Mobile Security

Mobile security or mobile phone security has become increasingly important in mobile computing. It is of particular concern as it relates to the security of personal information now stored on smartphones.

More and more users and businesses use smartphones as communication tools but also as a means of planning and organizing their work and private life. Within companies, these technologies are causing profound changes in the organization of information systems and therefore they have become the source of new risks. Indeed, smartphones collect and compile an increasing amount of sensitive information to which access must be controlled to protect the privacy of the user and the intellectual property of the company.

All smartphones, as computers, are preferred targets of attacks. These attacks exploit weaknesses related to smartphones that can come from means of communication like SMS, MMS, wifi networks, and GSM. There are also attacks that exploit software vulnerabilities from both the web browser and operating system. Finally, there are forms of malicious software that rely on the weak knowledge of average users.

Different security counter-measures are being developed and applied to smartphones, from security in different layers of software to the dissemination of information to end users. There are good practices to be observed at all levels, from design to use, through the development of operating systems, software layers, and downloadable apps.

Challenges of mobile security

Threats

A smartphone user is exposed to various threats when he uses his phone. These threats can disrupt the operation of the smartphone, and transmit or modify the user data. For these reasons, the applications deployed there must guarantee privacy and integrity of the information they handle. In addition, since some apps could themselves be malware, their functionality and activities should be limited (for example, accessing location information via GPS, address book, transmitting data on the network, sending SMS that are charged, etc.).

There are three prime targets for attackers:

- **Data**: smartphones are devices for data management, therefore they may contain sensitive data like credit card numbers, authentication information, private information, activity logs (calendar, call logs);
- **Identity**: smartphones are highly customizable, so the device or its contents are associated with a specific person. For example, every mobile device can
transmit information related to the owner of the mobile phone contract, and an attacker may want to steal the identity of the owner of a smartphone to commit other offenses;

- **Availability**: by attacking a smartphone you can limit access to it and deprive the owner of the service

The source of these attacks are the same actors found in the non-mobile computing space:[1]

- **Professionals**, whether commercial or military, who focus on the three targets mentioned above. They steal sensitive data from the general public, as well as undertake industrial espionage. They will also use the identity of those attacked to achieve other attacks;
- **Thieves** who want to gain income through data or identities they have stolen. The thieves will attack many people to increase their potential income;
- **Black hat hackers** who specifically attack availability. Their goal is to develop viruses, and cause damage to the device. In some cases, hackers have an interest in stealing data on devices.
- **Grey hat hackers** who reveal vulnerabilities. Their goal is to expose vulnerabilities of the device. Grey hat hackers do not intend on damaging the device or stealing data.[2]

**Consequences**

When a smartphone is infected by an attacker, the attacker can attempt several things:

- The attacker can manipulate the smartphone as a zombie machine, that is to say, a machine with which the attacker can communicate and send commands which will be used to send unsolicited messages (spam) via sms or email:[3]
- The attacker can easily force the smartphone to make phone calls. For example, you can use the API (library that contains the basic functions not present in the smartphone) PhoneMakeCall by Microsoft, which collects telephone numbers from any source such as yellow pages, and then call them.[3] But the attacker can also use this method to call paid services, resulting in a charge to the owner of the smartphone. It is also very dangerous because the smartphone could call emergency services and thus disrupt those services;[3]
- A compromised smartphone can record conversations between the user and others and send them to a third party.[3] This can cause user privacy and industrial security problems;
- An attacker can also steal a user's identity, usurp their identity (with a copy of the sim, telephone, etc.), and thus impersonate the owner. This raises security concerns in countries where smartphones can be used to place orders, view bank accounts or are used as an identity card;[3]
- The attacker can reduce the utility of the smartphone, by discharging the battery. [4] For example, he can launch an application that will run continuously on the smartphone processor, requiring a lot of energy and draining the battery. One factor that distinguishes mobile computing from traditional desktop PCs is their limited performance. Frank Stajano and Ross Anderson
first described this form of attack, calling it an attack of "battery exhaustion" or "sleep deprivation torture".\textsuperscript{[5]}

- The attacker can prevent the operation and/or starting of the smartphone by making it unusable.\textsuperscript{[6]} This attack can either delete the boot scripts, resulting in a phone without a functioning OS, or modify certain files to make it unusable (e.g. a script that launches at startup that forces the smartphone to restart) or even embed a startup application that would empty the battery;\textsuperscript{[5]}
- The attacker can remove the personal (photos, music, videos, etc.) or professional data (contacts, calendars, notes) of the user.\textsuperscript{[6]}

**Attacks based on communication**

**Attack based on SMS & MMS**

Some attacks derive from flaws in the management of SMS and MMS.

Some mobile phone models have problems in managing binary SMS messages. It is possible, by sending an ill-formed block, to cause the phone to restart, leading to denial of service attacks. If a user with a Siemens S55 received a text message containing a Chinese character, it would lead to a denial of service.\textsuperscript{[7]} In another case, while the standard requires that the maximum size of a Nokia Mail address is 32 characters, some Nokia phones did not verify this standard, so if a user enters an email address over 32 characters, that leads to complete dysfunction of the e-mail handler and puts it out of commission. This attack is called "curse of silence". A study on the safety of the SMS infrastructure revealed that SMS messages sent from the Internet can be used to perform a distributed denial of service (DDoS) attack against the mobile telecommunications infrastructure of a big city. The attack exploits the delays in the delivery of messages to overload the network.\textsuperscript{[citation needed]}

Another potential attack could begin with a phone that sends an MMS to other phones, with an attachment. This attachment is infected with a virus. Upon receipt of the MMS, the user can choose to open the attachment. If it is opened, the phone is infected, and the virus sends an MMS with an infected attachment to all the contacts in the address book. There is a real world example of this attack: the virus Commwarrior\textsuperscript{[6]} uses the address book and sends MMS messages including an infected file to recipients. A user installs the software, as received via MMS message. Then, the virus began to send messages to recipients taken from the address book.

**Attacks based on communication networks**

**Attacks based on the GSM networks**

The attacker may try to break the encryption of the mobile network. The GSM network encryption algorithms belong to the family of algorithms called A5. Due to the policy of security through obscurity it has not been possible to openly test the robustness of these algorithms. There are two main variants of the algorithm that are deployed today: A5/1 and A5/2 (stream ciphers), the latter being a weaker version of encryption for countries with legal restrictions on the use of cryptographic schemes. Since the encryption algorithm was made public, it was proved it was possible to
break the encryption in about 6 hours.[8] Both algorithms are at the end of their life and will be replaced by stronger public algorithms: the A5/3 and A5/4 (Block ciphers), otherwise known as KASUMI or UEA1[9] published by the ETSI. However it is necessary to bring GSM equipment using the A5/1 or A5/2 algorithms to manufacturers so they can incorporate new encryption algorithms, and thus it will take time to replace the A5/1 and A5/2 in practice.

Once the encryption algorithm of GSM is broken, the attacker can intercept all unencrypted communications made by the victim's smartphone.

**Attacks based on Wi-Fi**

An attacker can try to eavesdrop on Wi-Fi communications to derive information (e.g. username, password). This type of attack is not unique to smartphones, but they are very vulnerable to these attacks because very often the Wi-Fi is the only means of communication they have to access the internet. The security of wireless networks (WLAN) is thus an important subject. Initially wireless networks were secured by WEP keys. The weakness of WEP is a short encryption key which is the same for all connected clients. In addition, several reductions in the search space of the keys have been found by researchers. Now, most wireless networks are protected by the WPA security protocol. WPA is based on the "Temporal Key Integrity Protocol (TKIP)" which was designed to allow migration from WEP to WPA on the equipment already deployed. The major improvements in security are the dynamic encryption keys. For small networks, the WPA is a "pre-shared key" which is based on a shared key. Encryption can be vulnerable if the length of the shared key is short. With limited opportunities for input (i.e. only the numeric keypad) mobile phone users might define short encryption keys that contain only numbers. This increases the likelihood that an attacker succeeds with a brute-force attack. The successor to WPA, called WPA2, is supposed to be safe enough to withstand a brute force attack.

As with GSM, if the attacker succeeds in breaking the identification key, it will be possible to attack not only the phone but also the entire network it is connected to.

Many smartphones for wireless LANs remember they are already connected, and this mechanism prevents the user from having to re-identify with each connection.
However, an attacker could create a WIFI access point twin with the same parameters and characteristics as the real network. Using the fact that some smartphones remember the networks, they could confuse the two networks and connect to the network of the attacker who can intercept data if it does not transmit its data in encrypted form.\[10\]

Lasco is a worm that initially infects a remote device using the SIS file format.\[11\] SIS file format (Software Installation Script) is a script file that can be executed by the system without user interaction. The smartphone thus believes the file to come from a trusted source and downloads it, infecting the machine.\[11\]

**Principle of Bluetooth-based attacks**

Security issues related to Bluetooth on mobile devices have been studied and have shown numerous problems on different phones. One easy to exploit vulnerability: unregistered services do not require authentication, and vulnerable applications have a virtual serial port used to control the phone. An attacker only needed to connect to the port to take full control of the device.\[12\] Another example: a phone must be within reach and Bluetooth in discovery mode. The attacker sends a file via Bluetooth. If the recipient accepts, a virus is transmitted. For example: Cabir is a worm that spreads via Bluetooth connection.\[6\] The worm searches for nearby phones with Bluetooth in discoverable mode and sends itself to the target device. The user must accept the incoming file and install the program. After installing, the worm infects the machine.

**Attacks based on vulnerabilities in software applications**

Other attacks are based on flaws in the OS or applications on the phone.

**Web Browser**

*See also: Browser security*

The mobile web browser is an emerging attack vector for mobile devices. Just as common Web browsers, mobile web browsers are extended from pure web navigation with widgets and plug-ins, or are completely native mobile browsers.

Jailbreaking the iPhone with firmware 1.1.1 was based entirely on vulnerabilities on the web browser.\[13\] As a result, the exploitation of the vulnerability described here underlines the importance of the Web browser as an attack vector for mobile devices. In this case, there was a vulnerability based on a stack-based buffer overflow in a library used by the web browser (Libtiff).

A vulnerability in the web browser for Android was discovered in October 2008.\[citation needed\] As the iPhone vulnerability above, it was due to an obsolete and vulnerable library. A significant difference with the iPhone vulnerability was Android's sandboxing architecture which limited the effects of this vulnerability to the Web browser process.
Smartphones are also victims of classic piracy related to the web: phishing, malicious websites, etc. The big difference is that smartphones do not yet have strong antivirus software available.\footnote{\textit{citation needed}}

**Operating System**

Sometimes it is possible to overcome the security safeguards by modifying the operating system itself. As real-world examples, this section covers the manipulation of firmware and malicious signature certificates. These attacks are difficult.

In 2004, vulnerabilities in virtual machines running on certain devices were revealed. It was possible to bypass the bytecode verifier and methods to access the native underlying operating system.\footnote{\textit{citation needed}} The results of this research were not published in detail. The firmware security of Nokia’s Symbian Platform Security Architecture (PSA) is based on a central configuration file called SWIPolicy. In 2008 it was possible to manipulate the Nokia firmware before it is installed, and in fact in some downloadable versions of it, this file was human readable, so it was possible to modify and change the image of the firmware.\footnote{14} This vulnerability has been solved by an update from Nokia.

In theory smartphones have an advantage over hard drives since the OS files are in ROM, and cannot be changed by malware. However in some systems it was possible to circumvent this: in the Symbian OS it was possible to overwrite a file with a file of the same name.\footnote{14} On the Windows OS, it was possible to change a pointer to a general configuration file to an editable file.

When an application is installed, the signing of this application is verified by a series of certificates. You can create a valid signature without using a valid certificate and add it to the list.\footnote{15} In the Symbian OS all certificates are in the directory: `c:\resource\swicertstore\dat`. With firmware changes explained above it is very easy to insert a seemingly valid but malicious certificate.

**Physical attacks**

In 2010, researcher from the University of Pennsylvania investigated the possibility of cracking a device’s password through a smudge attack (literally imaging the finger smudges on the screen to discern the user’s password).\footnote{16} The researchers were able to discern the device password up to 68% of the time under certain conditions.\footnote{16}

**Malicious Software (Malware)**

As smartphones are a permanent point of access to the internet (mostly on), they can be compromised as easily as computers with malware.\footnote{17} A malware is a computer program that aims to harm the system in which it resides. Trojans, worms and viruses are all considered malware. A Trojan is a program that is on the smartphone and allows external users to connect discreetly. A worm is a program that reproduces on multiple computers across a network. A virus is malicious software designed to spread to other computers by inserting itself into legitimate programs and running
programs in parallel. However, it must be said that the malware are far less numerous and important to smartphones as they are to computers.

The three phases of malware attacks

Typically an attack on a smartphone made by malware takes place in 3 phases: the infection of a host, the accomplishment of its goal, and the spread of the malware to other systems. Malware often use the resources offered by the infected smartphones. It will use the output devices such as Bluetooth or infrared, but it may also use the address book or email address of the person to infect the user's acquaintances. The malware exploits the trust that is given to data sent by an acquaintance.

Infection

Infection is the means used by the malware to get into the smartphone, it can either use one of the faults previously presented or may use the gullibility of the user. Infections are classified into four classes according to their degree of user interaction:  

Explicit permission
- the most benign interaction is to ask the user if it is allowed to infect the machine, clearly indicating its potential malicious behavior. This is typical behavior of a proof of concept malware.

Implied permission
- this infection is based on the fact that the user has a habit of installing software. Most trojans try to seduce the user into installing attractive applications (games, useful applications etc.) that actually contain malware.

Common interaction
- this infection is related to a common behavior, such as opening an MMS or email.

No interaction
the last class of infection is the most dangerous. Indeed, a worm that could infect a smartphone and could infect other smartphones without any interaction would be catastrophic. Fortunately, at this time there haven't been any examples.

**Accomplishment of its goal**

Once the malware has infected a phone it will also seek to accomplish its goal, which is usually one of the following: monetary damage, damage data and/or device, and concealed damage:[20]

Monetary damages
the attacker can steal user data and either sell them to the same user, or sell to a third party.

Damage
malware can partially damage the device, or delete or modify data on the device.

Concealed damage
the two aforementioned types of damage are detectable, but the malware can also leave a backdoor for future attacks or even conduct wiretaps.

**Spread to other systems**

Once the malware has infected a smartphone, it always aims to spread one way or another:[21]

- It can spread through proximate devices using Wi-Fi, Bluetooth and infrared;
- It can also spread using remote networks such as telephone calls or SMS or emails.

**Example of malware**

Here are various malware that exist in the world of smartphones with a short description of each.

**Viruses and Trojans**

- **Cabir** (also known as Caribe, SybmOS/Cabir, Symbian/Cabir and EPOC.cabir) is the name of a computer worm developed in 2004 that is designed to infect mobile phones running Symbian OS. It is believed to be the first computer worm that can infect mobile phones
- **Commwarrior**, found March 7, 2005, is the first worm that can infect many machines from MMS.[6] It is sent in the form of an archive file COMMWARRIOR.ZIP that contains a file COMMWARRIOR.SIS. When this file is executed, Commwarrior attempts to connect to nearby devices by Bluetooth or infrared under a random name. It then attempts to send MMS message to the contacts in the smartphone with different header messages for each person, who receive the MMS and often open them without further verification.
• **Phage** is the first Palm OS virus that was discovered.\[6\] It transfers to the Palm from a PC via synchronization. It infects all applications that are in the smartphone and it embeds its own code to function without the user and the system detecting it. All that the system will detect is that its usual applications are functioning.

• **RedBrowser** is a Trojan which is based on java.\[6\] The Trojan masquerades as a program called "RedBrowser" which allows the user to visit WAP sites without a WAP connection. During application installation, the user sees a request on his phone that the application needs permission to send messages. Therefore, if the user accepts, RedBrowser can send sms to paid call centers. This program uses the smartphone's connection to social networks (Facebook, Twitter, etc.) to get the contact information for the user's acquaintances (provided the required permissions have been given) and will send them messages.

• **WinCE.PmCryptic.A** is a malicious software on Windows Mobile which aims to earn money for its authors. It uses the infestation of memory cards that are inserted in the smartphone to spread more effectively.\[22\]

• **CardTrap** is a virus that is available on different types of smartphone, which aims to deactivate the system and third party applications. It works by replacing the file that is used to start the smartphone and that of each application need to get started to prevent them from executing.\[23\] There are different variants of this virus such as Cardtrap.A for SymbOS devices. It also infects the memory card with malware capable of infecting Windows.

**Spyware**

*Main article: Spyware*

• **Flexispy** is an application that can be considered as a trojan, based on Symbian. The program sends all information received and sent from the smartphone to a Flexispy server. It was originally created to protect children and spy on adulterous spouses.\[6\]

**Number of malware**

Below is a diagram which shows the different behaviors of smartphone malware in terms of their effects on smartphones.\[18\]
Effects of Malware

We can see from the graph that at least 50 malwares exhibit no negative behavior, except their ability to spread.\textsuperscript{[18]}

**Portability of malware across platforms**

There is a multitude of malware. This is partly due to the variety of operating systems on smartphones. However attackers can also choose to make their malware target multiple platforms, and malware can be found which attacks an OS but is able to spread to different systems.

To begin with, malware can use runtime environments like Java virtual machine or the .NET Framework. They can also use other libraries present in many operating systems.\textsuperscript{[24]} Other malware carry several executable files in order to run in multiple environments and they utilize these during the propagation process. In practice, this type of malware requires a connection between the two operating systems to use as an attack vector. Memory cards can be used for this purpose, or synchronization software can be used to propagate the virus.

**Countermeasures**
The security mechanisms in place to counter the threats described above are presented in this section. They are divided into different categories, as all do not act at the same level, and they range from the management of security by the operating system to the behavioral education of the user. The threats prevented by the various measures are not the same depending on the case. Considering the two cases mentioned above, in the first case one would protect the system from corruption by an application, and in the second case the installation of a suspicious software would be prevented.

**Security in operating systems**

The first layer of security within a smartphone is at the level of the operating system (OS). Beyond the usual roles of an operating system (e.g. resource management, scheduling processes), on a smartphone, it must also establish the protocols for introducing external applications and data without introducing risk. A central idea found in the mobile operating systems is the idea of a sandbox. Since smartphones are currently being designed to accommodate many applications, they must put in place mechanisms to ensure these facilities are safe for themselves, for other applications and data on the system, and the user. If a malicious program manages to reach a device, it is necessary that the vulnerable area presented by the system be as small as possible. Sandboxing extends this idea to compartmentalize different processes, preventing them from interacting and damaging each other. Based on the history of operating systems, sandboxing has different implementations. For example, where iOS will focus on limiting access to its API for applications from the App Store, Android bases its sandboxing on its legacy of Linux.

The following points highlight mechanisms implemented in operating systems, especially Android.

**Rootkit Detectors**

The intrusion of a rootkit in the system is a great danger in the same way as on a computer. It is important to prevent such intrusions, and to be able to detect them as often as possible. Indeed, there is concern that with this type of malicious program, the result could be a partial or complete bypass of the device security, and the acquisition of administrator rights by the attacker. If this happens, then nothing prevents the attacker to study or disable the safety features that were avoided, deploy the applications they want, or disseminate a method of intrusion by a rootkit to a wider audience.\[^{25}\][\(^{26}\) We can cite, as a defense mechanism, the Chain of trust in iOS. This mechanism relies on the signature of the different applications required to start the operating system, and a certificate signed by Apple. In the event that the signature checks are inconclusive, the device detects this and stops the boot-up.\[^{27}\] If Operating System is compromised due to Jailbreaking, root kit detection may not work if it is disabled by Jailbreak method or software loaded after Jailbreak disables Rootkit Detection.

**Process isolation**

Android uses mechanisms of user process isolation inherited from Linux. Each application has a user associated with it, and a tuple (UID, GID). This approach serves as a sandbox: while applications can be malicious, they can not get out of the sandbox reserved for them by their identifiers, and thus
cannot interfere with the proper functioning of the system. For example, since it is impossible for a process to end the process of another user, an application can thus not stop the execution of another.\textsuperscript{[25][28][29][30][31]}

File permissions
From the legacy of Linux, there are also filesystem permissions mechanisms. They help with sandboxing: a process can not edit any files it wants. It is therefore not possible to freely corrupt files necessary for the operation of another application or system. Furthermore, in Android there is the method of locking memory permissions. It is not possible to change the permissions of files installed on the SD card from the phone, and consequently it is impossible to install applications.\textsuperscript{[32][33][34]}

Memory Protection
In the same way as on a computer, memory protection prevents privilege escalation. Indeed, if a process managed to reach the area allocated to other processes, it could write in the memory of a process with rights superior to his own, with root in the worst case, and perform actions which are beyond its permissions on the system. It would suffice to insert function calls are authorized by the privileges of the malicious application.\textsuperscript{[31]}

Development through runtime environments
Software is often developed in high-level languages, which can control what is being done by a running program. For example, Java Virtual Machines continuously monitor the actions of the execution threads they manage, monitor and assign resources, and prevent malicious actions. Buffer overflows can be prevented by these controls.\textsuperscript{[35][36][31]}

Security Software
Above the operating system security, there is a layer of security software. This layer is composed of individual components to strengthen various vulnerabilities: prevent malware, intrusions, the identification of a user as a human, and user authentication. It contains software components that have learned from their experience with computer security; however, on smartphones, this software must deal with greater constraints (see limitations).

Antivirus and firewall
An antivirus software can be deployed on a device to verify that it is not infected by a known threat, usually by signature detection software that detects malicious executable files. A firewall, meanwhile, can watch over the existing traffic on the network and ensure that a malicious application does not seek to communicate through it. It may equally verify that an installed application does not seek to establish suspicious communication, which may prevent an intrusion attempt.\textsuperscript{[37][38][39][26]}

Visual Notifications
In order to make the user aware of any abnormal actions, such as a call he did not initiate, one can link some functions to a visual notification that is impossible to circumvent. For example, when a call is triggered, the called number should always be displayed. Thus, if a call is triggered by a malicious application, the user can see, and take appropriate action.

Turing Test
In the same vein as above, it is important to confirm certain actions by a user decision. The Turing test is used to distinguish between a human and a virtual user, and it often comes as a captcha. It is theoretically impossible for a computer to solve such a test, and therefore suspicious activities may be subject to approval or denial by the user.\[40\]

Biometric Identification
Another method is to use is biometrics.\[41\] Biometrics is a technique of identifying a person by means of her morphology (by recognition of the eye or face, for example) or her behavior (her signature or way of writing for example). One advantage of using biometric security is that users can avoid remembering a password or other secret combination to authenticate and prevent malicious users to access their device. In a system with strong biometric security, only the primary user can access the smartphone.

Resource Monitoring in the smartphone
When an application passes the various security barriers, it can take the actions for which it was designed. When such actions are triggered, the activity of a malicious application can be sometimes detected if one monitors the various resources used on the phone. Depending on the goals of the malware, the consequences of infection are not always the same; all malicious applications are not intended to harm the devices on which they are deployed. The following sections describe different ways to detect suspicious activity.\[42\]

Battery
Some malware is aimed at exhausting the energy resources of the phone. Monitoring the energy consumption of the phone can be a way to detect certain malware applications.\[25\]

Memory Usage
Memory usage is inherent in any application. However, if one finds that a substantial proportion of memory is used by an application, it may be flagged as suspicious.

Network Traffic
On a smartphone, many applications are bound to connect via the network, as part of their normal operation. However, an application using a lot of bandwidth can be strongly suspected of attempting to communicate a lot of information, and disseminate data to many other devices. This observation only allows a suspicion, because some legitimate applications can be very resource-intensive in terms of network communications, the best example being streaming video.

Services
One can monitor the activity of various services of a smartphone. During certain moments, some services should not be active, and if one is detected, the application should be suspected. For example, the sending of an SMS when the user is filming video: this communication does not make sense and is suspicious; malware may attempt to send SMS while its activity is masked.\[43\]
The various points mentioned above are only indications and do not provide certainty about the legitimacy of the activity of an application. However, these criteria can help target suspicious applications, especially if several criteria are combined.

Network surveillance

Network traffic exchanged by phones can be monitored. One can place safeguards in network routing points in order to detect abnormal behavior. As the mobile's use of network protocols is much more constrained than that of a computer, expected network data streams can be predicted (e.g. the protocol for sending an SMS), which permits detection of anomalies in mobile networks.

Spam Filters

As is the case with email exchanges, we can detect a spam campaign through means of mobile communications (SMS, MMS). It is therefore possible to detect and minimize this kind of attempt by filters deployed on network infrastructure that is relaying these messages.

Encryption of stored or transmitted information

Because it is always possible that data exchanged can be intercepted, communications, or even information storage, can rely on encryption to prevent a malicious entity from using any data obtained during communications. However, this poses the problem of key exchange for encryption algorithms, which requires a secure channel.

Telecom Network Monitoring

The networks for SMS and MMS exhibit predictable behavior, and there is not as much liberty compared what you can do with protocols such as TCP or UDP. This implies that one can not predict the use made of the common protocols of the web; one might generate very little traffic by consulting simple pages, rarely, or generate heavy traffic by using video streaming. On the other hand, messages exchanged via mobile phone have a framework and a specific model, and the user does not, in a normal case, have the freedom to intervene in the details of these communications. Therefore, if an abnormality is found in the flux of network data in the mobile networks, the potential threat can be quickly detected.

Manufacturer's surveillance

In the production and distribution chain for mobile devices, it is the responsibility of manufacturers to ensure that devices are delivered in a basic configuration without vulnerabilities. Most users are not experts and many of them are not aware of the existence of security vulnerabilities, so the device configuration as provided by manufacturers will be retained by many users. Below are listed several points which manufacturers should consider.

Remove debug mode

Phones are sometimes set in a debug mode during manufacturing, but this mode must be disabled before the phone is sold. This mode allows access to different features, not intended for routine use by a user. Due to the speed of development and production, distractions occur and some devices are sold in
debug mode. This kind of deployment exposes mobile devices to exploits that utilize this oversight.[44][45]

Default settings
When a smartphone is sold, its default settings must be correct, and not leave security gaps. The default configuration is not always changed, so a good initial setup is essential for users. There are, for example, default configurations that are vulnerable to denial of service attacks.[25][46]

Security audit of apps
Along with smart phones, appstores have emerged. A user finds herself facing a huge range of applications. This is especially true for providers who manage appstores because they are tasked with examining the apps provided, from different points of view (e.g. security, content). The security audit should be particularly cautious, because if a fault is not detected, the application can spread very quickly within a few days, and infect a significant number of devices.[25]

Detect suspicious applications demanding rights
When installing applications, it is good to warn the user against sets of permissions that, grouped together, seem potentially dangerous, or at least suspicious. Frameworks like such as Kirin, on Android, attempt to detect and prohibit certain sets of permissions.[47]

Revocation procedures
Along with appstores appeared a new feature for mobile apps: remote revocation. First developed by Android, this procedure can remotely and globally uninstall an application, on any device that has it. This means the spread of a malicious application that managed to evade security checks can be immediately stopped when the threat is discovered.[48][49]

Avoid heavily customized systems
Manufacturers are tempted to overlay custom layers on existing operating systems, with the dual purpose of offering customized options and disabling or charging for certain features. This has the dual effect of risking the introduction of new bugs in the system, coupled with an incentive for users to modify the systems to circumvent the manufacturer's restrictions. These systems are rarely as stable and reliable as the original, and may suffer from phishing attempts or other exploits.[citation needed]

Improve software patch processes
New versions of various software components of a smartphone, including operating systems, are regularly published. They correct many flaws over time. Nevertheless, manufacturers often do not deploy these updates to their devices in a timely fashion, and sometimes not at all. Thus, vulnerabilities persist when they could be corrected, and if they are not, since they are known, they are easily exploitable.[47]

User awareness

Much malicious behavior is allowed by the carelessness of the user. From simply not leaving the device without a password, to precise control of permissions granted to applications added to the smartphone, the user has a large responsibility in the cycle of security: to not be the vector of intrusion. This precaution is especially important if the user is an employee of a company that stores business data on the device. Detailed below are some precautions that a user can take manage security on a smartphone.
Being skeptical

A user should not believe everything that may be presented, as some information may be phishing or attempting to distribute a malicious application. It is therefore advisable to check the reputation of the application that you want to buy before actually installing it.\textsuperscript{[50]}

Permissions given to applications

The mass distribution of applications is accompanied by the establishment of different permissions mechanisms for each operating systems. It is necessary to clarify these permissions mechanisms to users, as they differ from one system to another, and are not always easy to understand. In addition, it is rarely possible to modify a set of permissions requested by an application if the number of permissions is too great. But this last point is a source of risk because a user can grant rights to an application, far beyond the rights it needs. For example, a note taking application does not require access to the geolocation service. The user must ensure the privileges required by an application during installation and should not accept the installation if requested rights are inconsistent.\textsuperscript{[51][46][52]}

Be careful

Protection of a user's phone through simple gestures and precautions, such as locking the smartphone when it is not in use, not leaving their device unattended, not trusting applications, not storing sensitive data, or encrypting sensitive data that can not be separated from the device.\textsuperscript{[53][54]}

Ensure data

Smartphones have a significant memory and can carry several gigabytes of data. The user must be careful about what data it carries and whether they should be protected. While it is usually not dramatic if a song is copied, a file containing bank information or business data can be more risky. The user must have the prudence to avoid the transmission of sensitive data on a smartphone, which can be easily stolen. Furthermore, when a user gets rid of a device, she must be sure to remove all personal data first.\textsuperscript{[55]}

These precautions are measures that leave no easy solution to the intrusion of people or malicious applications in a smartphone. If users are careful, many attacks can be defeated, especially phishing and applications seeking only to obtain rights on a device.

Limitations of certain security measures

The security mechanisms mentioned in this article are to a large extent inherited from the knowledge and experience with computer security. The elements composing the two device types are similar, and there are common measures that can be used, such as antivirus and firewall. However, the implementation of these solutions is not necessarily possible or at least highly constrained within a mobile device. The reason for this difference is the differing technical resources offered by computers and mobile devices: even though the computing power of smartphones is becoming faster, they have other limitations than their computing power.

- **Single-task system:** some operating systems, including some still commonly used, are single-tasking. Only the foreground task is executed. It is difficult to introduce applications such as antivirus and firewall on such systems, because
they could not perform their monitoring while the user is using the device, when there would be most need of such monitoring.

- **Energy autonomy**: a critical one for the use of a smartphone is energy autonomy. It is important that the security mechanisms not consume battery resources, without which the autonomy of devices will be affected dramatically, undermining the effective use of the smartphone.

- **Network** directly related to battery life, network utilization should not be too high. It is indeed one of the most expensive resources, from the point of view of energy consumption. Nonetheless, some calculations may need to be relocated to remote servers in order to preserve the battery. This balance can make implementation of certain intensive computation mechanisms a delicate proposition.\[56\]

Furthermore, it should be noted that it is common to find that updates exist, or can be developed or deployed, but this is not always done. One can for example find a user who does not know that there is a newer version of the operating system compatible with the smartphone, or discover known vulnerabilities that are not corrected before long periods in a development cycles, which allow periods exploit loopholes updates.\[45\]