order to provide concerned users with means to check if their mobile service operator for HTTP header privacy leaks we have setup a simple web page. When a user browses to our page `http://www.mulliner.org/pc.cgi` we analyze the transmitted HTTP headers and give an instant feedback to the user. The feedback contains a general YES or NO feedback and in addition we display the transmitted headers that raised our concern.



MNO Privacy Checker

by Collin Mulliner

Looks like you have a problem

HTTP_USER_AGENT:Nokia6131NFC/2.0 (05.12)

MNO Privacy Checker

by Collin Mulliner

No obvious problem detected

Fig. 9. Our mobile network operator privacy checking website.

VIII. Conclusions

In this paper we present a study on privacy leaks of mobile phone web access. The privacy leak is related to the HTTP proxies operated by the mobile phone network operators. The proxies inject additional headers into HTTP connections and thus cause the privacy leak. The leaked data can be acquired by anybody who runs a website that gets visited from a mobile phone.

Through our study we now know why this privacy leak has not been detected by other people like those discussing in various online forums. The answer is the large number of different HTTP headers that carry the information. Looking only at one particular HTTP header does not provide you with the real picture, one needs to look at all headers. Also the kind of headers used seems largely dependent on the operator.

The privacy leak is not necessary and could potentially be exploited by malicious parties. Attacks based on the leaked data could range from longterm user tracking since some of the data is depended on the users account at the operator rather

[9] http://www.dastelefonbuch.de

than being bound to the users handset. The real names of website visitors could be determined through the use of reverse lookup databases that map phone numbers to real names.

References

[1] PHP: Hypertext Processor. http://www.php.net.

[2] AT&T. WAP 2.0 User Identification for Secure Services. http://developer.att.com/devcentral/tools_technologies/technologies/docs/WAP_2-0_User_Identification_Secure_Services.pdf, 2007.

[3] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. Hypertext transfer protocol – http/1.1, 1999.

[4] International Telecommunication Union. LIST OF ITU-T RECOMMENDATION E.164 ASSIGNED COUNTRY CODES. http://www.itu.int/dms_pub/itu-t/opb/sp/T-SP-E.164D-2009-PDF-E.pdf, April 2009.

[5] Nicolas Seriot. iPhone Privacy. http://seriot.ch/resources/talks_papers/iPhonePrivacy.pdf, 2010.

[6] WAP Forum. WAP-230-WSP Wireless Application Protocol Wireless Session Protocol Specification. http://www.wapforum.com, 2001.