**Mobile Security Threats**

Web-Based Mobile Security Threats. Web-based threats are subtle and tend to go unnoticed. They happen when people visit affected sites that seem fine on the front-end but, in reality, automatically download malicious content onto devices.  
Mobile Network Security Threats. Network-based threats are especially common and risky because cybercriminals can steal unencrypted data while people use public WiFi networks.  
Mobile Device Security Threats. Physical threats to mobile devices most commonly refer to the loss or theft of a device. Because hackers have direct access to the hardware where private data is stored, this threat is especially dangerous to enterprises.

**Why is mobile security important?**

Nowadays nearly all the tasks that you could only perform on a computer are achievable on mobile devices as well.  
more sensitive information will be stored on peoples' mobile devices than before. Employees are even able to do work on their mobile devices.  
more risks for proprietary information leaks as well.  
Additionally, the number of attempts of cybercrime has been increasing  
steadily in the recent years.  
This is even more important for Android because it is the most targeted  
platform due to its widespread usage and open source properties.  
The need for security is greater than ever for not only consumers, but large enterprises as well.

**Security Philosophy**

Finite time and resources: Humans have difficulty understanding risk.

Safer to assume thatmost developers do not understand securityand most users do not understand security.

Diagram

Description automatically generated

**Security philosophy cornerstones**

* Need to prevent security breaches from occurring
* Need to minimize the impact of a security breach
* Need to detect vulnerabilities and security breaches
* Need to react to vulnerabilities and security breaches swiftly

**Security flaws in Android**

The security flaws allow hackers to exfiltrate media files such as photos, videos, and call-recordings. Additionally, criminals could access real-time data, such as microphone data, GPS, and location data. The attackers could also freeze the affected Android phones, rendering them unresponsive.

**Security Model in Android**

Android is fundamentally based on a multi-party consent-model: An action should only happen if all involved parties consent to it.  
If any party does not consent, then the safe-by-default choice is for that action to be blocked.  
This is different to the security models that more traditional operating systems implement, which are focused on user access control and do not explicitly consider other stakeholders.

**Devices built on top of AOSP (Android Open-Source Project)**

Android Google Mobile Services-certified devices  
Ex. Pixel, Samsung, devices that use the "Android" name Includes Google apps (GMail, Maps, GMSCore, etc.) Build images must be approved by Google (series of test suites) Devices built on AOSP Ex. Amazon Fire tablets Doesn't include Google apps. "Android-compatible" means the device complies with the Android Compatibility Definition Document (CDD)

**Hardware Abstraction Layer Attacks**

In Broadcom Company's Wi-Fi chipset, which is used in most of the Android and iOS  
devices, a highly dangerous vulnerability was exposed which could allow the attacker to get the control of the complete system by using Booby-Trapped Signals.  
Attackers can also use applications to spy on the user's data using Audio Channels,  
GPS, Camera and other components as Google Play Services that offers some unsafe features to the applications which can be exploited by the intruders.  
If user disables these features, it affects the normal functioning of the application.  
Malicious applications installed in to the system can also use the Audio channels to steal the password. Such applications send deceitful commands through speaker of the victim's system to its microphone which behaves as confused deputy.  
TalkBack Accessibility Service of Android is used to accomplish the purpose of this attack, as it reads out the password typed by the victim to the attacker.

**Application Based Attacks**

Applications request for the privileges from the system at the time  
of installation for smoothly running later on.  
It has been found that the Applications which are installed from  
websites request for unnecessary permissions which are not required  
for performing their job. On Android, there are no restrictions on how an application  
can be written that are required to enforce security; in this respect, native code is sandboxed as interpreted code.  
The attackers are misusing the flaws in native libraries and the third  
party libraries found in the Applications; they use these libraries to  
request for combined permissions.

**Motivation for Malware Developers**

Dark Web: A marketplace for criminals to buy and sell mobile malware, which are often sold as component of software packages.

Permission System Exploitation: Permission System forms the important component of Android security mechanism. it restricts the applications from accessing users personal information.

Malware developers are taking advantage of weaknesses of not only Permission System but also exploiting Android Infrastructure susceptibility to attacks. Users simply accept permissions to be able to use the application; giving knowingly or unknowingly consent to the Attackers to get into the system. After crossing this checkpoint malicious application can gain root access to the system by exploiting kernel level vulnerabilities and can initiate Privilege escalation attacks

Lack of Awareness Regarding Security Protocols and Poor Developing Practices: Android developers while developing the application do not give much emphasis on secure communication of data. Using SSL certificates helps in securing the data from intruders. Because of the high cost involved in debugging android applications using SSL certificates, developers prefer using development servers consisting of unauthorized certificates for debugging. If SSL/TLS code is written incorrectly or steps for establishing the secured connection are not followed carefully it can lead to cryptographic attacks.

Lack of Documentation: Lack of documentation by developers can be misleading for the users and for other developers who are not directly involved in coding but with different modules of application development.