

Mallory – a threat to your mobile device?

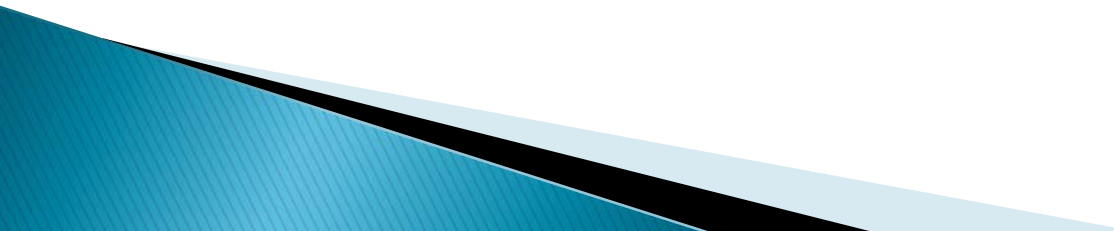
Lynn Margaret Batten

IT Security Research Services
&
Deakin University, Melbourne, Australia

June 2015



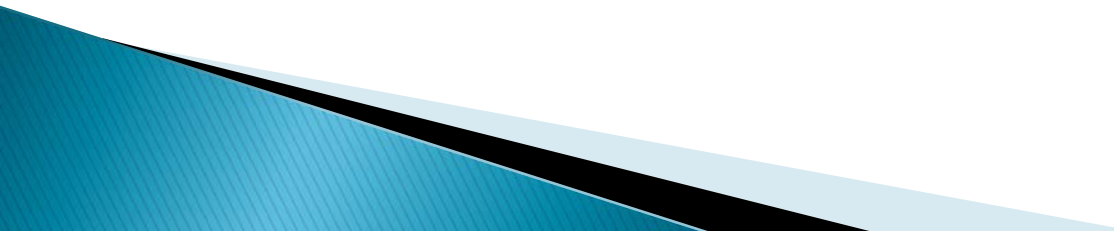
OUTLINE OF THE TALK

- Smart devices versus PCs
 - *WebView* versus *Web 2.0*
 - Tracking
 - Certificates
 - Mallory
 - Countermeasures.
- 

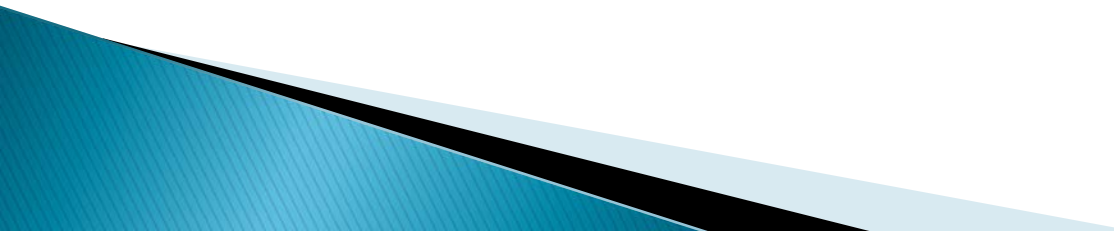
On a desktop machine:

- ▶ SSL secures transmissions between a browser and a web server.
- ▶ Applications are **Web-based** using *Web 2.0* technologies (allowing creation and sharing of online information) and can be displayed on the in-built web browser.
- ▶ Browsers have a pre-installed list of trusted CAs and communicate with a web server that has been issued a certificate from one of these CAs.
- ▶ The initial step is an exchange of SSL certificates issued by a Certificate Authority.

On the smartphone platform:

- ▶ APPS can be either web- or client-based.
 - ▶ **Client-based** applications use the *WebView* class to host HTML content in an APP.
 - ▶ Smartphones are sold with a set of pre-installed root certificates and APPs.
- 

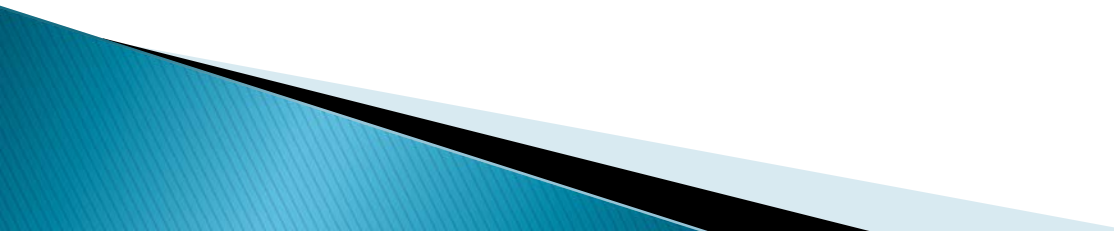
Web 2.0 versus WebView

- ▶ A **Web-based** application uses *Web 2.0* technologies: creation and sharing of online information, and display on the in-built web browser.
 - ▶ Mobile operating systems provide a client-based web container, called 'WebView' which hosts HTML content in an APP.
 - ▶ Mobile browsers provide less support for display of connection details and less warning of mixed content on html pages than desktop browsers.
- 

Web 2.0 versus WebView security

- ▶ Two essential pieces of Web 2.0's security infrastructure are weakened if WebView and its APIs are used:
 - the Trusted Computing Base at the client side, and
 - the sandbox protection implemented by browsers.
- ▶ Using [addJavascriptInterface\(\)](#) allows JavaScript to control your Android application. This can be a dangerous security issue.
- ▶ With WebView, many attacks can be launched either against APPs or by them; in particular, the current approach to sandboxing to test potential malware is impeded. [1,2]

THE ROLE OF ADVERTISING

- ▶ Developers include advertising libraries provided by official APP sites or by third-party advertising companies (e.g. Flurry, InMobi).
 - ▶ APP developers earn revenue from in-application advertisements and are encouraged to market their APPs free of charge; the more advertising libraries embedded in their APPs, the higher the revenue.
 - ▶ They also want to identify user preferences in order to offer customized services; for this identifying the user and/or device is necessary.
- 

How this information is obtained

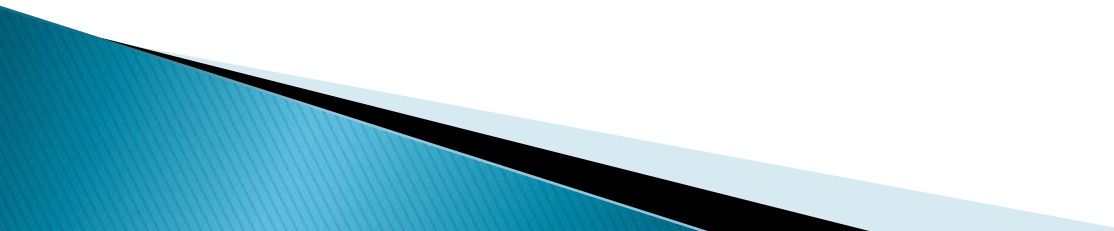
- ▶ There are several common methods of obtaining the above information. These include:
 - Malware
 - Permission system abuses via APPs
 - MiTM attacks
 - Certificate compromise.

I will consider each of them in this talk.

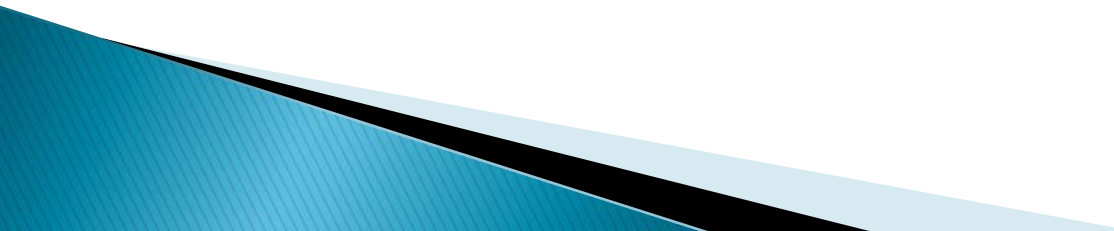


MALWARE

Malicious Software:

- ▶ You may download software that can monitor where you go online and record your keystrokes.
 - ▶ This allows the software to record confidential Internet banking passwords, logons, and other personal information.
 - ▶ Criminals can then access that information to commit fraud.
- 

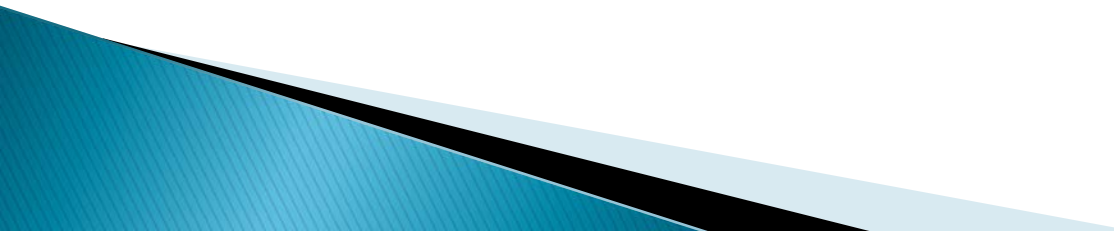
SMARTPHONE APPS and MALWARE

- ▶ Smartphones offer APPs from both official sites and third-party markets.
 - ▶ Official markets regularly test APPs to make sure they do not contain malware.
 - ▶ In third-party markets, APPs are not checked to determine if they are safe.
 - ▶ Individuals can post APPs on third-party markets which look similar to official market APPs but which contain malware.
- 

SOME APPS YOU MAY HAVE ON YOUR SMARTPHONE:



SECURING APPS

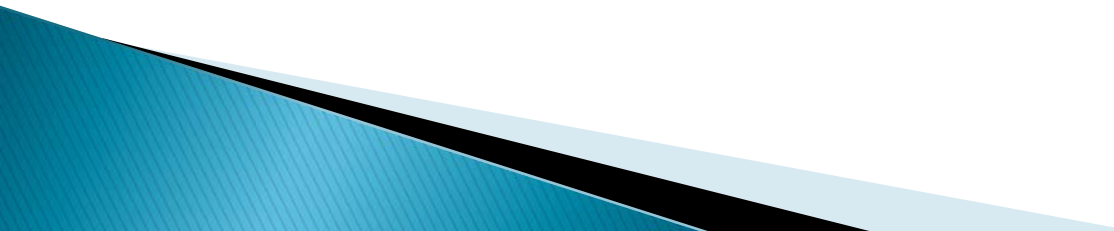
- ▶ All operating system providers attempt to protect the user from abuse of APPs.
 - ▶ The Android (Google) framework asks the user at APP install-time to authorise connections the APP may need to make.
 - ▶ Microsoft, Blackberry and Apple all have their own ways of achieving security goals to protect the user.
- 

SAMPLE PERMISSIONS


Most commonly
used permissions

Permission ID	Permission Name
<i>pms0001</i>	<i>INTERNET</i>
<i>pms0004</i>	<i>WRITE_EXTERNAL_STORAGE</i>
<i>pms0005</i>	<i>READ_PHONE_STATE</i>
<i>pms0006</i>	<i>ACCESS_NETWORK_STATE</i>
<i>pms0007</i>	<i>VIBRATE</i>
<i>pms0011</i>	<i>READ_LOGS</i>
<i>pms0013</i>	<i>RECEIVE_BOOT_COMPLETED</i>
<i>pms0021</i>	<i>SEND_SMS</i>
<i>pms0023</i>	<i>ACCESS_WIFI_STATE</i>
<i>pms0030</i>	<i>READ_SMS</i>
<i>pms0031</i>	<i>WRITE_SMS</i>

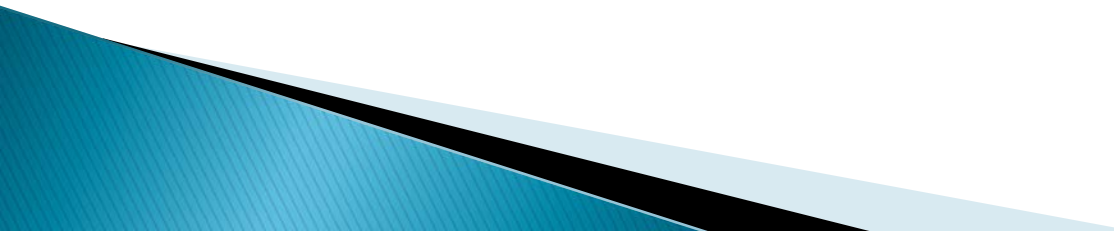
ADVERTIZING AND APPS

- ▶ The business model – major challenge in the development of APPs.
 - ▶ Solved by means of advertising revenue.
 - ▶ Google offers an APP software development kit that enables Android developers to add advertising libraries into their applications to generate revenue.
 - ▶ Third-party application developers are motivated by the revenue earned from APP advertising and embed many ads in their APPs.
- 

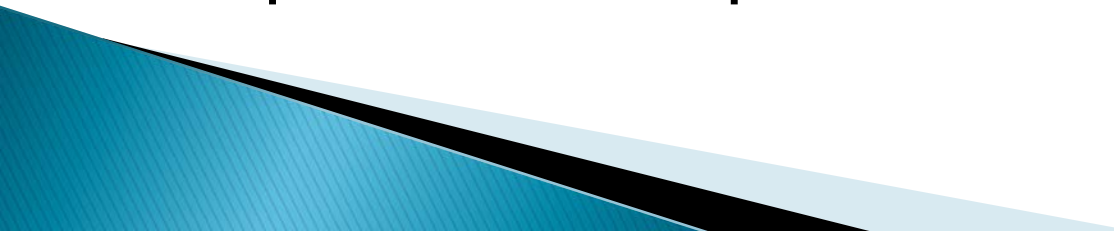
BENEFITS & THREATS OF ADS

- ▶ From an ad, the OS connects to an advertizing server which might collect the IMEI code or the IMSI number found in the sim card, thus identifying the mobile device.
 - ▶ Such identification allows developers to offer customized services.
 - ▶ Researchers have found APPs in the APP markets which send these phone identifiers to developers without informing the user.
- 

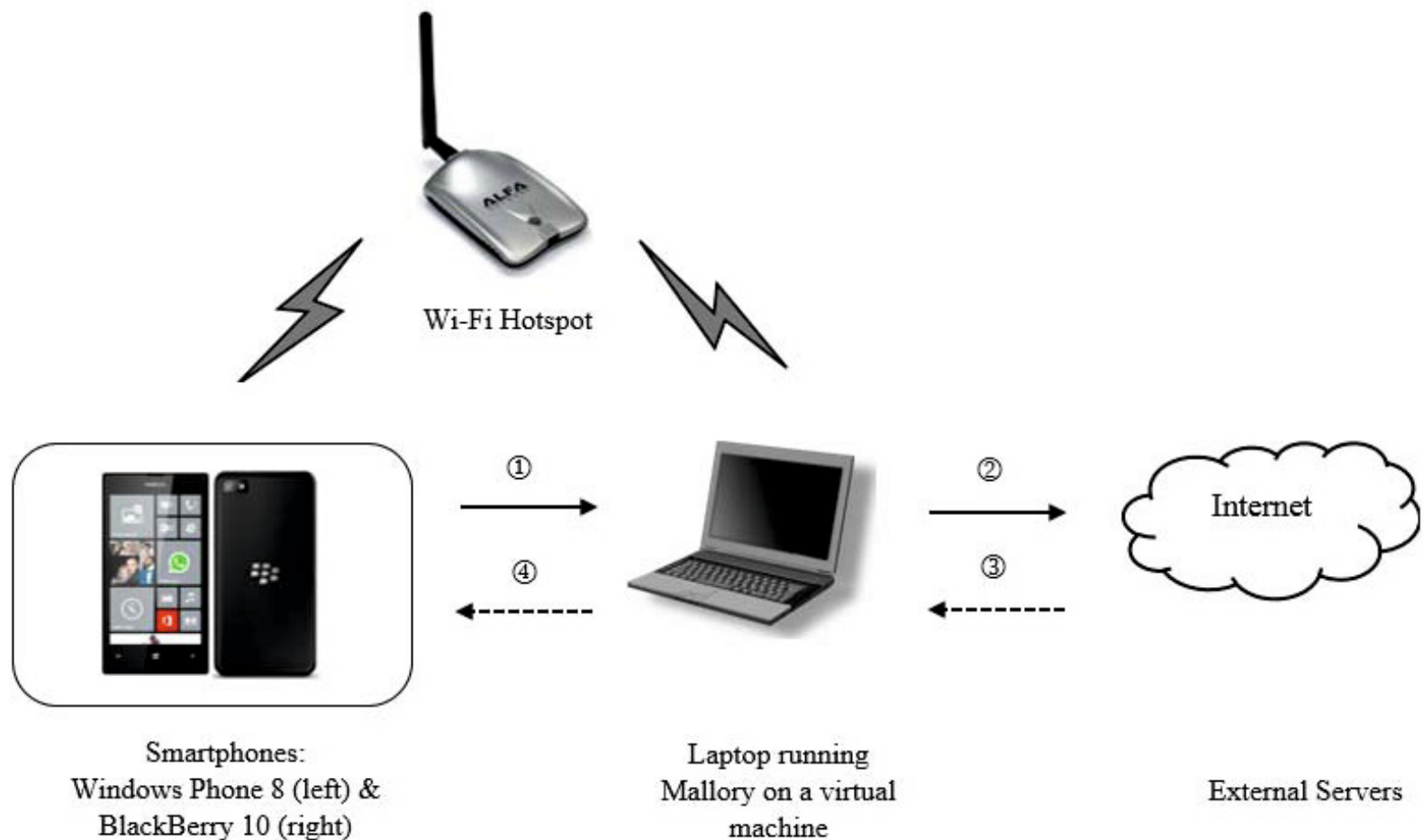
TRACKING SETTINGS

- ▶ On smartphones, all major operating systems (Google, Apple, Microsoft, Blackberry) allow the user to adjust tracking settings related to *Location Services* and to *Advertizing*.
 - ▶ The setting can usually be turned 'off' or 'on'; in some cases tracking cannot be turned off but can be 'limited'.
- 

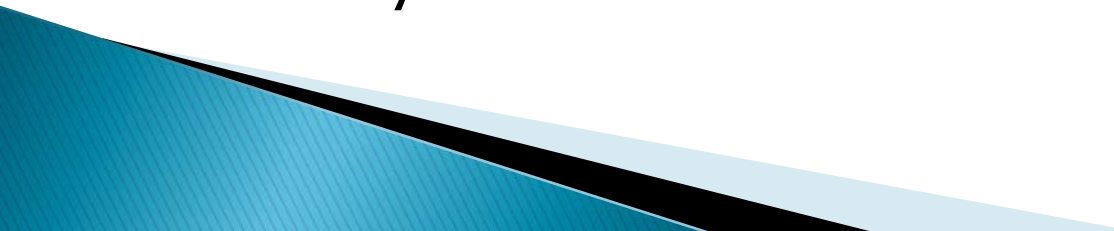
MY TEAM'S TEST SETUP

- ▶ My team decided to check these settings to determine how well they worked.
 - ▶ We set up a method based on easily available software linking to wifi connections.
 - ▶ The software we used is called *Mallory*.
 - ▶ We used the setup to test what data is captured when phones are being tracked.
- 

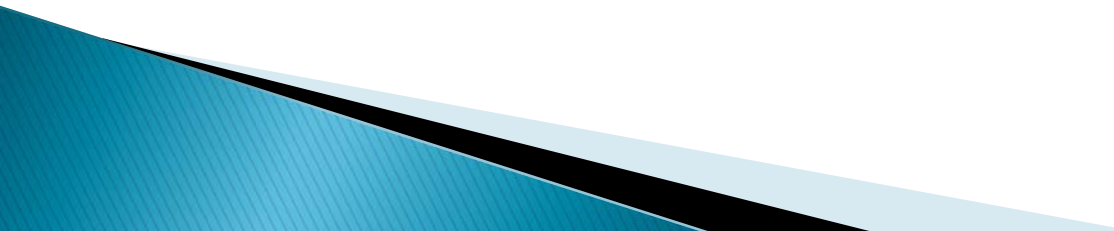
THE MALLORY WIFI INTERCEPT SETUP



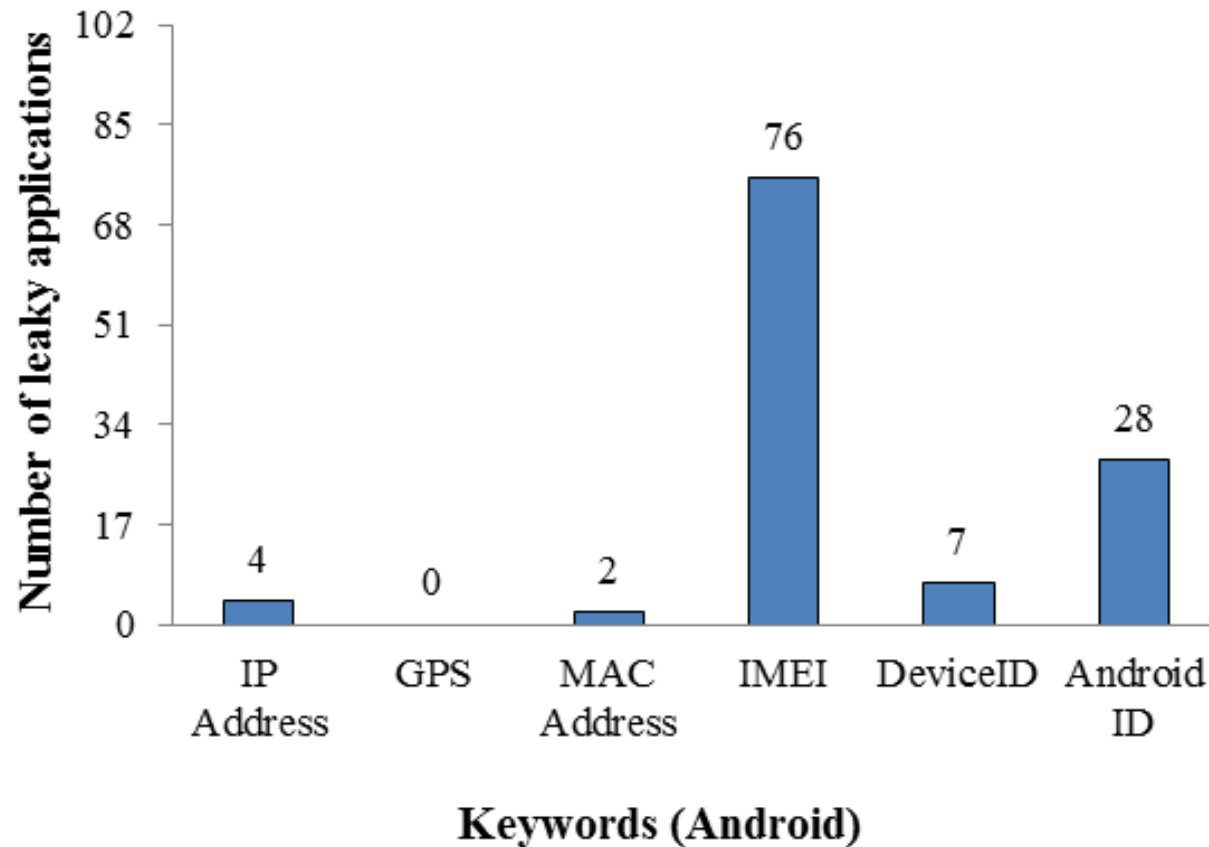
CONTRIBUTIONS OF MY TEAM IN THIS AREA

- ▶ We demonstrate that, without any explicit mention of it in the Terms and Conditions agreement, advertising libraries can access the mobile device's Device ID and Subscriber ID.
 - ▶ We estimate that there are, on average, three advertising libraries included in any application downloaded from third-party markets.
 - ▶ We observe that Android APPs that make use of permission systems are also likely to track the activity timeline of a user.
- 

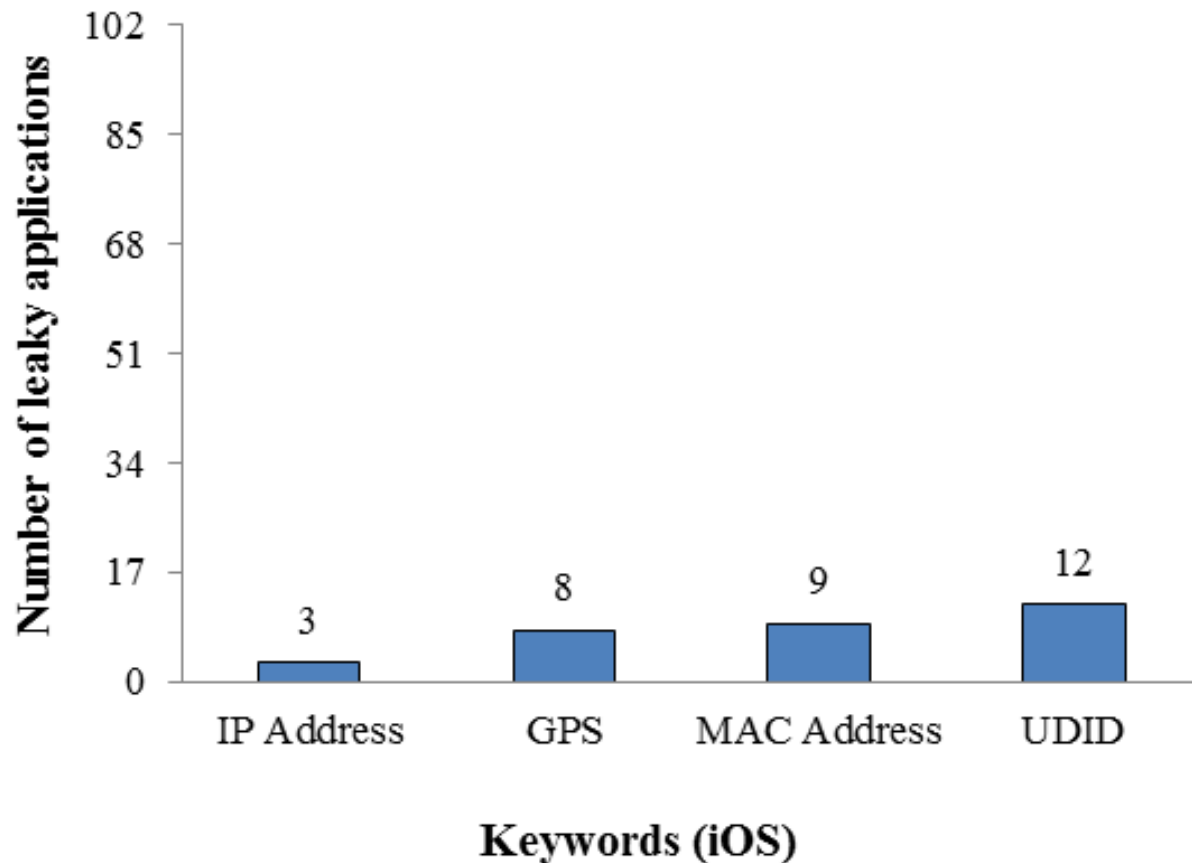
WHAT IS TRACKED?

- ▶ For location, the network address and global position (GPS) are obtained.
 - ▶ For advertising, the unique identifiers of the device and of the SIM card are obtained.
 - ▶ The user would normally expect that if she turns 'off' a tracking setting, then none of this information would be collected.
 - ▶ My team's research showed that this is not the case:
- 

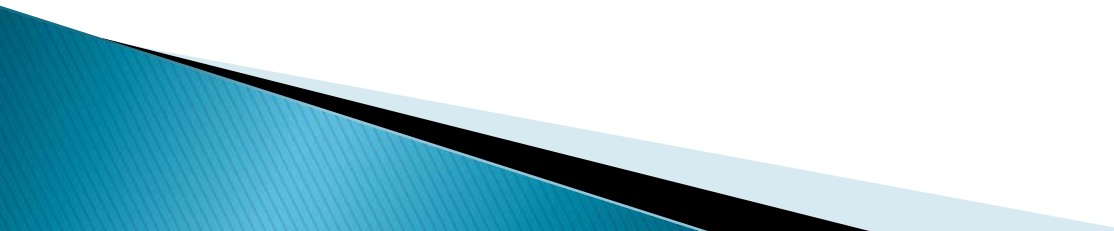
Data Leaked from Android APPs with Tracking OFF (from a sample of 102)



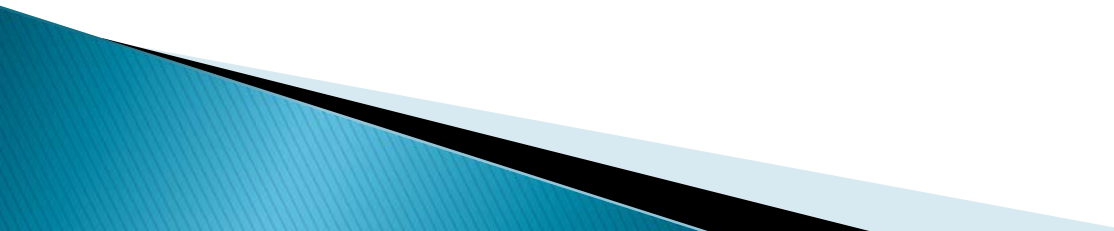
Data Leaked from iOS APPs with Tracking OFF (from a sample of 102)



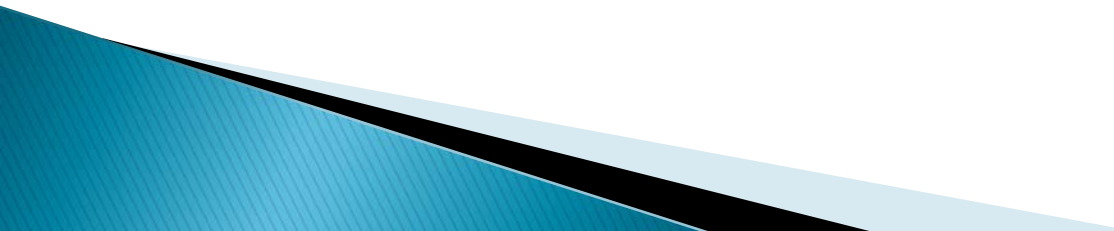
HOW ABOUT TRACKING ON?

- ▶ My team also determined that when tracking (for either location or advertising) was turned on, the user's smartphone was not always tracked.
 - ▶ We did similar work, obtaining similar results, with Blackberry and Windows 8 smartphones.
 - ▶ APPs for iOS and Android were chosen if they were developed by the same developer; this was also true for Blackberry and Windows APPs. (So we could observe differences in developer behaviour.)
- 

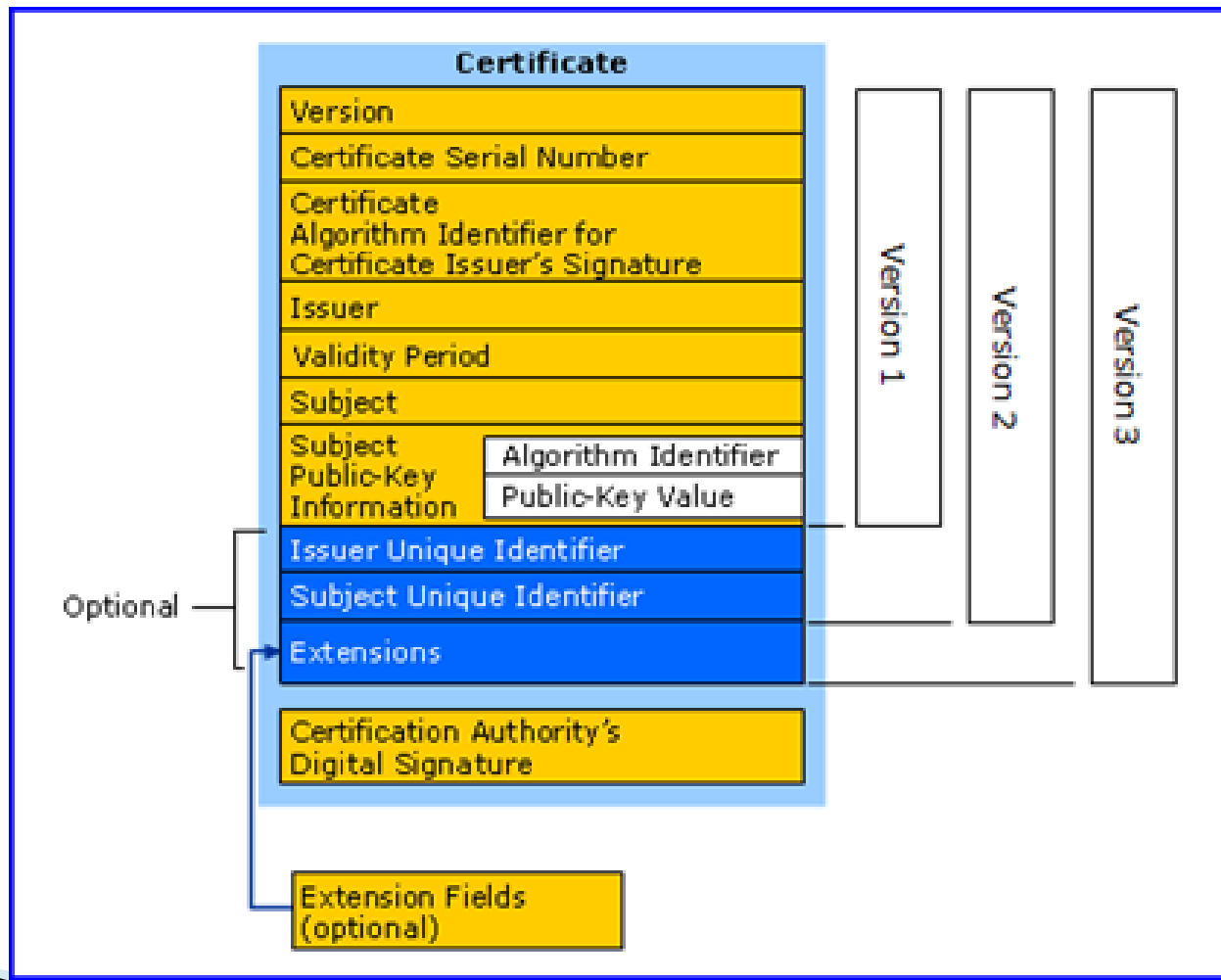
SUMMARIZING OUR RESULTS

- ▶ Applications are capable of leaking phone-related information without the user's knowledge.
 - ▶ Third-party advertising libraries were the principal cause of all the information leaks that were recorded for our datasets.
 - ▶ Apps are not updated to reflect any updated protections of the host OS.
 - ▶ *Since APPs with advertising are often not malicious, they are not identified by anti-virus software.*
- 

TAKING MALLORY FURTHER

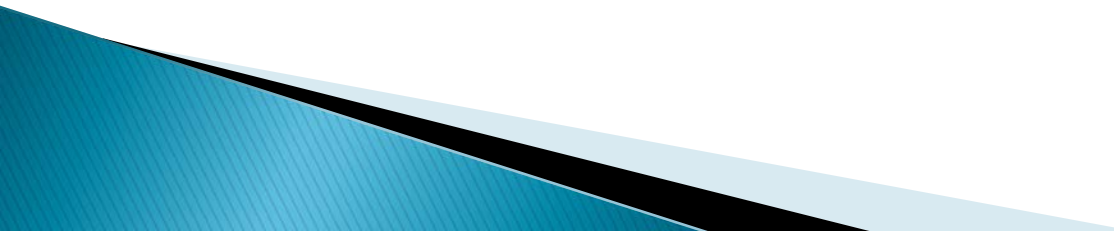
- ▶ Once we had Mallory set up between a device and a server, we tested its use in ‘Man-in-the-Middle’ attacks.
 - ▶ In particular, we were interested in capturing and reading data between the phone and a server.
 - ▶ Such connections, especially to ‘secured’ websites are based on certificates used to ensure a ‘trusted relationship’.
- 

An X.509 certificate contains the digitally signed ID of the issuer

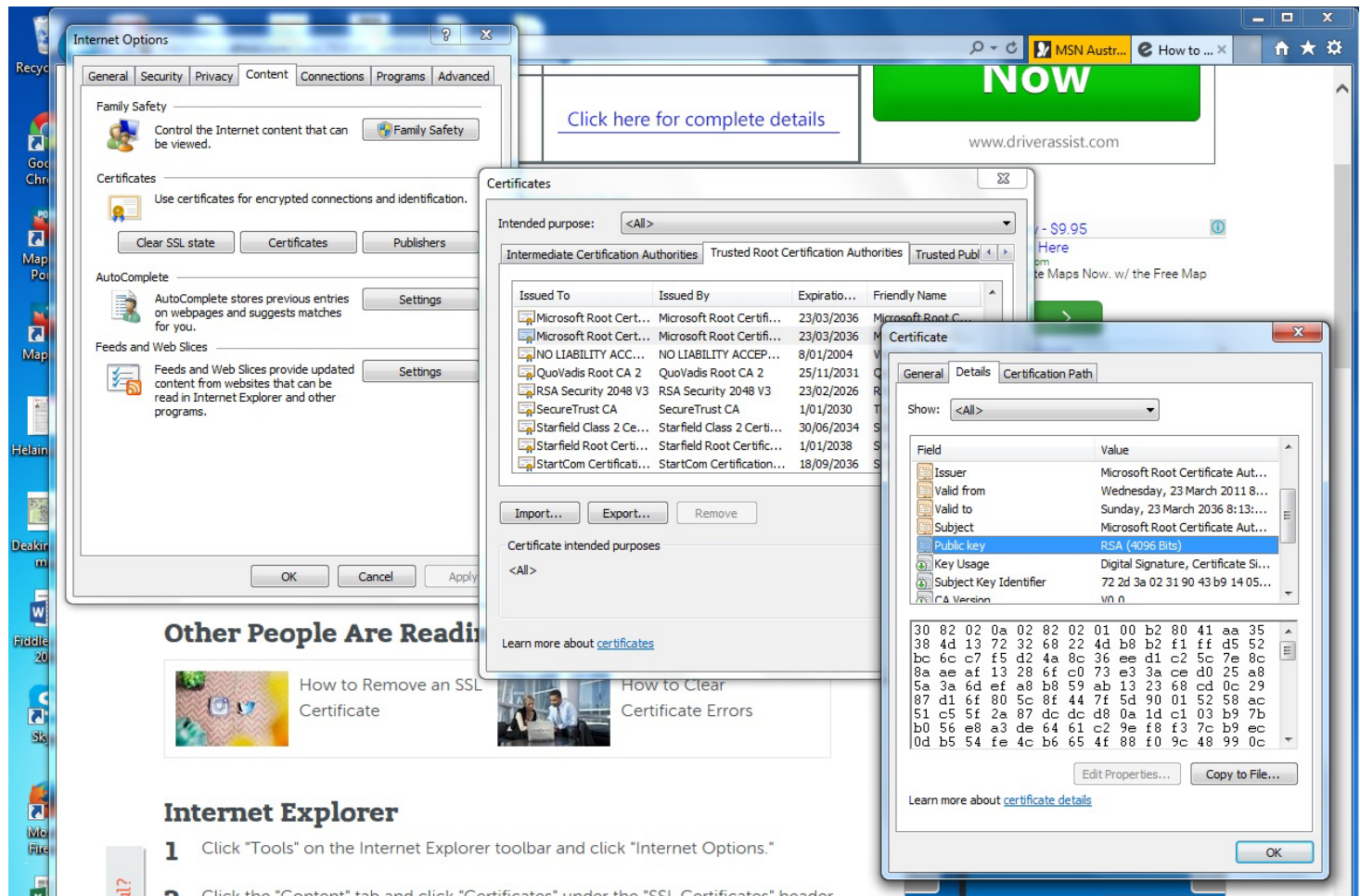


Certificate Verification

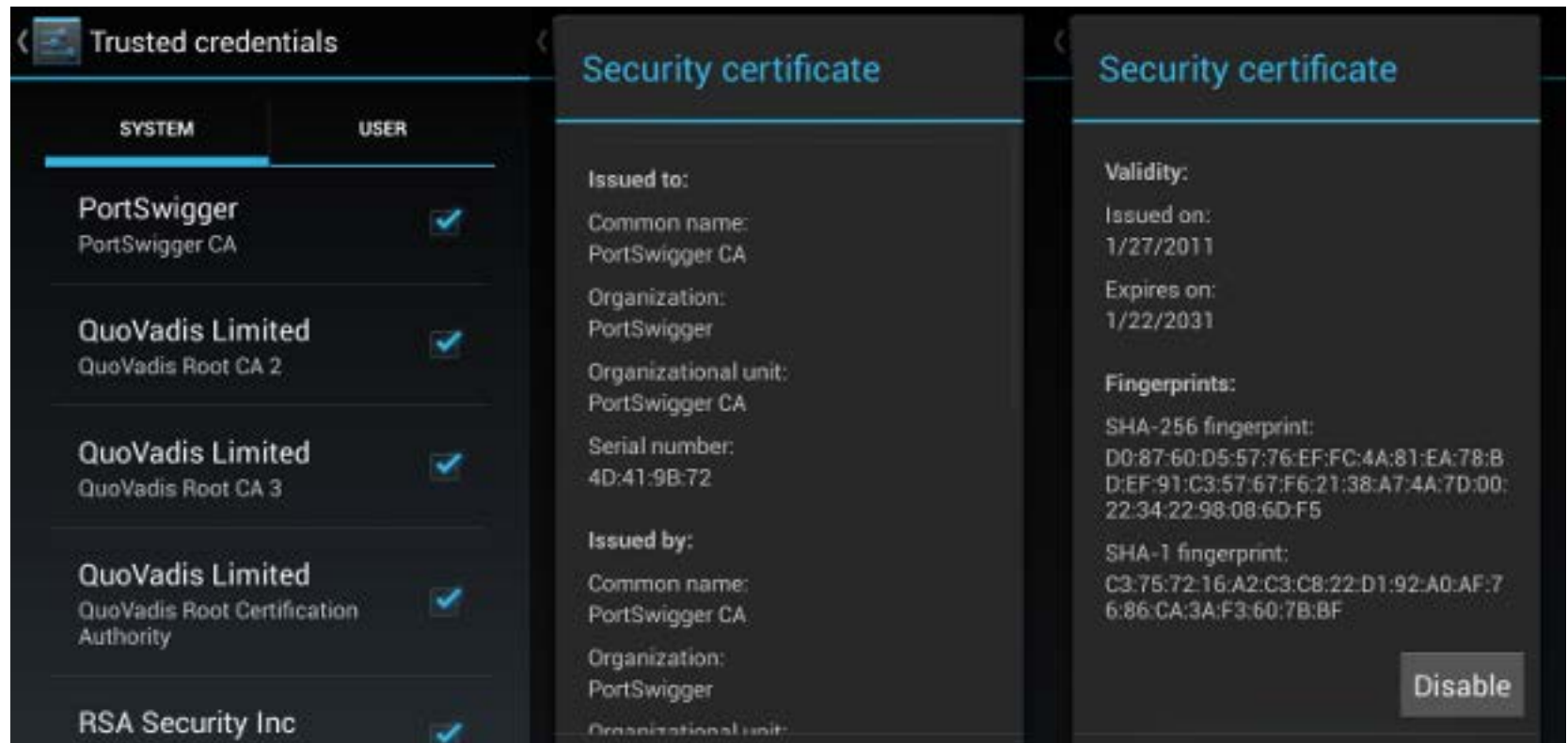
A digital certificate links a subject identity and a public/private key in a signed and therefore verifiable digital document.

- ▶ The subject of the certificate must match the resource subject (eg.URL)
 - ▶ The validity period must be within the time frame the certificate is planning on being used (and must be unrevoked).
 - ▶ The certificate must be used by a trusted Certificate Authority. (Match with an existing certificate will do.)
- 

Example of a web browser certificate



Disabling some of the 160 Root Certs on an Android Smartphone



MISUSE OF CERTIFICATES:

- ▶ From a study quoted in [*] on https use, of 13,500 Android APPS tested, over 1000 did not validate the host name.
- ▶ Any CA can issue a browser-acceptable certificate for any site.
- ▶ “In the research literature, it is becoming more common for threat models to assume an adversary possesses a valid certificate for a targeted site.”

[*]. *Clark and van Oorschot (2012). ‘SoK:SSL and HTTPS’. Proceedings of the IEEE Symposium on Security and Privacy.*



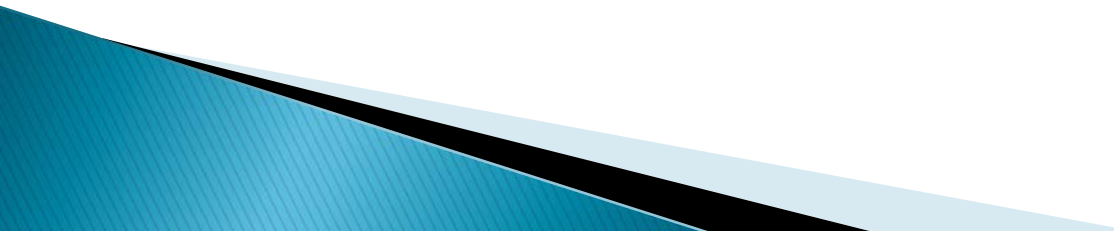
Man-in-The-Middle Attacks

The Mallory setup is described at

<https://bitbucket.org/IntrepidusGroup/mallory/wiki/Home>

- ▶ We set up MiTM attacks against smartphones Android v.4+, iOS v.6.2, Blackberry Z10 and Windows 8,
- ▶ using laptop software:
 - Oracle VM VirtualBox
 - Ubuntu

Man-in-the-Middle Attacks

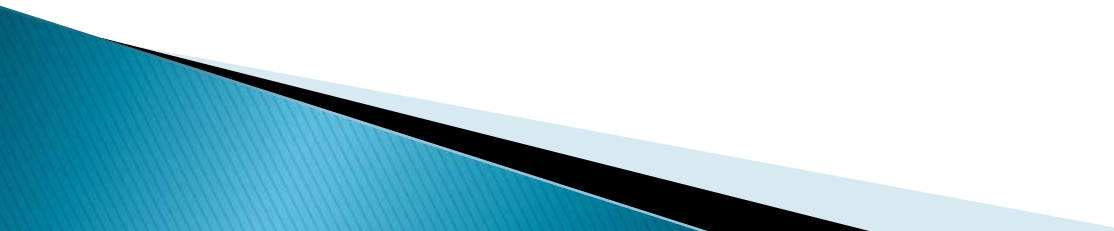
- ▶ MiTM is an active eavesdropping attack.
 - ▶ The attacker inserts himself between the client/server communication flows.
 - ▶ Once inserted, the attacker relays traffic to and from the client and server without either endpoint noticing the presence of a third-party.
 - ▶ Attackers are now focusing on smartphone users as their MiTM victims.
 - ▶ We describe three popular attacks (SSL Hijacking, SSL Stripping, DNS Spoofing) which target smartphone applications.
- 

SSL HIJACKING

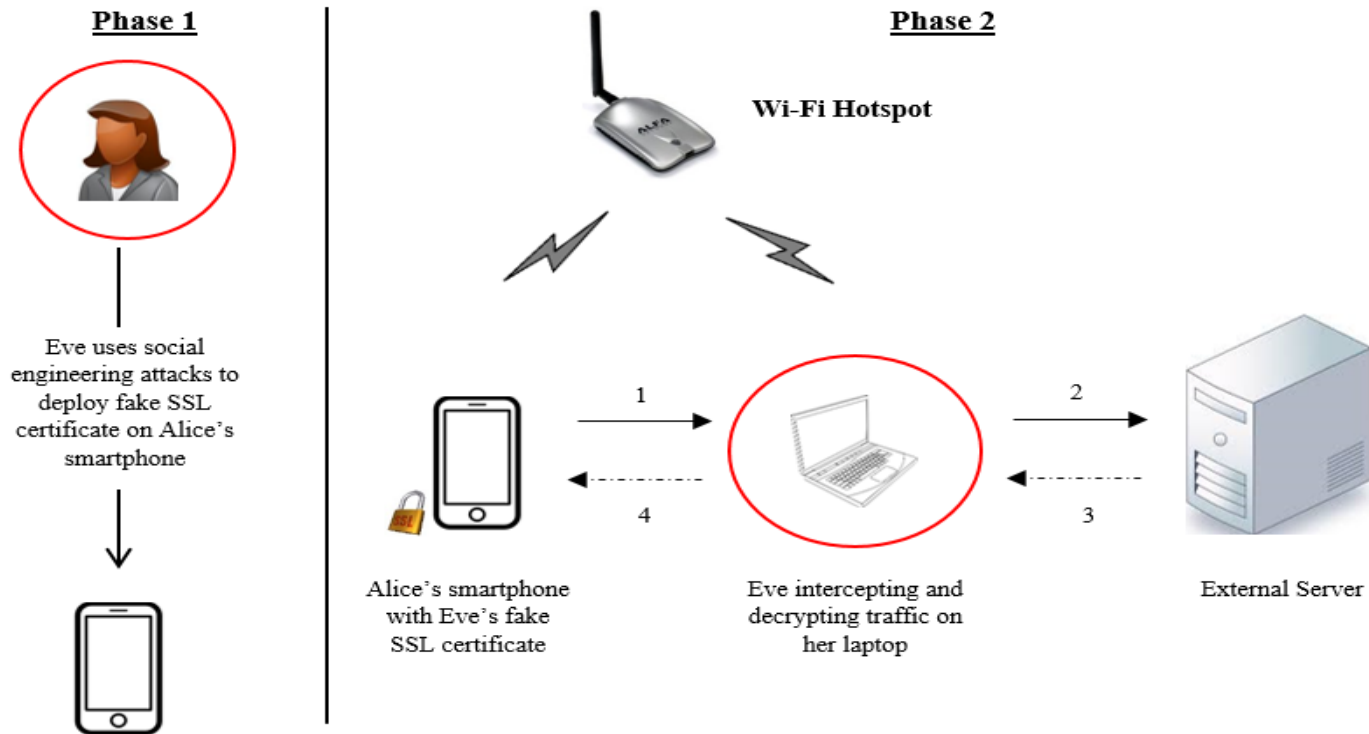
▶ Phase 1:

- Using social engineering, Eve finds out Alice's favourite games, and
- tricks her to install a free application.
- The free app contains a fake SSL certificate.

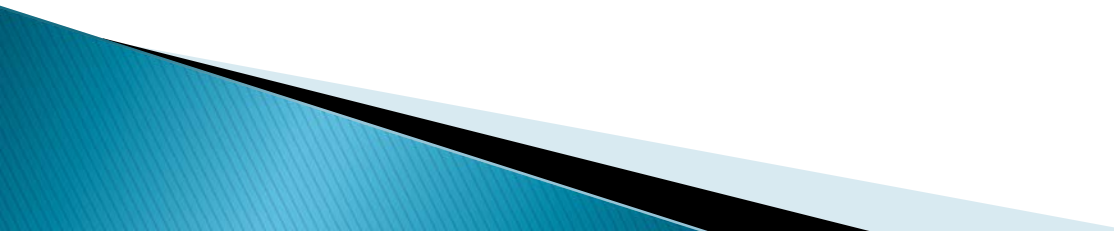
▶ Phase 2:

- Eve sets up a wifi hotspot near Alice's device, and
 - captures and decrypts all the traffic from and to the device
 - – as shown in the next Figure.
- 

SSL HIJACKING DIAGRAM



SSL STRIPPING

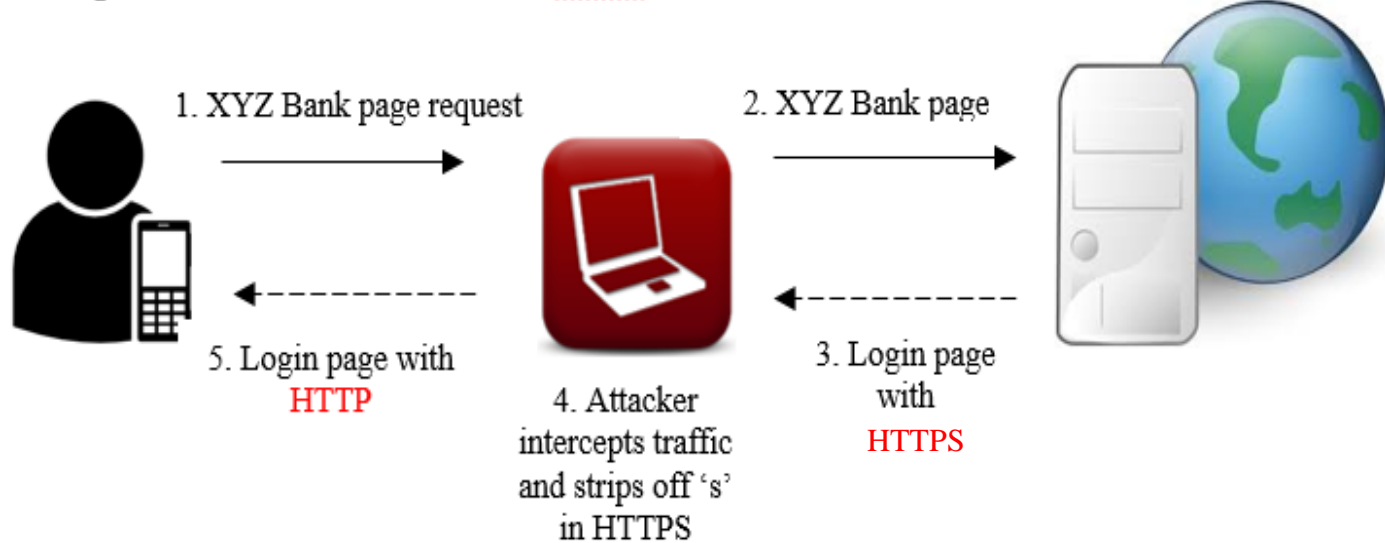
- ▶ When a user types https, http or part of a URL, normally, the application directs the traffic over an HTTPS connection;
 - ▶ The victim believes he is communicating over an HTTPS connection.
 - ▶ The MiTM SSL Strip attack intercepts the HTTPS redirect and maps the link to its HTTP equivalent.
 - ▶ The attacker communicates with the server over an HTTPS connection, while the client (unknowingly) receives traffic over an HTTP connection.
 - ▶ This is depicted in the next Figure.
- 

SSL STRIPPING DIAGRAM

Client using browser

MiTM Attacker

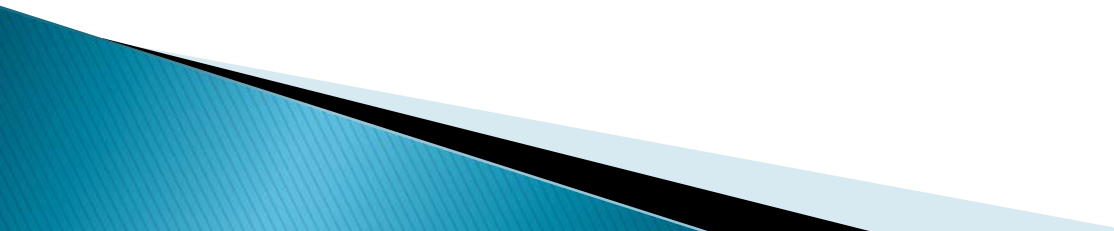
XYZ Bank Server



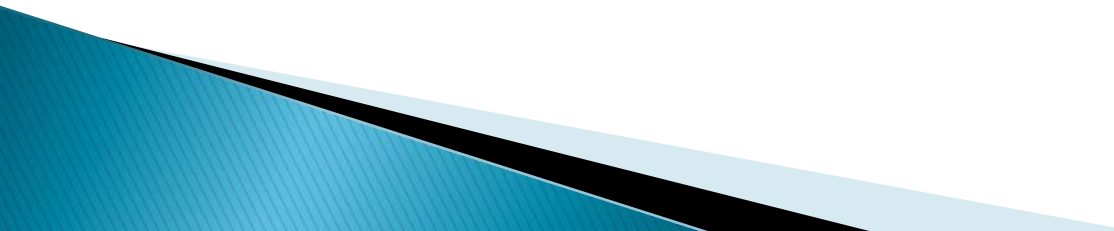
DNS SPOOFING

- ▶ This MiTM attack targets the DNS protocol which translates logical web addresses into their corresponding IP addresses.
- ▶ To carry out DNS Spoofing, the attacker
 - intercepts a DNS query,
 - extracts its unique ID, and
 - creates a fake DNS response for the client.
- ▶ Currently, such attacks cannot be easily detected on the smartphone.

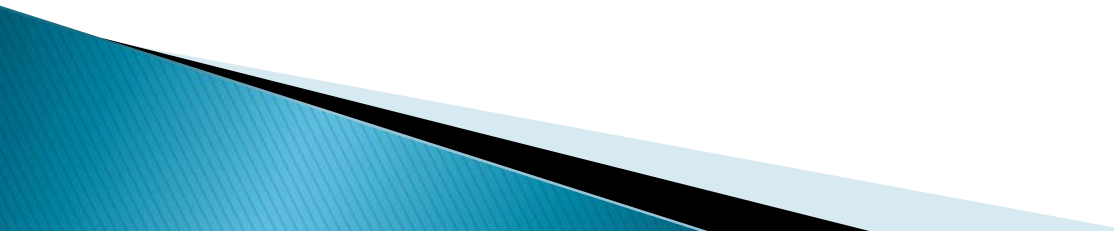
SSL PINNING

- ▶ Attempts to prevent SSL Hijacking.
 - ▶ Ensures that the application checks the server's certificate against a known copy bundled in the application before it is deployed on the market.
 - ▶ Is the responsibility of the application developer.
 - ▶ The developer specifies in the APP source code the certificates that should be trusted.
- 

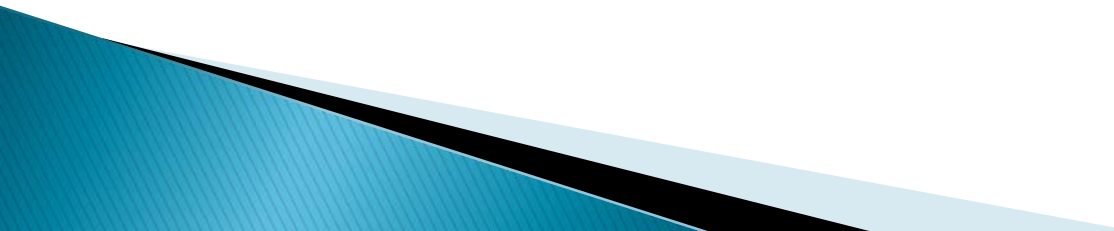
LIMITATIONS OF SSL PINNING

- ▶ Relies heavily on correct implementation by the developer.
 - ▶ Can be disabled by reverse engineering the application and forcing it to accept spoofed SSL certificates.
 - ▶ Developers using third-party advertising libraries are required to consent to the use of the certificates provided by the advertising companies.
 - ▶ Implementation of SSL Pinning varies depending on the host OS.
- 

DNS and DNSSEC

- ▶ The Domain Name System (DNS) is a query mechanism linking logical names to IP addresses.
 - ▶ No authentication checks are done during this process, which provides opportunities for attackers to divert traffic via MiTM proxies.
 - ▶ DNSSec (introduced in 1997) uses Public Key Cryptography to authenticate the origin of data and data integrity.
 - ▶ Digital signatures are computed for legitimate URLs and stored;
 - ▶ When directing to an IP address, a digital signature is computed and checked against the stored data.
- 

LIMITATIONS OF DNSSEC

- ▶ Bandwidth and storage requirements are increased by about a factor of 6 over DNS.
 - ▶ The most important advantages of enhanced DNS transaction security can be reached using existing infrastructures and technologies.
 - ▶ The amount of software that allows implementation of DNSSec on DNS servers is limited.
 - ▶ All the layers including the Root zone have to use the same digital signing algorithm.
- 

SETTING UP MALLORY*

Step 1: Set up Virtual Machine (VM)

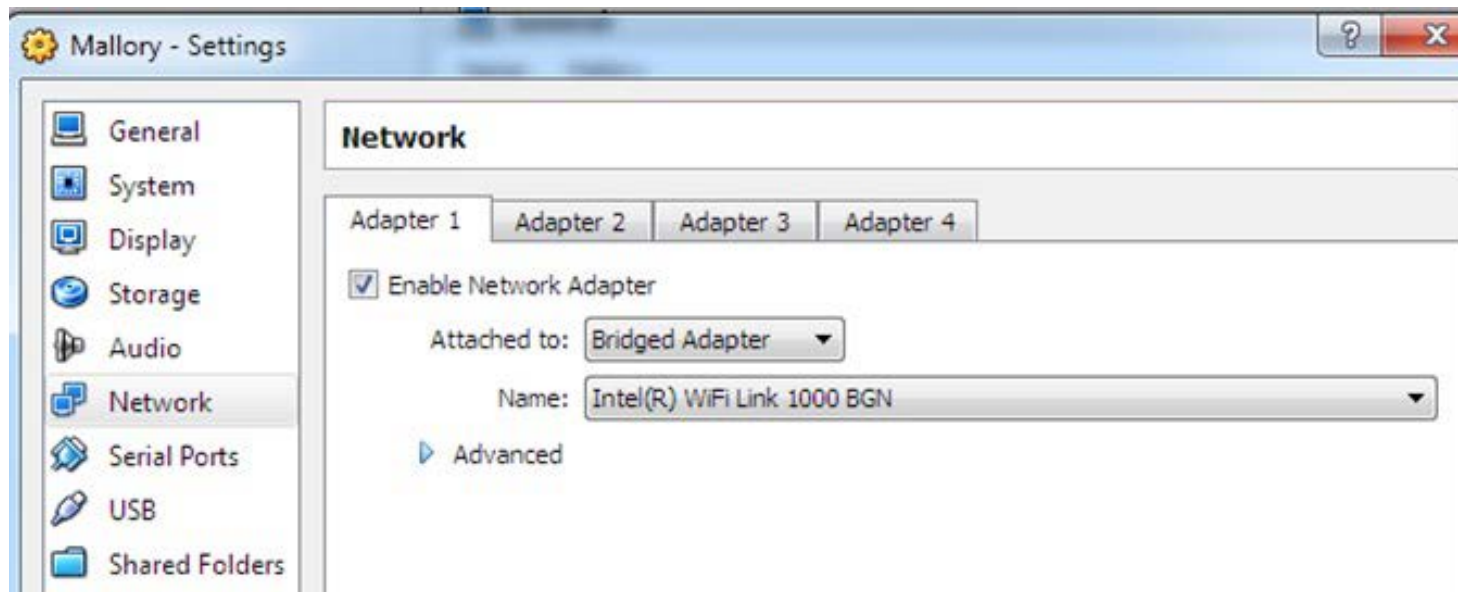
- ▶ Download executable file onto laptop from <https://www.virtualbox.org/wiki/Downloads>
- ▶ Run .exe file and install Oracle VirtualBox
- ▶ Download Ubuntu 11.04* from <http://old-releases.ubuntu.com/releases/natty/>
 - Either burn the image on a CD or download it on a USB stick.

**Copyright V. Moonsamy.*

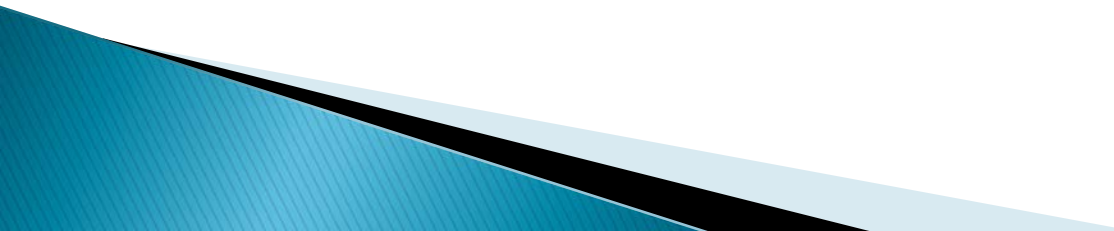


Step 1 – continued

- ▶ **Important:** Make sure that Network settings for Ubuntu* are as below:
- ▶ Back to Virtual Box, create a new VM and install Ubuntu.



SETTING UP MALLORY – cont'd

- ▶ **Step 2:** Install Mallory in VM.
 - ▶ **Step 3:** Configure Mallory – add DNS settings and usernames, and restart.
 - ▶ **Step 4:** Configure Smartphone on same network as the PPTP server (connected to a wireless access point on the LAN).
- 

The steps cont'd

- **Step 5:** Start Mallory by opening two consoles on the Ubuntu VM.
- **Step 6:** Configure the Mallory GUI.
- **Step 7:** Access the Internet on the smartphone over the VPN; Mallory will capture the traffic.

See the next slide:



Mallory - Transparent MITM Proxy

Mallory Help

Interfaces

Protocols

Rules

Streams

Advanced

0

c2s

819

192.168.0.235:49356

74.125.237.159:443

S

1

s2c

476

192.168.0.235:49356

74.125.237.159:443

S

2

c2s

773

192.168.0.235:49357

128.184.216.200:80

S

3

c2s

792

192.168.0.235:49358

74.125.237.129:80

S

4

s2c

367

192.168.0.235:49357

128.184.216.200:80

S

5

s2c

376

192.168.0.235:49358

74.125.237.129:80

S

6

c2s

1005

192.168.0.235:49359

128.184.216.200:80

S

7

s2c

327

192.168.0.235:49359

128.184.216.200:80

S

8

c2s

381

192.168.0.235:49360

199.27.75.130:80

S

9

s2c

1448

192.168.0.235:49360

199.27.75.130:80

S

10

s2c

8192

192.168.0.235:49360

199.27.75.130:80

S

11

s2c

8192

192.168.0.235:49360

199.27.75.130:80

S

12

s2c

8192

192.168.0.235:49360

199.27.75.130:80

S

13

s2c

8192

192.168.0.235:49360

199.27.75.130:80

S

14

s2c

8192

192.168.0.235:49360

199.27.75.130:80

S

15

s2c

8192

192.168.0.235:49360

199.27.75.130:80

S

Intercept

Auto Send

Send

Clear Streams

Text

Hex

Save Text Changes

```

GET
/urI?sa=t&source=web&cd=1&ved=0CDYQFjAA&url=http%3A%2F%2Fwww.deakin.edu.au%2F&ei=FMmGUc-2GK2zQeEvYCACA&usg=AFQjCNEK3BHNQvOoGLWuO3Jg6N0DKcBFmA HTTP/1.1
Host: www.google.com.au
Referer: https://www.google.com.au/search?q=deakin&ie=UTF-8&oe=UTF-8&hl=en&client=safari
Header: fake
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Cookie: PREF=ID=539ee388fc754097:U=ee03e08639dc9aba:FF=0:TM=1367024295:LM=1367024295:S=QBOjL6fc5qL7b6x3; NID=67=NVW7Oc7x0L7minT-vyI8eQmAMhLHCKWcPOkVx4LP-ey4OKdXvh4p4zzZthkAIBMUbeCmEGhA/vOC0UrxgrN6YA3mPDCucQfgAWPsT1GjojePPkqZcyMVTIA6TNp9fNj-E4itBhHojn55esn4W6
Accept-Language: en-us
Connection: keep-alive
User-Agent: Mozilla/5.0 (iPhone; CPU iPhone OS 6_1_3 like Mac OS X) AppleWebKit/536.26 (KHTML, like Gecko) Version/6.0 Mobile/10B329 Safari/8536.25

```

Mallory picking up a plaintext password

ations: Intercept Auto Send Send Clear Streams

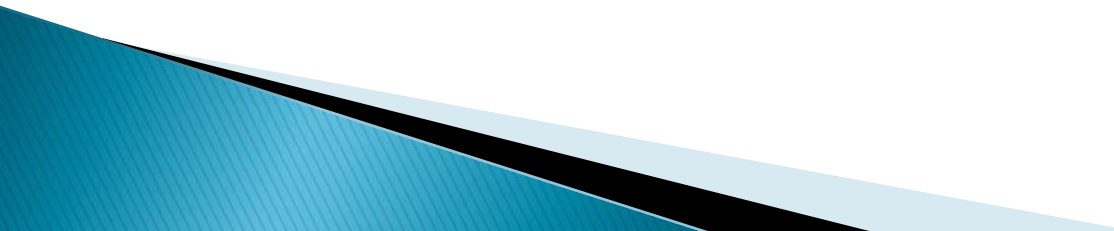
Text Hex

Save Text Changes

POST /ServiceLoginAuth HTTP/1.1
Accept-Encoding: gzip,deflate
Origin: https://accounts.google.com
Accept-Language: en-GB,en;q=0.8
User-Agent: Mozilla/5.0 (BB10; Touch) AppleWebKit/537.10+ (KHTML, like Gecko) Version/10.1.0.1720 Mobile Safari/537.10+
Content-type: application/x-www-form-urlencoded
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Referer: https://accounts.google.com/ServiceLogin?service=mail&passive=true&continue=http://mail.google.com/mail/?ui%3Dmobile%26zyp%3DI&sc=1<mpl=ecobxgm&nui=5&btmpl=mobile
Cookie: NID=67=IKluOCLYvQDU7f-II1YbC3dDQlhB--
CFTwe0R0o9MAwObVvY8DYSnUi3aQOWnjH4cVtHxw_NQj_UzPZTPjysp4aYGzis1dgVGkXNgEOGNixE7rjdzhk5Br9rLSje3igvN;__utma=72592003.234945783.1373463266.1373463266.1373463266.1373463266.1;__utmz=72592003.1373463266.1.1.utmcsr=(direct)|utmccn=(direct)|utmcmd=(none); GAPS=1:-LqM0ZSXlHv23xjJXOLWJC4EuEx2FA:rBFyGu7ElZNwzGjG; GALX=FLvI70EjhG0; GMAIL_LOGIN=T1373500230396/1373500230396/1373500252528
Host: accounts.google.com
Connection: keep-alive
Content-Length: 648

continue=http%3A%2F%2Fmail.google.com%2Fmail%2F%3Fui%3Dmobile%26zyp%3DI&service=mail&nui=5&dsh=-2795779443633958655<mpl=ecobxgm&btmpl=mobile&sc=1&GALX=FLvI70EjhG0&timeStmp=&secTok=&_utf8=%E2%98%83&bgrresponse=%21A0IH_smkRoMjv0Rqa5OL12ixkgIAAAEHUgAAADUqAO7cua7BGERsENoSpxCknWTKPdErnAtG5V8aUkrJGLGxFFWR6k90NxZNau5c_h_lb44Q5oFw6ljmeVxR2HO6Sajix4TffaPC--esq4_3g26qvqnkcUrfoPkEeRg_Vok4dOKTwOaKUXHaZEd3NMkH8F_T8eTbUyX-8E_Jm7qZNQ39jn24eXWpCTaMx_9dva74jd5NPAvDAfjS2i2ms-kfIAi4aaEJqYvhLCPM6lyoMyv7ja_kYaWIOgXogq2n-aKbIFluaWPI659J5rFMLyaY0Q7048KA5mj1_yuioZAbQ9YmCSQruF-thUhPi-a4SNXS&Email=rahul&Passwd=rahul123&signIn=Sign+in&PersistentCookie=yes&rmShown=1

SOME OF THE CHALLENGES WE FACED IN SETTING UP MALLORY

- ▶ Unstable wifi connections;
 - ▶ Different OS use different terminology and require different settings:
 - With Android and iOS we used a VPN connection,
 - VPN was not available with Blackberry and Windows – a wifi adapter was used instead.
- 

UPPER LEVEL SUGGESTIONS FOR MITIGATING THESE PROBLEMS

- ▶ Re advertising
 - give the user the option of denying sending data to third parties.
 - make developers adhere to strict rules about certificate use.
 - Companies such as <http://www.geoedge.com> offer to ensure that your ‘mobile ads’ are ‘clean and safe’, by checking for
 - Malware
 - Malicious Code Activities
 - Redirects

Enhancement of certificate trust models

See the research literature including:

- ▶ J. Clark and P.C.van Oorschot (2012). 'SoK:SSL and HTTPS'. Proceedings of the IEEE Symposium on Security and Privacy.
- ▶ Yasodharan, R., R. Sivabalakrishnan, and P. Devendran. 2015 "Trusted Routing with an Efficient Certificate Revocation for Mobile Ad Hoc Network." IJSET
- ▶ Mall, Tarun, and Samarth Gupta. 2014 "Critical Evaluation of Security Framework in Android Applications: Android-level security and Application-level security." Journal of Computers and Electronics Engineering.
- ▶ Vallina-Rodriguez, Narseo, et al. 2014 "A Tangled Mass: The Android Root Certificate Stores." *Proceedings of the 10th ACM International on Conference on emerging Networking Experiments and Technologies*. ACM.

Measures that users can apply

User understanding and user behavior are key aspects that can mitigate the propagation of rogue applications. This includes:

- ▶ User comprehension of OS security
- ▶ User carries out a thorough background check before downloading an APP
 - Read reviews, ratings, number of downloads, ...
- ▶ Modify Phone Settings to turn off targeted advertising
 - Does not stop ads, only prevents APP from using unique device IDs as targets

(Explained at <http://dottech.org/21999/android-tip-turn-off-interest-based-ads-by-flipping-a-switch-in-android-market/>)

Bibliography

- ▶ 1. Luo et al. 2011 'Attacks on WebView in the Android system. In *Proceedings of the 27th Annual Computer Security Applications Conference* (pp. 343–352). ACM.
- ▶ 2. Luo, T. (2014). Attacks and countermeasures for WebView on mobile systems. PhD dissertation, Syracuse University, May 2014
- ▶ 3. Alazab, M., Moonsamy, V., Batten, L., Tian, R. & Lantz, P. (2012). Analysis of Malicious and Benign Android Applications. In proceedings of International Conference Distributed Computing Systems (ICDCS 2012). Macau, China.
- ▶ 4. Enck, H. Analysis Techniques for Mobile Operating System Security, PhD thesis, Pennsylvania State University, 2011.
- ▶ 5. Enck, W., Gilbert, P., Chun, B. G., Cox, L. P., Jung, J., McDaniel, P. & Sheth, A. N. (2010). TaintDroid : An information-flow tracking system for realtime privacy monitoring on smartphones. *Proceedings of the 9th USENIX conference on Operating systems design and implementation*.
- ▶ 6. Fahl, S., Harbach, M., Muders, T., Baumgärtner, L., Freisleben, B., & Smith, M. (2012). Why Eve and Mallory love Android: An analysis of Android SSL (in) security. In *Proceedings of the 2012 ACM conference on Computer and communications security*. ACM.
- ▶ 7. Moonsamy, V.; Alazab, M. & Batten, L. (2012), Towards an Understanding of the Impact of Advertising on Data Leaks' In *International Journal of Security and Networks* (IJSN). Vol. 7, 181–193.
- ▶ 8. Moonsamy, V. & Batten, L. (2012), Zero Permission Android Applications: Attacks and Defences, In Proceedings of the 3rd Workshop on Applications and Techniques in Information Security (ATIS), Melbourne, Australia, November 2012.
- ▶ 9. Moonsamy, V. and Batten, L. (2014), 'Mitigating Man-in-The-Middle Attacks on Smartphones – a Discussion of SSL Pinning and DNSSec'. In proceedings of The Australian Information Security Management Conference, Perth, Edith Cowan University, pp. 5–13.
- ▶ 10. Rahulamathavan, Yogachandran, Veelasha Moonsamy, Lynn Batten, Su Shunliang, and Muttukrishnan Rajarajan. "An Analysis of Tracking Settings in Blackberry 10 and Windows Phone 8 Smartphones." In *Information Security and Privacy*, pp. 430–437. Springer International Publishing, 2014.
- ▶ 11. Pearce, P., Felt, A. P., Nunez, G. & Wagner, D. (2012). AdDroid: Privilege Separation for Applications and Advertisers in Android. *Proceedings of AsiaCCS*.

Thank
you