Mobile Threats and the Underground Marketplace

Principal Investigator and Corresponding Author

Jart Armin

Contributing Researchers

Andrey Komarov, Mila Parkour, Raoul Chiesa, Bryn Thompson, Will Rogofsky

Panel & Review

Peter Cassidy (APWG), Dr. Ray Genoe (UCD), Robert McArdle (Trend Micro), Edgardo Montes de Oca (Montimage), Dave Piscitello (ICANN), Foy Shiver (APWG)

APWG Mobile Fraud web site - http://apwg.org/resources/mobile

Table of Contents

Published May 8th, 2013
ISBN # 978-0-9836249-9-8

Abstract

A rapidly advancing mobile market and a corresponding decline in PC sales, sees 2013 at a crucial intersection. Termed in a market trend as the "post-PC" era, mobile devices increasingly present an attractive, practical and economical alternative. In the next few years global mobile payments are predicted to exceed $1.3tn.

While there is already an established mobile malware market, now is the time to take stock, to demonstrate the existence of such an industry and how it operates through stealthy intrusion and crime ware supply chains.

This paper defines these malware markets and demonstrates the modus operandi of an industry that is self-funding, prosperous, vertically stratified and agile.

Types of malware and attack methods under analysis include: spyware, phishing direct attacks, Trojans, worms, apps delivered through malware, pocket botnets and blended attacks, many of which are designed to steal or pilfer money from users. Equally as invasive can be "track and trace" intrusion techniques used to extract intelligence about an owner's usage and habits.

This paper will provide a rhetorical approach towards mobile crime ware and the intrusion supply chain's structure as it examines subjects in depth from a practitioner's perspective.

Introduction and Starting Position

- By 2015 it is estimated there will be 2 billion + mobile devices.
- China as an example now has 564 million Internet users: 75% are mobile.
- Global mobile payments are predicted to exceed $1.3tn[1].
- Virustotal currently shows 5.6 million reported potentially malicious files for Android (APK, dyn-calls, checks-GPS, etc.) of which 1.3 million are confirmed malicious by 2 or more AV vendors.
- While there is already an established mobile malware market, now is the time to take stock, to demonstrate the existence of such an industry and how it operates through stealthy intrusion and crime ware supply chains.
- Types of malware and attack methods under analysis include: spyware, phishing direct attacks, Trojans, worms, apps delivered through malware, pocket botnets and blended attacks, many of which are designed to steal or pilfer money from users.
- Equally as invasive can be "track and trace" intrusion techniques used to extract intelligence about an owner's usage and habits.

[1] http://www.juniperresearch.com/viewpressrelease.php?pr=332

A Global Overview

Figure 1: Mapping the World – Countries currently at high risk from Mobile Threats

Vulnerabilities Overview

Types of vulnerabilities

Architecture

Infrastructure

Hardware vulnerabilities

Permission systems

Software vulnerabilities

Communication/delivery channels (Wi-Fi, SMS, Bluetooth)

Near Field Communication

PtH (Passing the Hash)

Architecture

Open and programmable smartphones are transforming mobile communications. Powerful sensors, capable of interacting with a growing number of mediums, drive the change from multiple pieces of equipment to one convenient pocket-sized device. With valuable data centralized, malicious attackers aim for the weakest link – the infrastructure architecture.

One highly customizable device has many obvious advantages. The capability of sensors in gathering an array of information into one small location may, however, come at a price. Data fuels an ever expanding industry where, sadly, not all the players are scrupulous and trustworthy. There are genuine concerns over privacy; the majority of users find the selection of privacy settings and application permissions challenging, if not confusing. Of concern, too, is the availability, and granularity, of modifiable permissions. And where does that leave the important issue of security when most users fail to even secure their devices with a password or PIN? These are not encouraging signs when, increasingly, work and leisure use of smartphones is overlapping.

The new generation of powerful sensors has added the "smart" to smart-phone. Through GPS, accelerometers, gyroscopes, magnetometers, proximity sensors, microphones, cameras and radio (cellular, Bluetooth, Wi-Fi, RFID, NFC) we can interact with domains such as social networks, entertainment, education, transportation, gaming and mobile banking. "Smart" interaction continues to grow and to bring rich enhancements into our lives. Sadly, despite these technological advances attacks against smartphones are on the increase too, so have "smart" defense been forgotten?

Infrastructure vulnerabilities (handsets)

Malicious attackers seek out the weakest targets. In the case of smartphones attackers are quick to exploit inherent infrastructure vulnerabilities.

Attackers will choose the attack mode depending on the target. However, some basic features are strikingly similar across all operating systems. Devices may vary on design, functionality or network stack Android, iOS, Symbian OS, Microsoft Window Mobile and Palm OS, all offer:

- Access or support of a mobile network.
- Access to the Internet through interfaces such as Bluetooth, WLAN, infrared or GPRS,
- TCP/IP protocol stack.
- Desktop PC synchronization.
- The ability to simultaneously run multiple applications.
- Open Application Programming Interface (APIs) to develop the applications.

In effect this means that targets can be condensed into four main categories: hardware, software, user, communication/delivery channels.

Basically, targets are the same whether they are via a desktop computer, laptop or mobile device – the difference lays in the associated risk.

It could be argued, for example, that smartphone users should bear a greater responsibility for the safety of a device they carry around, since the likelihood is that at some time or other a device will be lost, stolen or forgotten. This presents an additional risk factor over static equipment, such as PCs or laptops/tablets, when size makes them obtrusive or more obvious when not present. But is this just diverting from the real problem? The solution to data being compromised, hijacked or stolen should, essentially, be at the point where the data is stored, shared or inherent in the data itself.

Hardware vulnerabilities

Screen size

Phishing is advanced through the constraints of a small screen. URLs can be stealthily hidden or disguised to greater effect than on a desktop. Typosquatting, too, may pose an additional risk as misspelling, or hitting a wrong letter, is an easy mistake to make on a small touch input keyboard.

Keyboards

Just as for the screen size, the keyboard's usability on mobile devices represent a serious facilitator for the success of frauds and attacks towards mobile users.

The BYOD paradigm here is enhanced, in that mobile users often choose "easy" (weak) passwords for accounts that they use mainly via mobile devices. It is especially true that with touchscreen devices and virtual keywords, users tend to rely on short and non-complex passwords (i.e., no special character, as the user needs to activate a different keyboard layer in order to type it), and this makes it easier to type a password while on the move.

On the other hand, "real" keyboards, i.e., some Nokia or BlackBerry devices, do not allow much of a "typing area" which results in constraints when logging into an account.

3G & always-on

3G and 'always on' presents hackers with new, stealthy, methods of exfiltrating the data. In the past stolen data was accomplished mainly via outgoing SMS or by taking advantage of the phone's built-in modem. With the availability of IP-based 3G services the bad guys can take advantage of IP data traffic flat contracts with no extra cost passed on to the victim.

Flat rate contracts can mask the fact that data is being stolen when, in the past, the user may have discovered this upon receipt of the phone bill, or due to a sudden decrease in phone credit on a prepaid SIM card. This usually determined that mobile fraud, via texts to PRN numbers (dialers) in particular, had a short-life span.

Kernel

The kernel acts as a bridge between the hardware and software programs. It is a high risk target for any attack where control of any of the central functions is required.

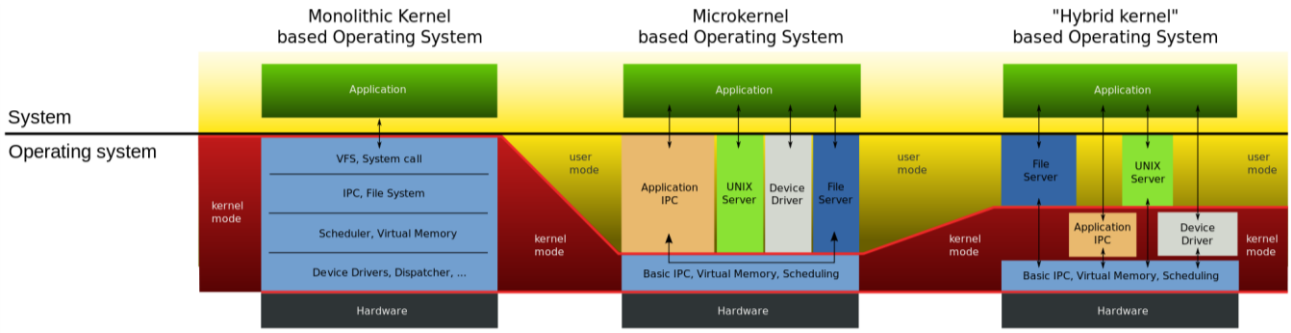

Figure 2: Monolithic & Microkernel structures

The Android kernel is especially vulnerable to specific attacks on its infrastructure architecture. Android operates Linux using a single structured operating system known as a monolithic architecture[2]. This architecture prevents the isolation of internal processes and increases the risk of exploitation once the Android kernel has been compromised. If a bug is present in any of the subsystems, it can be exploited to bypass the security processes and to control **all** of the permissions. Kernel code validation, testing and updates are of vital importance to Android.

The number of lines of code in Linux-based kernels provides ammunition, for competitors of the Android OS, when compared to, for example, microkernel-based systems[3]. Using an average of one exploitable bug per line of code, a much quoted but little quantified figure, the assumption is that more code equates to more bugs. This is a very simplistic view of an area with many variables. A research study at Purdue University in 2010 looked at variables including bug fixes, code complexity and reliability and found that bug density in both Android and Symbian 'were surprisingly low'[4]. However, the customizability of Android "*…comes at a cost for a significant fraction of bugs—between 11% and 50%...*" Quality management, therefore, is an issue that needs careful execution.

Kernel integrity is an essential of basic security. Android rooting or jail breaking the iOS system in order to customize smartphone features can seriously compromise this integrity, especially if attempted without proper knowledge of how the system operates. The first worm for jailbroken iPhones, 'Ikee', demonstrated this in November 2009. 'Ikee' successfully replaced the wallpaper of infected iPhones with a photo of a

[2] http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6335029

[3] http://www.techradar.com/news/phone-and-communications/mobile-phones/how-blackberry-10-avoids-android-s-security-issues-1103381

[4] https://engineering.purdue.edu/dcsl/publications/papers/2010/android_issre10_submit.pdf

1980's pop singer and, at the same time, scanned the network for other vulnerable phones to infect[5].

The vulnerability of jailbroken/rooted devices poses a significant threat to the management of BYOD (Bring Your Own Device) in the workplace. Jailbreaking is popular with users. Within a few days of the release of the first untethered jailbreak for iPhone 5 and devices running iOS 6.x, installs reached 7 million[6].

Codeproof Mobile Security detected approximately 11.19% of jailbroken mobile devices worldwide[7]. The situation in China is worse. The percentage of jailbroken handsets dropped to 27.3% but increased by 5% when the iOS 6.1 jailbreak was announced[8].

Chinese users are particularly vulnerable to attack which is compounded by Chinese reluctance to upgrade compared to 'their overseas counterparts'.

The legality of jailbreaking remains a complex issue with no uniform law enforceable worldwide. In the United States, legislation under the remit of the Digital Millennium Copyright Act (DMCA), makes it illegal to unlock any new phone bought after January 23 2013. Jailbreaking or rooting remains legal until at least 2015[9].

Permission Systems

Android uses permission systems to implement mandatory access control (MAC). When an application is installed it requests permission to access system resources such as location, Internet, or the cellular network, from the user. Many users are unaware of the implications of granting such permissions and install regardless of the level of access requested. Permission to access the Internet, for example, allows unrestricted communication with any server. Permission to access areas where personal data is held leaves a user's contacts and text messaging open to abuse.

Software vulnerabilities

Delayed software updates

Delays in software updates can leave vulnerabilities open to exploitation. The large code base of monolithic kernels, such as Android, renders the system especially open to attack when a known vulnerability is slow to be patched - lagging on an update may be the only flaw that an attacker needs in order to compromise the whole device. In comparison microkernel-based OSes can be easier to manage due to its small size and relatively 'clean' performance.

---

[5] http://www.symantec.com/connect/blogs/ikee-worm-rickrolls-jailbroken-iphones

[6] http://www.forbes.com/sites/andygreenberg/2013/02/08/evasi0n-is-the-most-popular-jailbreak-ever-nearly-seven-million-ios-devices-hacked-in-four-days/

[7] https://www.codeproof.com/PressRelease/Jailbroken_phones_as_of_Jan_02_2013

[8] http://www.slideshare.net/umengnews

[9] https://www.eff.org/is-it-illegal-to-unlock-a-phone

Also, mobile users, who have little knowledge of data security, do not patch their devices at once. This is especially true if travelling: there are plenty of cases where the user will not launch the update procedure due to the long time needed, or in case the phone discharges. These users then connect the phone to the power plug and are then unable to properly use the phone until returning home to carry out the update. This can result in a delay in applying the patch of days, or even weeks.

Applications

Making and deploying a malicious app does not require a high level of skill and enticing users to download malicious software is relatively straightforward. Many users are drawn to offers of free items, games or attractive pictures which they will install even if unsure of the source.

Smart technology is revolutionizing the way we control things. Apps have the capacity to deliver a fully automated living environment with appliances controlled from outside the home. Anything from heating, lighting, alarm systems, smart TVs, entertainment systems, even refrigerators, to management of financial data can be instructed, or manipulated, from apps on one mobile device.

Trust in app stores may be misplaced in some cases. While some level of security is in place to check for malware, bugs or exploit code may escape cursory checks. Once installed, exploit code can attack a vulnerability to access passwords, personal data, bank account numbers, text messages, etc., or exploit the camera/microphone for use as a spying device.

Custom device interfaces and features

Device manufacturers offer enhanced features to gain competitive advantage. This may require an element of source code modification and adds to the potential for vulnerabilities or flaws. In addition, the device vendor may need to port the custom user interfaces and features when updates are released, adding time and cost to a device that is already sold.

The device manufacturer has little to gain from thorough quality management of interfaces and may cache updates instead of promptly addressing known vulnerabilities. At any one time this could amount to millions of devices with unpatched vulnerabilities.

Version	Codename	API	Distribution
1.6	Donut	4	0.2%
2.1	Éclair	7	2.2%
2.2	Froyo	8	8.1%
2.3 - 2.32	Gingerbread	9	0.2%
2.3.3 - 2.3.7		10	45.4%
3.1	Honeycomb	12	0.3%
3.2		13	1.0%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	29.0%
4.1	Jelly Bean	16	12.2%
4.2		17	1.4%

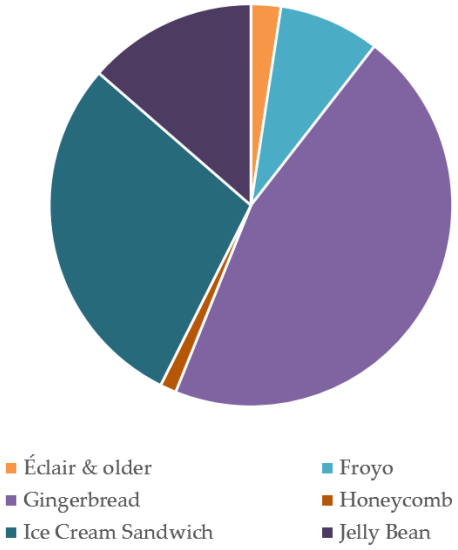

Figure 3: Android versions and distribution

Figures from Google for Android distribution illustrate the problem, as old versions of the operating systems continue to present fragmentation as a serious problem[10].

Improved security is rendered relatively useless if old systems are still widely in use.

Users (as a Vulnerability)

Users are often maligned and even labelled as irresponsible for their apparent lack of care over smartphone management. Low take-up on password or PIN application only adds fuel to the fire

While users may bear some of the responsibility for a portable device that can easily be lost or forgotten, is it fair to expect more than this?

Manufacturers, and vendors, want to sell their products and provide little guidance other than basic start-up procedures. Devices are easy to buy and to operate and come without instruction on how to stay safe other than a cursory "install only from trusted sources". As detailed above – trusted sources may not be a trusted as we should like.

Likewise, bills based on usage provide few incentives to carriers when they stand to benefit from apps that use, misuse or exfiltrate data.

Users may have concerns about privacy of sensitive information but few understand what this means in terms of giving permission to access certain types of data. Knowing what connects to what and why is not always straightforward. For example, what does "third party servers" mean in a device's handbook? Should users be told what their phone identifiers – phone number, IMEI, IMSI, or ICC-ID – connects to or tracks?

[10] http://bgr.com/2013/02/08/android-version-distribution-february-2013-316698/

General advice on the dangers posed by phishing or "smishing" is available through the support or community sections of the operators' websites but advice specific to mobiles is hard to find.

The size of the screen is exploitable; users may not be able to spot when a URL leads to a fake site or to recognize when a shortened link sends them to a destination that mimics a well-known brand.

QR codes offer new ways for fraudsters to trick mobile users. Malware can be hidden within an app, if it is not from a trusted source, or can be embedded within the code.

Operators can help alleviate some of these problems with a more proactive approach towards providing educational material for users and by making it easier to obtain information on the latest threats.

Communication/Delivery Channel Vulnerabilities

Smartphones connect to and deliver a growing number of communication channels. The advent of 4G will increase the capability and introduce new platforms for cybercriminals to exploit.

Wi-Fi

Open Wi-Fi networks are used extensively by hotels, cafes, bars, buses, trains, airports, etc., but more recently stores are enticing shoppers with an additional range of services, or 'personal concierges', available directly on the mobile, while in-store. Services can range from product details, to deals and offers, which aim to enhance the 'shopping experience', and to gain more sales. JiWire Mobile Audience Reports for Q2 2012 found that 93.6% of smartphone owners used their mobile device while in-store[11]. Further, the availability of in-store Wi-Fi influenced where the majority would shop. Abuse of privacy is not the only issue; malicious intruders could access to the network, hijacking the system or snoop on individuals.

As part of the firmware Wi-Fi chips can be vulnerable to attack from bugs in the coding. An example of this type of vulnerability was disclosed by 'Core Security' in Oct 2012 with the issue of an advisory detailing how the Wi-Fi NIC could be prevented from responding[12]. A patch was subsequently released by Broadcom.

[11] http://www.jiwire.com/sites/default/files/JiWire_Insights_Q4_2012.pdf

[12] http://www.coresecurity.com/content/broadcom-input-validation-BCM4325-BCM4329

SMS

Successful exploitation of SMS can be carried out through spoofing or hijacking which has led to large financial losses in many countries, and the regulations in specific countries' telecoms may unintentionally facilitate this[13]. Unlike other features SMS cannot be turned off. Attackers find the singular messaging communication process used by Android easier to abuse than the log-driver mechanism used by some other operating systems such as Windows.

Example

France - Oct 2012 - 20-year-old hacker using fake Android apps, established a virus on 17,000 users' smartphones to send premium rate SMS messages. $650,000 (€ 500,000) within 8 months.

http://www.frandroid.com/actualites-generales/117583_six-mois-de-prison-ferme-pour-notre-hacker-national-damiens/

Malicious programs, known as man-in-the-middle, hijack SMS containing mTANs (Mobile Transaction Numbers) codes used by financial institutions around the world to authenticate online banking transactions. MTANS were considered to be safe from attack until hackers modified the Zeus and SpyEye (SPITMO, MITMO) Trojans, snatched mTAN codes from Android mobiles to steal millions[14].

Bluetooth

The first recognized virus that propagated via an open Bluetooth was discovered in 2004[15]. The virus, known as Cabir, exploited a programming vulnerability that enabled the infected device to connect to other Bluetooth devices in the same vicinity.

Bluetooth vulnerabilities continue to be discovered and fixes subsequently issued. Users frequently leave Bluetooth enabled, with many seemingly unaware of the increased risk from attack that this poses. Permission selection and incorrect implementation of Bluetooth protocols, leave devices vulnerable and insecure.

'Bluesnarfing' is unauthorized access from a wireless connection through a Bluetooth connection[16]. There are several tools on the market to safeguard against this type of attack.

Bluetooth is also used for spamming[17] in crowded places for all nearby discoverable/visible BT devices through using the OBEX protocol, more specific OOP (Obex Object Push) and/or OBEX-FTP (OBEX File Transfer Protocol), such technique

---

[13] http://tamsppc.tamoggemon.com/2007/10/05/public-service-announcement-sms-scam-running-rampage-in-austria/

[14] http://blogs.mcafee.com/mcafee-labs/spitmo-vs-zitmo-banking-trojans-target-android

[15] http://www.theregister.co.uk/2004/06/15/symbian_virus/

[16] http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6269870

[17] Bluetooth Spammer https://play.google.com/store/apps/details?id=com.smartmadsoft.bluetoothspammer

is called «BlueSpam». It is also knows that some companies[18] use it for marketing purposes as well as with some phishing schemes.

Near Field Communication (NFC)

The benefits of NFC technology used for contactless payments, file transfers, the sharing of information via social networking and Wi-Fi enabling tags, etc., are sure to increasingly unfold as the standard continue to evolve. Sadly, malicious attackers will also take up the challenge to find ways in which to circumvent the technology. Some NFC vulnerabilities are already detailed. For example, ATMs enabled with NFC technology can have card skimmers installed, tags can be infected with malicious apps, and ID cards can be cloned. The NFC stack is the target of one type of attack that 'fuzzes' selective layers including the protocol and application. A recent analysis showed that an attacker in close proximity can parse one of over 20 different formats without any interaction by the user[19].

PtH (Pass-the-Hash)

A technique that bypasses the need to crack or guess a password to gain unauthorized access to an account, network or within the emerging threat landscape of the mobile environment.

Passwords are stored in environments that can be compromised, for example, in plaintext, reversible encryption or in a hash format. An attacker can remove the need for time-consuming password once he holds the password hash. (This technique is described in greater detail in the SANS Institute InfoSec Reading Room paper[20].)

The existence of Pass-the-Hash has been known about for more than fifteen years but still remains relatively unknown among the wider security industry. PTH is often employed as part of Advanced Persistent Threats (APT) and used for industrial espionage. Once inside a system an intruder may lay undetected for long periods of time.

---

[18] http://www.blueblitz.com/

[19] http://media.blackhat.com/bh-us-12/Briefings/C_Miller/BH_US_12_Miller_NFC_attack_surface_WP.pdf

[20] http://www.sans.org/reading_room/whitepapers/testing/pass-the-hash-attacks-tools-mitigation_33283

The Underground Mobile Market

Pricing

Sample Toolkits & Service	Price (US$) - March 2013	Example Descriptions
Mobile intrusion (keyloggers)	Open Source - 400	Java & Python Keyloggers, Mobistealth,
Mobile Intrusion (surveillance)	500 – 5,000	Re-engineered Finfisher, Finfisher Lite & FlexiSpy extended copies
Mobile malware for banking theft	10,000 – 30,000	Eurograbber, ZitMo, Tinba Trojan, DroidCleaner, Citadel (inc. PtH capabilities)
Mobile botnet (rental)	50 - 400	Hourly rates
Mobile botnets (operational & tailored source code)	4,000 - 30,000	Mobile ISP service, SMS, & Drive by
Mobile malware for black SEO and underground partnership programs	5,000 – 10,000	Used to traffic redirects, J2ME midlets, or standard applications for the popular platforms.
Mobile traffic by targeted country	10 – 30 per 1,000 hosts	Can be bought through special underground services (by area, by country)
Mobile SMS spam service	2-8 cents per 1 SMS	Mobile spamming
Mobile SMS spamming tool	30-50	SMS spamer by klychev v0.3
Mobile flooder (Skype or SIP)	30-80	Skype Flooder

Table 1: eCrime market current pricing (March 2013)

Market status

The mobile malware black market remains a work-in-progress as it continues to develop and evolve according to demand. Cybercriminals skilfully manipulate the market and maximize the potential from advertising on underground forums and social networking sites.

The most popular mobile malware available on the market take advantage of a number of key properties, for example:

- Brands with well-known graphical designs, famous applications or legal entities such as financial institutions, e-commerce, stock/e-trading applications, applications for social networking, etc.;
- SMS or phone calls to other numbers (sometimes with a silent install of mobile malware on the system);
- Mobile banking applications that store customers credentials in insecure and plaintext form;

- "Jailbroken" devices,
- Non-verification of application source,
- Wireless channels (NFC, WPAN networks).

The demand for surveillance tools on the underground market is high and may, or may not, include malware. Essentially, the techniques used are similar on all devices or computers. The key targets of cyber espionage are:

- Information stealing including contact lists, text messages, calls history, information about compromised device (IMEI, Device IDs, external IP and etc.)[21];
- Call recordings including Skype conversations.

With the addition of special tools used traditionally for penetration testing, modern smartphones can become an aid in cyber espionage[22]. The mobile environment has provided new opportunities for the hacker.

Underground partnership programs enable cybercriminals to monetize through mobile malware. SMS plays a large part in these operations and can be distributed across the range of popular OSes. Unsolicited SMS or calls to expensive numbers in different countries run up large bills to the credit of the billing provider.

Some underground cybercrime services take advantage of mobile traffic using targeted attacks to download software which is charged on the basis of "Pay Per Install" (PPI)[23].

On the other side, the 0days underground market is rising up too, offering fresh and unknown exploited vulnerabilities (especially in the Android environment). These often work only on specific brands (i.e. Samsung, Huawei), and handsets models, which will then be "weaponized" and used massively to distribute mobile malware campaigns.

[21] Some of malware used for cyber espionage also intercept GPS coordinates, acoustic snapshots to monitor the person and different situations

[22] AFE (Android Framework for Exploitation)

[23] See the *Cybercrime Supplement* for further details and examples.

Mobile DNS & Traffic

It is important to monitor wireless users within a corporate network and check for signs of "jailbroken" devices. There are Mobile Device Management systems designed specifically for this purpose.[24].

Testing network traffic for UDID transmission is a useful aid within the mobile penetration testing environment. UDID's can be used to directly identify iPhone owners and to collate personal data on an individual, for example geo-location data, which may be passed to third parties without the consent of the user. If UDID's are present in network traffic, it may indicate the presence of malicious activity.

The UDID (Unique Device Identifier) of an iPhone can be located using this formula: UDID = SHA1(Serial Number + ECID + LOWERCASE (WiFi Address) + LOWERCASE(Bluetooth Address).

> *Note: The ECID (Exclusive Chip Identifier) is used mainly for beta testing firmware and apps but an app developer can add in protection in a cracked version to block the UCID and replace it with a fake/spoofed UCID.*
>
> *The SHA1 (Secure Hash Algorithm) is the most widely used of the NIST approved hash algorithms.*

In some cases the device transfers special timing data to the network. This can be used during forensic analysis to indicate the presence of intruders:

```
%  wget –user-agent="HTMLGET 1.0"
92.61.38.16/xml.p.php?id=1234502:       --HH:MM:SS–
http://92.61.38.16/xml.p.php?id=1234503: => p.php@id=12345'
```

[24] http://www.f5.com/products/mobile-app-manager/overview/

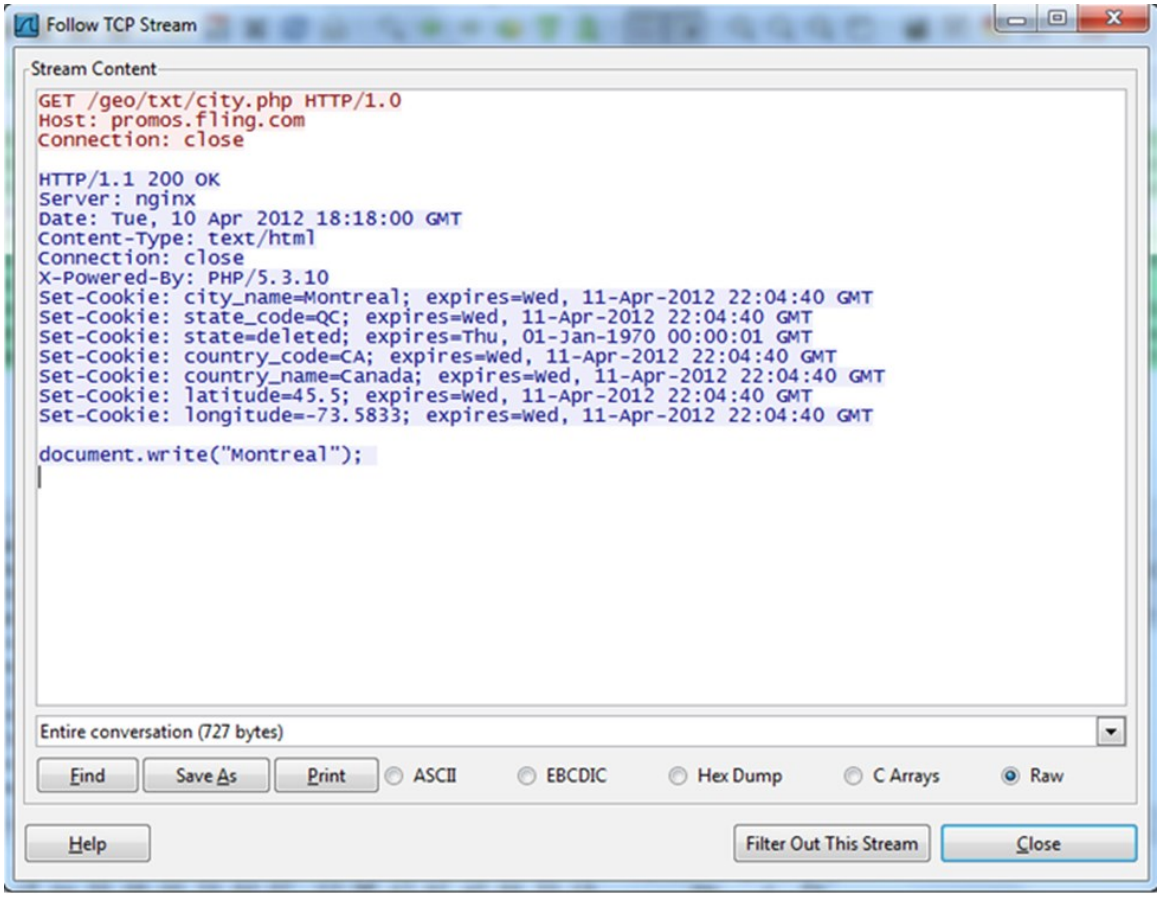

Figure 4: Valuable personal data being transferred via the network

This is useful information in regard to active C&C fingerprinting on the mobile device.

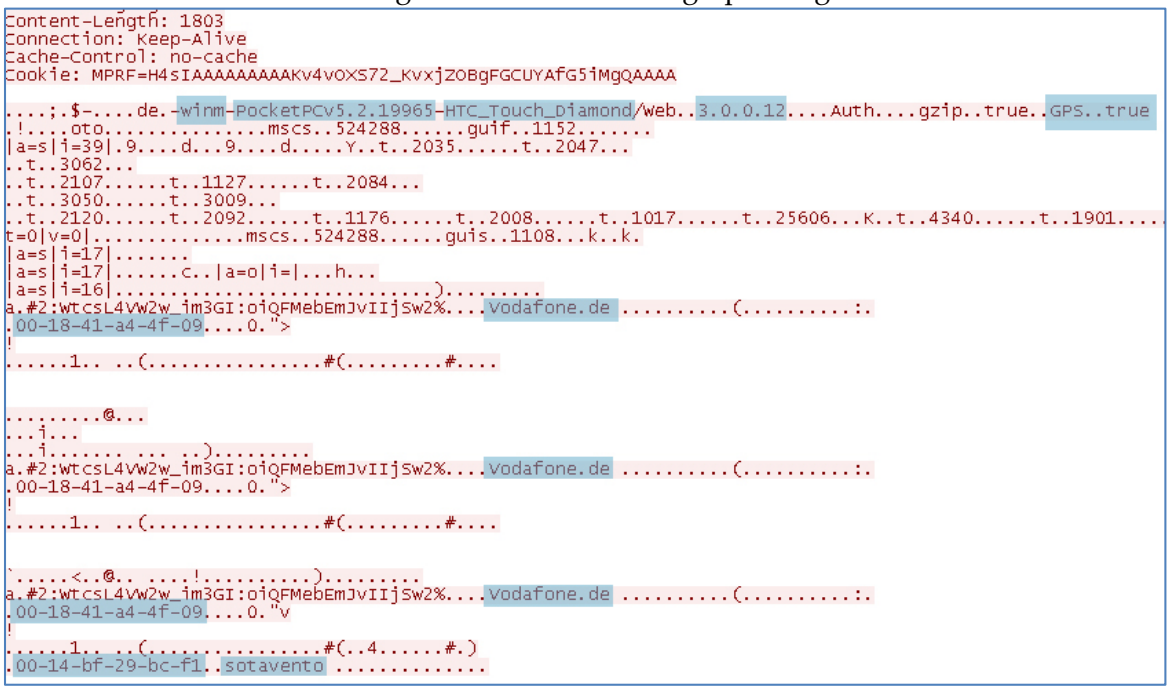

Figure 5: The device communicates valuable data

In this example specific MTAN data, used for online-banking transfer validation, is transfered along with other criteria which is targeted by banking mobile malware:

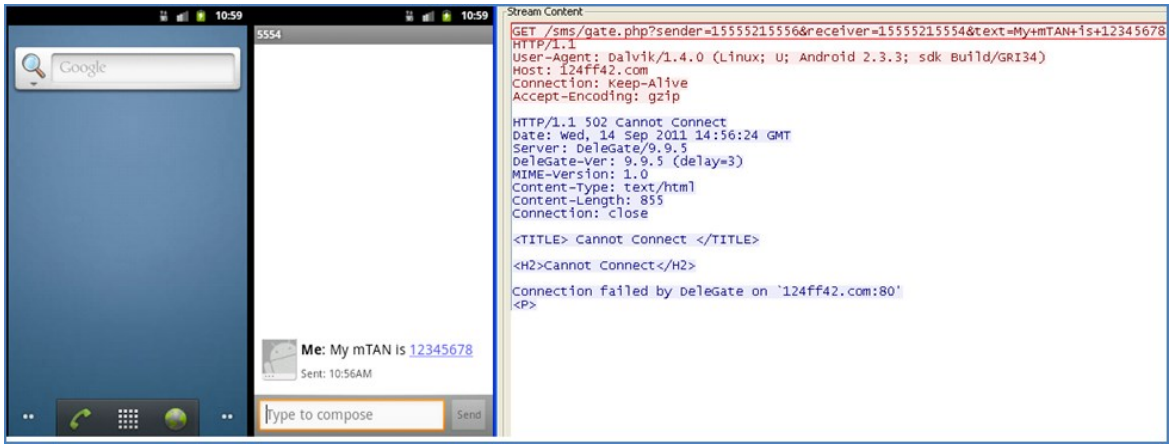

Figure 6: An attack targets MTAN data sent vis SMS

Each iPhone application has its own unique identifier stored in the */var/mobile/applications* directory and it is from here that all the code for the application executes.

The *Plist* (Property List) directory stores user preferences and configuration information. All communications between a botnet and its C&C are usually organized with help of the preference file '*com.apple.period.plist*', and the '*syslog shell scripts*'. The 'iKeeB' iPhone botnet, for example, installs the preference '*plist*' files, and has the ability to archive all SMS messages[25].

The bot probes every few seconds and regularly polls back to the C&C, every 5 minutes with 'iKeeB', which allows the addition of new scripts as the bot evolves and spreads.

The detection of suspicious external behavior, such as described here, can become a useful tool within the mobile penetration testing environment and organizations should be proactive in their approach to BYOD management. The monitoring of mobile network traffic can play an important role and more needs to be done to raise awareness of its value as BYOD evolves within the workplace.

[25] http://mtc.sri.com/iPhone/

iBots & the Pocket Botnet

In a relatively short period of time, mobile botnets have passed from theory into reality. It is not only possible for botnets to control mobile devices, but mobile botnets are becoming increasingly sophisticated.

The methodology and possible architecture is quite similar to classical botnet malware for PCs.

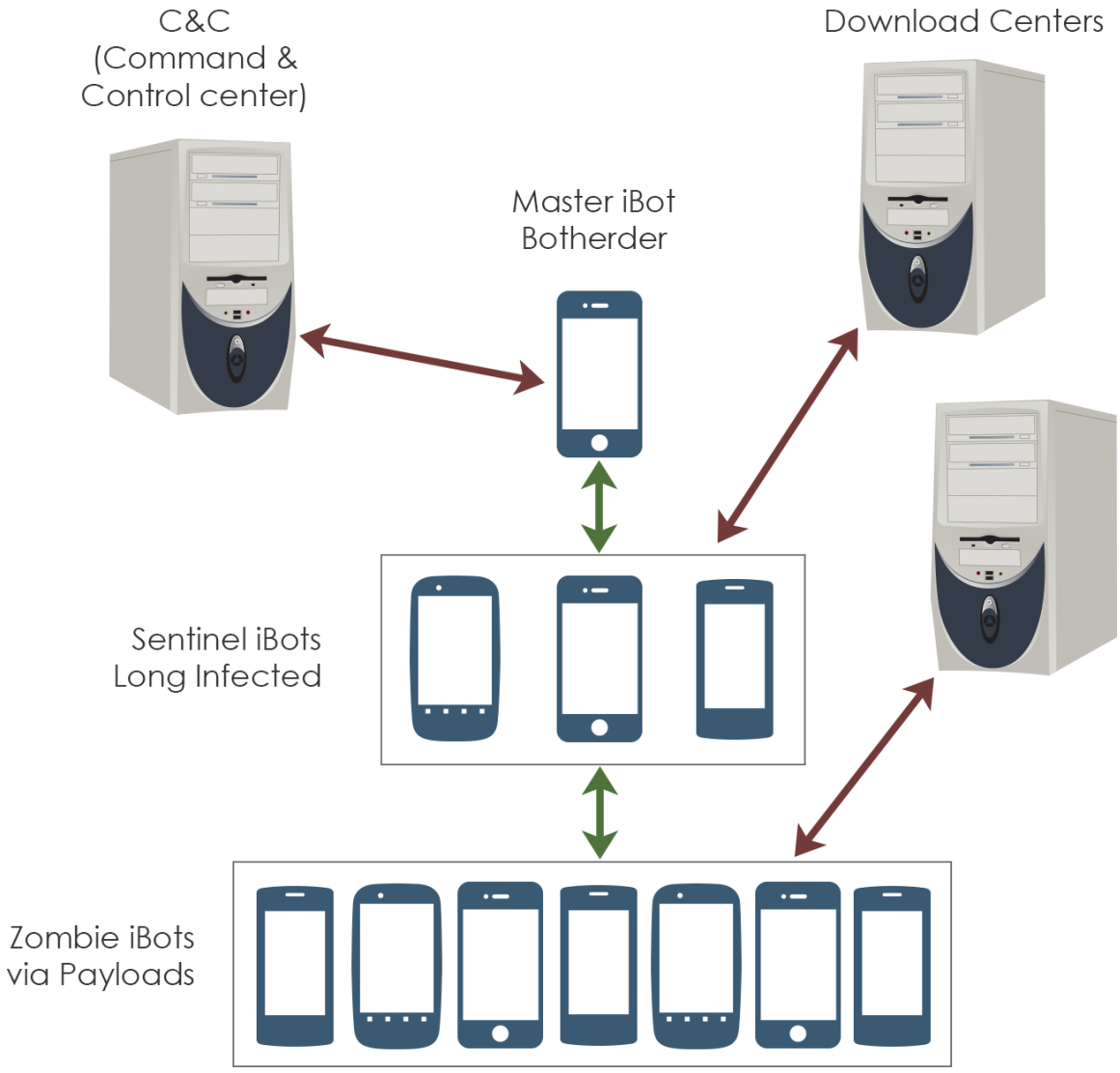

Figure 7: Mobile 'pocket' botnets can be similar to PC botnets

In the same way that botnets can be used to redirect traffic from websites accessed via PCs, botnets can be used to redirect the traffic on mobiles to malicious websites or for the purpose of monetizing.

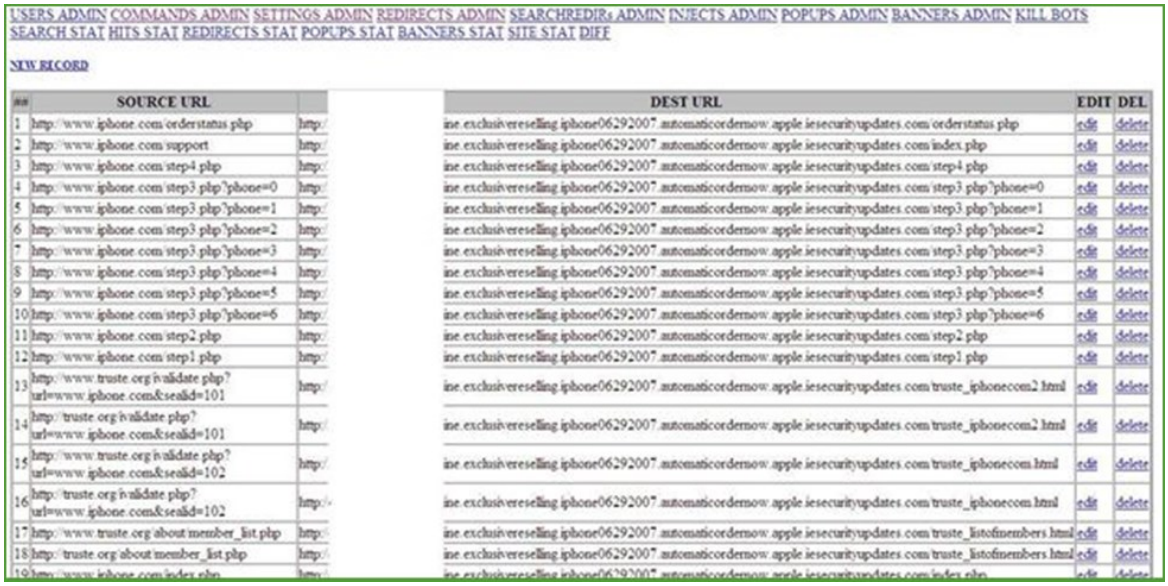

Figure 8: Mobile iPhone botnet for mobile traffic redirects

Mobile botnets can be launched via DDoS attacks. Devices may be infected for a long time without detection just the same as DDoS sent from static computers that become part of a zombie network. Android.DDoS.1.origin creates an application icon that looks similar to Google Play. It connects to a remote server and responds to the command to send packet requests to a specified address[26].

Another kind of bot acts like a worm within the mobile operator's network. One of the most famous examples is IKee.B malware[27], which scans an IP range to detect iPhones with the default "alphine" password on SSHD:

```
sshpass -p alpine ssh -o StrictHostKeyCheck
```

[26] http://news.drweb.com/show/?i=3191&lng=en&c=14

[27] http://mtc.sri.com/iPhone/

Pocket Botnet Examples

Android SmsSend family

Primarily similar in deceptive techniques as fake anti-virus.

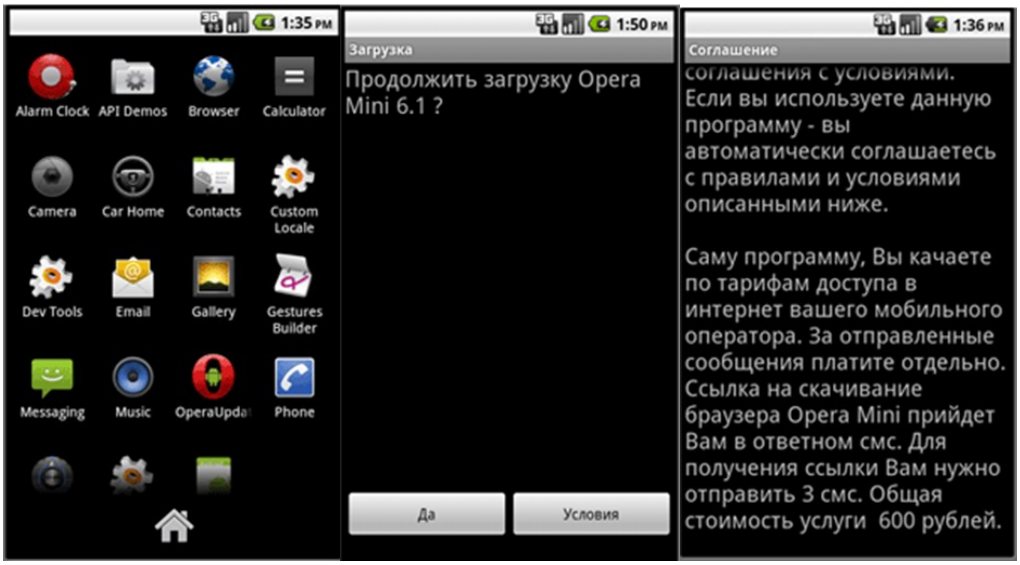

Figure 9: An example of Android SmsSend

ANSERVER-A-Family

Based on permissions and use of C&C servers.

Figure 10: Fake installs appear realistic

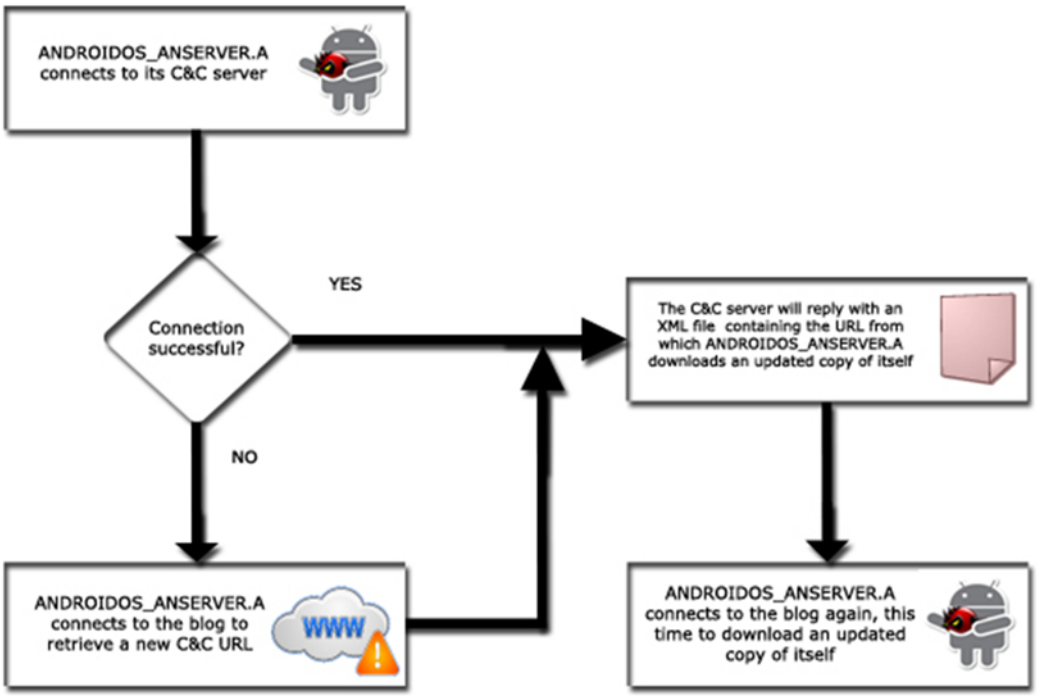

Figure 11: An example of the mobile ANSERVER_A_Family

ThemeInstaller.A

Zombie China

- Infected 1 million Symbian smartphones in 1 week & slower propagation (CNcert)
- Concealment – clear logs, self-destruction, acts when phone not used
- Defense – attacks security software
- Transmission – infects other devices via SMS, downloads new malware from C&C

Pocket Botnet Takedown

US Telco & GG tracker

- GG tracker (abusing premium SMS by malware)
- Signup via website, SMS used to authenticate
- Subscriber pays $9.99 / call
- Operator pays SMS aggregator
- Aggregator pays to content provider
- Content provider pays spammers etc.
- Around 30,000 victims in one week before takedown

There are several tools available which make the building of a botnet for the 'would-be' bot-herder as simple as possible.

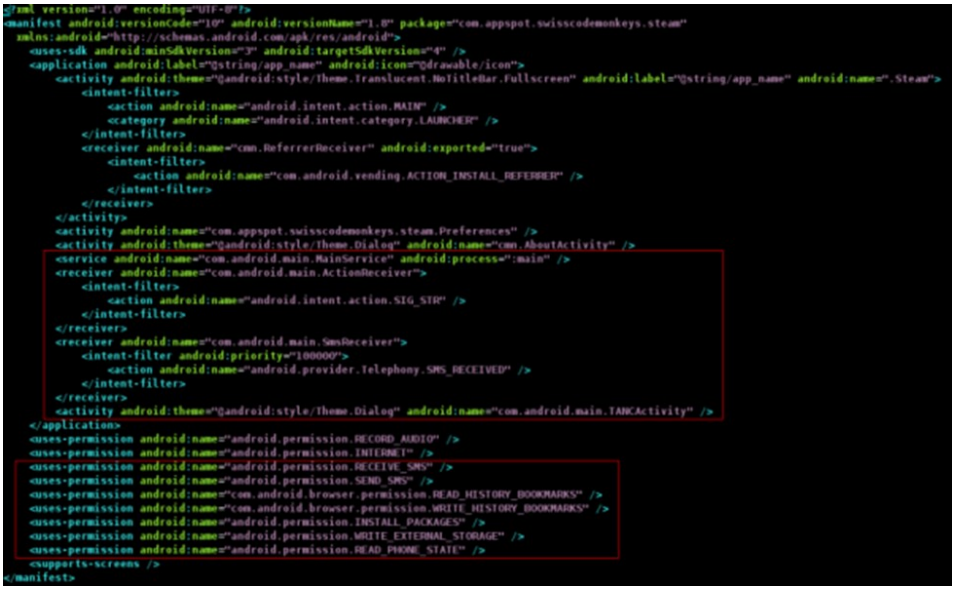

Figure 12: The Pocket Botnet – Build your Own? - Android.Pjapps

Here is a sample of a few easy steps:

Establish a dial in server - based on modem configuration for mgetty

- Establish: `#/AutoPPP/ - a_ppp /usr/sbin/pppd auth -chap +pap login debug`
- Change to = `/AutoPPP/ - a_ppp /usr/sbin/pppd auth -chap +pap login debug`
- Setup PPP options e.g. `ms-dns 3.4.5.6 #replace 3.4.5.6 with DNS address Slave`
- Add users (iBots / zombies) to pap-secrets
- Create Linux users
- Broadcast

iBot –Mobile Zombie detection

The following steps, as an example, provide information about botnet abuse on a mobile device[28].

- agent on mobile devices where possible - (H)IDS
- KB-IDS for android
- Umit[29]
- Mobile Sniffer for Android[30]
- TaintDroid application[31]
- Andromaly[32]

Mobile agents with corporate IDS (NIDS Style), BYOD

- Mobile: agent
- Server: Suricata or Snort
- Connection: VPN through Server

[28] http://www.xlab.si

[29] http://dev.umitproject.org/projects/umitproject/wiki

[30] https://github.com/umitproject/pm-mobile

[31] http://appanalysis.org/

[32] http://code.google.com/p/andromaly/

Mobile Intrusion

Mobile intrusion tools are easy to obtain with a variety of products openly available for purchase via the internet. In many countries it is perfectly legal to buy and to use these products, even if, technically, restrictions apply on how they can be used. Although a gray area, these tools offer a wide range of surveillance techniques under the guise of 'tracking' your children.

Spyware

One such tool, 'FlexiSpy', details its range of features as:

- Call logging
- Sim card change notifications
- Gps tracking
- Sms spying
- Email interception
- Messenger monitoring
- Cell phone bugging
- Call interception

This is a heavy duty range of surveillance tools and legal to buy. The website does, however, display a disclaimer warning that it may be an offense to install software onto another person's phone and that you should seek the advice of a local attorney if you want to use this to monitor anyone.

Legality

The product review page contains further information on the legality of 'Flexipsy' in the United States:

"If you are wondering if Flexispy is legal then the answer to that question is absolutely. Cell phone spy software on the whole is used by many surveillance organizations such as the police which is government regulated. Outside of government, agencies such as private investigators use these apps to determine employee espionage, theft or the most popular reason which is to catch a cheating spouse or lover.

When you think about a cell phone spy app in general you should really think of it as a tool. Like any tool Flexispy is legal however depending on what you are doing with it then you may be using it illegally."[33]

Essentially, Flexispy was designed to be used for monitoring employees and children's mobile phones but there is nothing to stop this software being used by anyone on anyone.

[33] http://flexispy-review.com/category/legal-issues/

Surveillance is an area that is particularly gray with laws that are state or country centric. In the US, intrusion tools are not illegal but the monitoring of an individual may be. The issue relies heavily on moral or ethical judgment as privacy laws have failed to keep pace with the evolution of the internet[34].

FinFisher

In July 2012, researchers from the 'Citizen Lab' published the results of an investigation into 'The FinFisher Suite'[35]. FinFisher is software used to carry out remote intrusion and surveillance by law enforcement and intelligence agencies. It is marketed and sold by UK-based Gamma Group and touted as 'lawful interception' for monitoring criminals[36]. Researchers found that it was 'used in targeted attacks against human rights campaigners and opposition activists in countries with questionable human rights records'[37].

FinFisher was known to have a component that captured passwords and Skype calls with data sent to a FinSpy command and control CC2) server. Additional research published in August 2012 highlighted mobile variants of the Fin Fisher Toolkit, more command and control servers and mobile Trojans[38]. Research recently published 'provides examination of a Fin Spy Mobile sample found in the wild which appears to have been used in Vietnam'[39]. It contains mobile-specific features such as GPS tracking and functionality for silent 'spy' calls to snoop on conversations near the phone.

Further, the FinFisher suite of software has been found in a total of 25 countries.

Drive-by-Downloads

Drive-by-download is an emerging threat vector for Android. A variant of a long-standing desktop attack method, the iframe triggered to download and execute the payload when an infected website is visited by an Android browser. The install does happen automatically but presents as an update which the unsuspecting user is encourage to accept.

The latest threat 'AirDroid' is a web interface that contains a XSS vulnerability[40]. AirDroid sends a text message containing malicious code which, when viewed on the AirDroid web interface, can conduct a cross-site scripting attack. This can lead to

---

[34] http://www.aclu.org/node/36123

[35] https://citizenlab.org/2012/07/from-bahrain-with-love-finfishers-spy-kit-exposed/

[36] https://www.gammagroup.com/

[37] http://www.nytimes.com/2012/08/31/technology/finspy-software-is-tracking-political-dissidents.html

[38] https://citizenlab.org/2012/08/the-smartphone-who-loved-me-finfisher-goes-mobile/

[39] https://citizenlab.org/2013/03/you-only-click-twice-finfishers-global-proliferation-2/#1

[40] http://www.kb.cert.org/vuls/id/557252

information leakage, privilege escalation, and/or denial of service on the host computer.

Mobile Apps

App Stores

"Downloading from App Stores is Risky Business", according to McAfee[41]. App stores are beginning to recognize that scammers are able to exploit users through this widely used channel. Apple, for example, recently updated its app-submission rules in a bid to stem the scammers who upload fake screenshots once their app has been approved[42].

Users remain vulnerable app installation from untrustworthy sites and, despite frequent warnings, are prepared to risk infection when an app is appealing.

Trend Micro recently analyzed more than 2 million apps and classified 293,091 as 'outright malicious'[43]. Some 68,740 of these 'were sourced directly from 'Google Play', which out of around 700,000 Google Play apps in total equates to 1 in 10 apps being malicious.

The Japanese Information Technology Agency (IPA) recently advised users to use alternative app stores in preference to 'Google Play' due to concerns that too few checks were being made before allowing apps to be uploaded to the store[44].

Android APK

There are a number of tools available that can reverse engineer the binary code within the Android APK, which houses the folders and files for app installation. One such tool, 'ApkTool', helps to decode an app, change it and rebuild it. Although intended for 'GOOD' purposes, tools such as this are open to abuse[45]. A recent Andoird blog demonstrated how SwiftKey APK could easily be turned into a key logger using the ApkTool decompiler[46].

---

41 http://www.mcafee.com/uk/resources/white-papers/wp-downloading-apps-risky.pdf

42 http://www.informationweek.co.uk/security/vulnerabilities/apple-targets-app-store-bait-and-switch/240145930

43 http://countermeasures.trendmicro.eu/android-malware-believe-the-hype/

44 http://www.theregister.co.uk/2013/03/04/android_app_google_play_fraud/

45 https://code.google.com/p/android-apktool/

46 http://www.android-app-development.ie/blog/2013/03/06/inserting-keylogger-code-in-android-swiftkey-using-apktool/

Virustotal currently shows 5.6 million reported potentially malicious files for Android (APK, dyn-calls, checks-GPS, etc.), of which 1.3 million are confirmed malicious by 2 or more AV vendors[47].

Intervention, Rules & Classification

Yara – an introduction[48]

Yara is an open source tool developed by Víctor Álvarez in 2009 and as the project states, is designed for malware classification and identification. In reality, the tool is flexible and robust that it can be used to scan any file, malicious or not and quickly identify components, content, and metadata.

Yara does not replace antivirus but is extremely useful when it comes to high volume processing of files with custom signatures. It offers easy to understand language based on PCRE (Perl Compatible Regular Expressions). There are add-ons for text editors offering syntax highlighting and editor tools like G-YARA – a web based rule editor.

Yara is also suitable for the detection of intrusion malware on mobiles.

Here are some of the possible real life scenarios of using YARA for network defense, incident response, forensics, and malware analysis:

- Inspect emails at the gateway for incoming malware or phishing and/or outgoing company sensitive data.
- Use as a secondary antivirus with custom signatures for scanning computer systems.
- Process URLs to classify content of suspicious links before deeper analysis. Scan network data with the help of Chopshop or Yaraprocessor - packet processing open source tools developed by MITRE[49] or use in modules for proxy servers (see Yara C-ICAP Server Module by Fyodor Grave)[50].
- Use signatures in FireEye appliances or incorporate in custom tools, sandboxes and scanners with the help of available Yara-python or Yara-ruby flavors[51].
- Scan found files or compromised systems to determine if this is something you have already seen before, or find malware not detected by other means.
- Classify malware by families, campaigns, actors, unique traits, or exploits used.
- Use as part of open source tools like Cuckoo sandbox, MIDAS (Metadata Inspection Database Alerting System), Moloch (IPv4 packet capturing (PCAP),

[47] https://www.virustotal.com

[48] http://www.deependresearch.org/2013/02/yara-resources.html

[49] http://www.mitre.org/work/cybersecurity/blog/cyber_tools_shields.html

[50] https://github.com/MITRECND/yaraprocessor

[51] http://www.fireeye.com/

indexing and database system) or add custom scanning functionality to custom tools and systems.

- Check for personally identifiable information (PII), financial / credit card, or sensitive / classified information on compromised systems during incident response triage phase.
- Leverage Volatility when looking for malware artefacts in computer memory dumps and log2timeline outputs. Use with SIFT and RemNux forensics virtual machines that have it installed.
- Scan forensic images for compromise or investigative clues.
- Share research and indicators of compromise with other groups in the form of signatures.

More information and links to these and other Yara tools can be found at *DeepEnd Research: Yara Resources*[52].

Yara Exchange[53]

Yara Exchange was formed by DeepEnd Research in August 2012 in order to build a community where security researchers gather to discuss various topics related to the use of YARA.

Sample Yara Rules for Mobiles

Yara can identify suspicious behavior on mobiles. It is especially useful in the detection of mobile intrusion malware and covers Blackberry and Android for 'Flexispy'[54]:

```
{
    meta:
        description = "FlexiSpy, FeelSecure and other mobile spyware from
Vervata."
        author = "Tim Ehrhart"
        source = "Various mobile malware samples"
        date = "2012-11-15"
        version = "1.1"
    strings:
        $feelsecure1 = "wefeelsecure.com"
        $feelsecure2 = "res/raw/feelsecure"
        $flexispyandroid1 = "res/layout/gps_time_interval_dialog.xml"
        $flexispyandroid2 = "assets/libsmitm.so"
        $flexispyandroid3 = "res/drawable/fspy.png"
```

[5252] http://www.deependresearch.org/2013/02/Yara-resources.html>

[53] http://www.deependresearch.org/2012/08/Yara-signature-exchange-google-group.htm>

[54] Courtesy 'Lookout Mobile Security' https://www.lookout.com/

```
        $flexispyandroid4 = "assets/temp_app.apk"
        $flexispyblackberry1 = "LICENSE_GENERATOR_AUTHENTICATION_ERROR"
        $flexispyblackberry2 = {D8 2C 20 4C 41 43 3A 00 00 06 00 24 D8 2C 20
4D
43 43 3A 00 00 06 00 24 D8 2C 20 4D 4E 43 3A 00 00 14 00 24}
        $flexispyblackberry3 = "EmailCapture.startCapture()"
        $flexispyblackberry4 = "SpoofSMSCmd"
        $flexispyblackberry5 = "ENABLE_SPYCALL"
        $flexispyblackberry6 = "MONITOR_NUMBER"
        $flexispyblackberry7 = "Spy call is enabled"
        $flexispyblackberry8 = "isFlexiKey"
        $flexispyblackberry9 = "net_rim_platformapps_resource_security"
    condition:
        any of them
}
```

Figure 13: Detection of 'FlexiSpy' in action

Components of Yara can detect valid APK, for further investigation. In this example the APK is disassembled from a JAR/ZIP/etc.

```
{
       meta:
author = "Tim Strazzere"
date = "10/25/2012"
version = "1.0"
tag = "Android"
comment = "Attempted to detect an APK file with a classes.dex that is signed"
     strings:
$PK_HEADER = {50 4B 03 04}
$MANIFEST = "META-INF/MANIFEST.MF"
$DEX_FILE = "classes.dex"
       condition:
$PK_HEADER in (0..4) and $MANIFEST and $DEX_FILE
}
```

Figure 14: Detection of suspected suspicious decompiling of APK code

Strategy Guide

What can be done? What should be done?

There are no 'quick fix' solutions to the problem of mobile malware or to the related existence of an underground economy whose success depends on a supply and demand economy.

Using the example, and the failings, of the existing computer industry, it is obvious that piecemeal solutions alone to not work. Global problems require global efforts and this remains one of the biggest hurdles to overcome.

An integrated approach requires efforts from mobile operators, developers, app stores, vendors, manufacturers and, of course, users. It requires an improvement in hygiene, disclosure on how data is collect and used, more transparency around apps, guidelines for developers, Terms of Service and policies, monitoring of traffic misuse or abuse, and tougher demands from users on permissions and settings accessibility.

Industry specific measures, too, can go a long way to help protect against mobile fraud. For example, voice biometry recognition fraud detection software helps to aggregate voice records and retain the history of customers' phone calls for further analysis.

From a pragmatic viewpoint we should accept that to some extent mobile fraud is here to stay. With that in mind organizations can take control of mobile management within the workplace and implement a number of damage limitation measures. Here, some of these measures are set out together with some suggestions on information exchange and guidelines for application developers.

A Practical Guide to Risk Analysis for Organizations

OWASP (The Open Web Application Security Project) issues a regular report, the 'Mobile Security Project' which details the 'Top 10 Mobile Risks". The OWASP analysis estimates factors such as 'likelihood' and 'impact' as well as the 'threat' factor to determine the greatest mobile vulnerabilities.

The latest "Top 10 Mobile Risks" (5 March 2013) are as follows:

1. Insecure Data Storage
2. Weak Server Side Controls
3. Insufficient Transport Layer Protection
4. Client Side Injection
5. Poor Authorization and Authentication
6. Improper Session Handling
7. Security Decisions Via Untrusted Inputs
8. Side Channel Data Leakage
9. Broken Cryptography
10. Sensitive Information Disclosure

The OWASP approach provides a practical guide towards risk analysis which can be a valuable aid in determining what solutions can, or need, to be applied.

According to OWASP the No. 1 risk, 'Insecure Data Storage' reinforces the conclusion of a number of other sources. Insecure data storage, in the form of 'Jailbroken' devices, is heavily criticized from a number of sources, including service providers, as seen in a recent warning from Apple[55].

This warning comes in the light of a ruling in the United States, enforced 23 January 2013, requiring users to obtain permission from the provider before 'unlocking' new smartphones[56].

This now gives operators in the United States a legal platform from which the service to 'unlocked' devices may be denied. It will be interesting to see what impact the new ruling has and what users of 'unlocked' phones will do if the service to their phone is blocked. Older devices, and those outside of the United States, remain free from this restriction.

App stores and permissions

A mechanism to verify applications before publishing on Apple Store, Google Play and Android Market should be introduced. A standard application policy and responsibility for regular auditing would help strengthen the verification process.

App permissions are currently over-generalized. Developers should be required to specify exactly what the permissions are required for. For example, if an application wants to "test access to protected storage", the user who is considering installing it should be able to easily find out why.

Information Exchange

CERTs/CSIRTs should tighten-up on their incident response and analysis timeframes via their hotlines as it sometimes it takes more than 8 hours to delete malicious application from there.

A transparent cyber-intelligence exchange for information relating to "underground" mobile subscription programs could be set-up by APWG members. This would provide valuable information on mobile crime which, currently, lacks proper consideration. Cybercriminals find it all too easy to use the underground markets to commercialize their activities.

High on the agenda for information exchange would be "short number" abuses, something that the phone operators all too easily ignore. Within the exchange, APWG members would be able to share information about detected "short numbers", WHOIS details of the links found in the underground communities promoting spam/smishing/phishing activities and to aggregate the data in a centralized way for

[55] http://support.apple.com/kb/HT3743

[56] http://arstechnica.com/tech-policy/2012/10/jailbreaking-now-legal-under-dmca-for-smartphones-but-not-tablets/

further investigation. It would also be valuable to monitor the e-commerce websites which help the cybercriminals to cash-out money (EPASE, SMS billings).

Service operators should aim to improve end-user awareness of malicious apps and other security measures. Here, the APWG can help with special recommendations and white papers that the customers of Mobile ISPs can circulate as part of an education program. This could be distributed during the agreement signing process and through the operator's own corporate channels.

Guidelines for secure application development

The developers of online-banking and mobile e-commerce applications should follow an agreed set of guidelines on secure application development. An example, in brief, of secure mobile development practices for app development is available from Via Forensics[57]. However, there are many good sources for this information with a few listed in the 'Further Reading' section. In summary, the key principles are:

1) Mobile Metadata security

Specific compilation parameters such as PIE (Position Independent Executable), SSP (Stack Smashing Protection) and ARC (Automatic Reference Counting) should be used in support of ASLR (address space layout randomization) and to detect and mitigate software vulnerabilities.

It is strongly recommended that all debugging mechanisms such as NSLog should be disabled before banking mobile application are published on Google Play, Apple Store or the Android market otherwise it may still be possible to replicate some sensitive information there and permit hackers to steal sensitive information locally after a device has been infected.

2) Application Protocols security

Some mobile applications are developed with debugged or disabled SSL functions. It aids attackers to carry out "Man in the Middle" attacks and to intercept or to modify data.

Online-banking and e-commerce server-side communications need to be secured to avoid attacks using XSS, CSRF and XXE that carry out integrity and confidentiality modifications.

3) Embedded Databases and Storage security

The choice of embedded database is an important factor in storage security for mobile applications.

SQLite is probably the most widely deployed database engine for mobile storage security but this, and other similar mechanisms, are not risk free. Embedded databases

[57] Secure mobile development best practices (https://viaforensics.com/resources/reports/best-practices-ios-android-secure-mobile-development/)

are not automatically encrypted, enabling attackers to gain access and to extract data, sometimes using SQL injection.

4) Password information Storage security

The Keychain file (iOS) stores passwords on the iPhone including any used for email. Attackers can overwrite the keychain giving access to all locally stored data including emails, contacts, photos, etc,. As at 05.12.12 all versions of iOS up 6.0.1 were affected[58].

It is strongly recommended that nothing sensitive is created or stored on the client-side (local) of iOS devices in order to avoid possible breaches.

[58] https://www.sit.fraunhofer.de/fileadmin/dokumente/sonstiges/iPhone_keychain_faq.pdf

Conclusions

This report verifies the existence of a mobile malware market that is poised to take full advantage of an industry, yet in its infancy, as it continues to grow and evolve. The underground mobile market benefits from the existence of an established *modus operandi* for desktop computer malware that is well-structured and successful.

Mobile malware marketeers, like their desktop counterparts, are savvy, resourceful, and quick to exploit new technologies, applications and functions as soon as they develop. The incentive is to cash in on a global mobile payments market expected to exceed $1.3tn by 2017.

The growing prosperity of the emerging economies sees Asia driving the mobile market but it would be dangerous to under-evaluate the influence of both Africa and South America.

The emerging markets have an important role to play in the expansion of mobile infrastructures and services. Equally important are differences in way of life, culture and individual countries' economic market i.e.,: salaries, transportation, education and lack of wired telecommunication infrastructures, etc., as these have a bearing on the ultimate success of an underground economy based on supply and demand.

The mobile malware market is already alive and thriving. Only an integrated, global response based on cooperation, education and awareness can limit its success.

Appendix 1 - Further Reading

'Dissecting android malware - Characterization and evolution'[59]

Zhou and Jiang

An in-depth analysis of more than 1,200 malware samples collected between august 2010 and October 2011. The study follows the mobile malware process from installation to activation researching various 'modus operandi' and malicious payload performances against current detection methods. The authors conclude that detection rates ranging from as little as 20.2% to 79.6% indicate a need for "better developed next generation anti-mobile malware solutions."

Presented at the IEEE Symposium on Security & Privacy, 2012

'SMS stealing apps uploaded to Google Play by Carberp banking malware gang'[60]

Group-IB – January 2013

Several month ago Group-IB detected mobile-banking malware through Google Play by Sberbank request (Russian leading national bank). Analyzing the functionality of the agent it is possible to classify it as SMSStealer.APK designed to infect Android devices. Evidence files after installation displays the following graphical user interface used to request user's authorization through a phone number verification process. The scam schema based on the interception of SMS used in the authentication process could be very useful to banking frauds. US and Canada banks, but also other financial institutions, use One Time Password token sent via SMS, clearly an attacker intercepting it could complete fraudulent transactions.

'Russian Underground 101' Trend Micro Incorporated Research Paper 2012[61]

Max Goncharov

An analysis of the Russian underground market based on data found on online forums and other services popular with cybercriminals. The move from simply a hobby to a structured means of earning a living, online fraud is examined from the viewpoint of the professional fraudster. Hacking, the generation of traffic, and code writing for Trojans, exploits and other malware are just some of the services that can be bought online. The fundamental concepts of the Russian underground market are examined along with relationships within the community. Data is provided on the prices charged on a range of services. This paper contributes valuable information on the types of criminal activity that pass as legitimate services to fill the pockets of online fraudsters operating a successful underground economy.

---

[59] http://www.malgenomeproject.org/

[60] http://thehackernews.com/2013/01/dissecting-mobile-malware.html

[61] http://www.trendmicro.com/cloud-content/us/pdfs/security-intelligence/white-papers/wp-russian-underground-101.pdf

'Guidelines on Hardware-Rooted Security in Mobile Devices' (Draft) – Recommendations of the National Institute of Standards and Technology

Lily Chen, Joshua Franklin, Andrew Regenscheid – October 2012

In this draft paper NIST advises on the lack of fundamental security at the root level of current mobile devices in contrast to the newer generation laptops and computers. The paper recommends industry action to ensure that these capabilities are implemented as a basic component and to aid organizations in securing their own mobile devices or personally-owned devices used for work (BYOD). NIST provides guidelines for the introduction of baseline security technologies for a wide range of mobile devices comprising three main security elements implemented at the Roots of Trust (RoTs) level.

This paper received criticism from the Telecommunications Industry Association (TIA) for its emphasis on Trusted Platform Module (TPM) as the solution to mobile security problems[62]. The TIA described NIST's proposal as "over-prescriptive" and suggested that TPM was only one of many ways in which to implement security.

Mobile Malware Evolution: Part 6[63]

Denis Maslennikov, Kaspersky Labs – February 13

A regular overview of the events of the previous year: in Part 6 Kaspersky Labs reflects on 2012 with trends depicted through qualitative and quantitative analysis with forecasts on the development of mobile malware in 2013.

Kaspersky Labs reports on the emergence of a mobile version of FinSpy with an in-depth analysis on this 'notable' event along with the 'Red October' incident when mobile devices were found to have been targeted as part of an espionage attack.

Safety on the Line – A project report from Freedom House supported by the Board of Governers[64]

Freedom House, the independent watchdog organization dedicated to the expansion of freedom around the world, evaluates the risk and vulnerabilities of mobile services and apps in 12 countries: the Republic of Azerbaijan, the Republic of Belarus, the People's Republic of China, the Arab Republic of Egypt, the Islamic Republic of Iran, Libya, the Sultanate of Oman, the Kingdom of Saudi Arabia, the Syrian Arab Republic, the Tunisian Republic, the Republic of Uzbekistan, and the Socialist Republic of Vietnam.

Mobile technologies such as operating systems, applications and protocols are analyzed for security and privacy. The researchers conclude that, in the countries

---

[62] http://www.networkworld.com/news/2012/121712-nist-tia-265172.html?page=1

[63] http://www.securelist.com/en/analysis/204792283/Mobile_Malware_Evolution_Part_6

[64] http://www.freedomhouse.org/report/special-reports/safety-line-exposing-myth-mobile-communication-security

studied, "… there is a phenomenally high level of penetration of mobile handsets in almost all markets…" There are significant risks to mobile devices in these countries at multiple levels from hardware through to operating and regulatory level.

Do-It-Yourself Guide to Cell Phone Malware[65]

William r. Mahoney & Craig A. Pokorny, University of Nebraska at Omaha – Jan 2009

Off-the-shelf code kits for mobiles are easy to find and to purchase, as this study suggests, and aid the creation of malicious software. Sourcing the programming interfaces and tools presented few obstacles too as the researchers found to their surprise. Just a little ability and some knowledge of syntax errors is all that is needed to create malware for mobiles.

Android Malware Forensics: Reconstruction of Malicious Events[66]

Juanru Li, Dawu Gu, Yuhao Luo, Dept. of Computer Science and Engineering, Shanghai Jiao Tong University

Using a real-life case study involving a malicious event on an Android operating system researchers from Shanghai Jiao Tong University demonstrate how suspicious programs can be quickly identified and disabled. The key to defeating the malicious code is found in malware behavior analytics. The paper follows the forensic analysis and systematic process used in the detection of malware, and describes typical malicious behaviors observed on Android systems.

Legal Implications of Countering Botnets[67]

A joint report from the NATO Cooperative Cyber Defence Centre of Excellence and the European Network and Information Security Agency (ENISA)

This joint report uses Estonian and German legislation to analysis the legal implication of botnet mitigation from two different legal entities. The report suggests a legal framework in the case of unlawful damage, or similar circumstances, which may occur during the botnet mitigation process. It sets out a number of legal considerations and potential risks that may arise as a result of botnet mitigation.

---

[65] http://paper.ijcsns.org/07_book/200901/20090135.pdf

[66] http://ieeexplore.ieee.org/xpl/articleDetails.jsp?reload=true&arnumber=6258204

[67] http://www.ccdcoe.org/205.html

Other Resources

- http://ssv.sebug.net/IOS_Application_Security_Testing_Cheat_Sheet
- http://www.exploit-db.com/wp-content/themes/exploit/docs/18831.pdf
- http://media.blackhat.com/bh-us-12/Briefings/Engler/BH_US_12_Engler_SIRA_WP.pdf
- ENISA Smartphone Security: http://www.enisa.europa.eu/activities/identity-and-trust/risks-and-data-breaches/smartphones-information-security-risks-opportunities-and-recommendations-for-users/at_download/fullReport

Appendix 2 – Glossary of Terms

ADSL (Address Space Layout Randomization)

A technique used to increase the search space in order to lower the chance of code location detection.

AFE (Android Framework for Exploitation)

An open source project to explore vulnerabilities and weaknesses in Android devices. The framework is extendible to allow integration of custom tools or modules[68].

APK (.apk)

Android application file format package file for distribution of application software to run on Google's Android operating system.

App

A mobile software application designed to operate on mobile devices.

AS (Autonomous System)

An AS is a unit of router policy, either a single network or a group of networks that is controlled by a common network administrator on behalf of an entity such as a university, a business enterprise, or Internet service provider. An AS is also sometimes referred to as a routing domain. Each autonomous system is assigned a globally unique number called an Autonomous System Number (ASN).

Badware

Software that fundamentally disregards a user's choice about how their computing device will be used. Types of badware are **spyware**, **malware**, or **deceptive adware**. Common examples of badware include free screensavers that surreptitiously generate advertisements, malicious web browser toolbars that lead to an unexpected website, and **keylogger** programs that can transmit personal data to malicious parties.

Blacklist

In computing, a blacklist is a basic access control mechanism used to deny right of entry. The opposite of this is a whitelist, which only entry to the named entities. As a sort of middle ground, a gray list contains entries that are temporarily blocked or temporarily allowed. Gray list items may be reviewed or further tested for inclusion in a blacklist or whitelist. Some communities and webmasters publish their blacklists for the use of the general public.

[68] http://news.softpedia.com/news/Experts-Demonstrate-Security-Holes-in-Android-with-Exploitation-Framework-285047.shtml

Bluetooth

The Bluetooth standard was officially adopted in 1998. Today, Bluetooth technology is available everywhere enabling wireless communication between many different types of devices.

Botnet

Botnet is a term for a collection of software robots, or bots, that run autonomously and automatically. The term is now mostly associated with malicious software used by cyber criminals, but it can also refer to the network of infected computers using distributed computing software.

Blended Attack

A combination of attacks such as worms, viruses, Trojans used to maximize the extent of the exploit. One combination has been named 'MALfi' meaning a combination of RFI (remote file inclusion), LFI (local file inclusion), XSA (cross server attack), and RCE (remote code execution).

C&C Servers

The central communication hub that bots report back to and where the bot herder (or master) communicates with the botnet group. Communication channels vary but IRC (Internet Relay Chat) can be used for covert messaging.

DDOS (Distributed Denial of Service)

DDoS attacks or floods can be executed in a variety of ways. The desired effect is to interrupt the normal business of a web service. Attackers use the power of multiple computer systems, either via or a botnet or from number of users, to flood the system with multiple requests until it crashes. Another method to launch an attack is to amplify DNS requests via open resolvers which uses few resources to achieve its aim.

DNS (Domain Name System)

DNS associates various information with domain names; most importantly, it serves as the "phone book" for the Internet by translating human-readable computer hostnames, e.g. www.example.com, into IP addresses, e.g. 208.77.188.166, which networking equipment needs to deliver information. A DNS also stores other information such as the list of mail servers that accept email for a given domain, by providing a worldwide keyword-based redirection service.

Exploit

An exploit is a piece of software, a chunk of data, or sequence of commands that take advantage of a bug, glitch or vulnerability in order to cause irregular behavior to occur on computer software, hardware, or something electronic. This frequently includes such things as violently gaining control of a computer system or allowing privilege escalation or a denial of service attack.

GPRS (General Packet Radio Service)

A wireless data service capable of supporting a number of protocols 'on the move'. Mobiles benefit from instant internet access, SMS and MMS messaging and location-based services.

Hosting

Usually refers to a computer (or a network of servers) that stores the files of a web site which has web server software running on it, connected to the Internet.

IMEI (International Mobile Station Equipment Identity)

Identification number used by mobile network providers. IMEI can be used to "blacklist" a stolen phone or to identify a target for interception, lawful or otherwise.

IP (Internet Protocol)

IP is the primary protocol in the Internet Layer of the Internet Protocol Suite and has the task of delivering data packets from the source host to the destination host solely based on its address.

IPv4

Internet Protocol version 4 (IPv4) is the fourth revision in the development of the Internet Protocol (IP). Pv4 uses 32-bit (four-byte) addresses, which limits the address space to 4.3 billion possible unique addresses. However, some are reserved for special purposes such as private networks (18 million) or multicast addresses (270 million).

IPv6

Internet Protocol Version 6 (IPv6) is a version of the Internet Protocol that is designed to succeed IPv4. IPv6 uses a 128-bit address, IPv6 address space supports about 2^128 addresses

ISP (Internet Service Provider)

A company or organization that has the equipment and public access to provide connectivity to the Internet for clients on a fee basis, i.e. emails, web site serving, online storage.

Jailbreak

Unlocking or 'cracking' the operating system (iOS) root program to remove restrictions. Jailbreaking enables the use SIM cards from other providers, to run 3rd party programs, to exploit software/hardware.

Kernel

A module where the main processes of the operating system occur. The kernel manages (bridges) the essential services used by other parts of the operating system or application.

Keylogging

A program that captures keyboard activity often without the user's knowledge. Keylogging is used to surreptitiously record and capture personal data.

LFI (Local File Inclusion)

Use of a file within a database to exploit server functionality. Also for cracking encrypted functions within a server, e.g. passwords, MD5, etc.

Malicious Links

Links planted on a site to deliberately send a visitor to a malicious site to plant viruses, spyware or any other type of malware on a computer e.g., fake security system. They can be planted within a feature of the site or masked to misdirect the visitor.

MITM (Man-in-the-Middle)

Interception between two systems used as a means of attack against the user by the re-routing of a connection to establish a proxy service. The attacker can execute actions against the user including reading, inserting or modifying data.

MTAN (Mobile Transaction Authentication Number)

Mobile banking version of the classic TAN used to authorize online financial transactions proving two-factor authentication.

NFC (Near Field Communication)

The technology and standards that enable devices to communicate using radio frequency from close proximity. NFC enables contactless payment, file transfer and pairing via Bluetooth, and the sharing of information via social networking, etc.

NS (Name Server)

Every domain name must have a primary name server (eg. ns1.xyz.com), and at least one secondary name server (ns2.xyz.com etc). This requirement aims to make the domain still reachable even if one name server becomes inaccessible.

Open Source Security

Most commonly applied to the source code of software or data made available to the general public with relaxed or non-existent intellectual property restrictions. Users are able to create user-generated software content and advice through incremental individual effort or through collaboration.

OWASP (The Open Web Application Security Project)

A worldwide not-for-profit charitable organization dedicated to improving security in software. Its stated mission is to make software security visible, so the individuals and organizations can make informed decisions about true software risks.

PIE (Position Independent Executable)

Binaries made entirely from position-independent code that executes regardless of its absolute address.

Phishing

A type of deception designed to steal valuable personal data, such as credit card numbers, passwords, account data, or other information. Phishing is typically carried out using e-mail (where the communication appears to come from a trusted website) or SMS.

Pocket Botnet

Botnets networked via mobile devices. Mobile botnets are similar in structure to the 'classic' botnet but can be spread via the mobile operator's network or any network that the device connects to. The growing trend for BYOD within the workplace gives attackers a potential access point into an organization's network.

PtH (Pass-the-Hash)

Unauthorized privileged access provides a dump of account name and password hashes (algorithm based password & plain text data encryption). This is an emerging technique within mobile malware, especially designed for financial fraud, to capture the user's primary and two factor authentication sessions and to new sessions.

RFI{D} (Remote File Inclusion)

A technique often used to attack Internet websites from a remote computer. With malicious intent, it can be combined with the usage of XSA (Cross Server Attacks) to harm a web server. During XSA, malicious clients, or attackers, compromise the servers of web hosting companies and use their services for their own purposes or to enhance their attack.

Smishing

Messaging version of the phishing scam. A text message is delivered with a URL/link that leads to a scam website or a telephone number that connects to an automated message. The aim is to collect sensitive, personal or financial data from the device user.

SMS (Short Message Service)

A quick and convenient way of sending short messages via text using a mobile phone. SMS is available on a wide range of other devices including satellite and landline networks.

Spam

Spam is the term widely used for unsolicited e-mail. Spam is junk mail on a mass scale and is usually sent indiscriminately to hundreds or even hundreds of thousands of inboxes simultaneously.

Spyware

Software used to gather personal information without the consent of the user usually in the form of advertisements (Adware). Spyware is used mainly to track online behaviour but it can also change settings and allow additional programs to be installed.

Trojans

Also known as a Trojan horse- software that appears to perform, or actually performs, a desired task for a user while executing a harmful task without the user's knowledge or consent.

UDID (Unique Device Identifier)

An Apple device identifier containing 40 characters. UDIDs are used to track a subscriber's behaviour for the purpose of targeted application marketing but UDIDs can also be abused by third-parties. Apple recently announced it will no longer accept new apps or app updates that access UDIDs[69].

Virus

A malicious software program that spreads as a consequence of an action by the user. A virus reproduces by attaching itself to a computer program.

Wi-Fi (Wireless Fidelity)

Wireless radio technology that enables connectivity at home, in the office, at the airport, or 'on the move'.

Worms

A malicious software program that can reproduce itself and spread from one computer to another over a network. A worm is self-contained, requires no action by the user and can send copies of itself across a network.

WLAN (Wireless Local Area Network)

Wireless high-frequency radio waves communicate between devices to enable Internet connection through an access point or local network.

YARA

An open source tool designed for malware classification and identification. It can be used to scan any file to identify its components, content and metadata. Yara Exchange gives security researchers the opportunity to discuss the results of their findings.

[69] https://developer.apple.com/news/