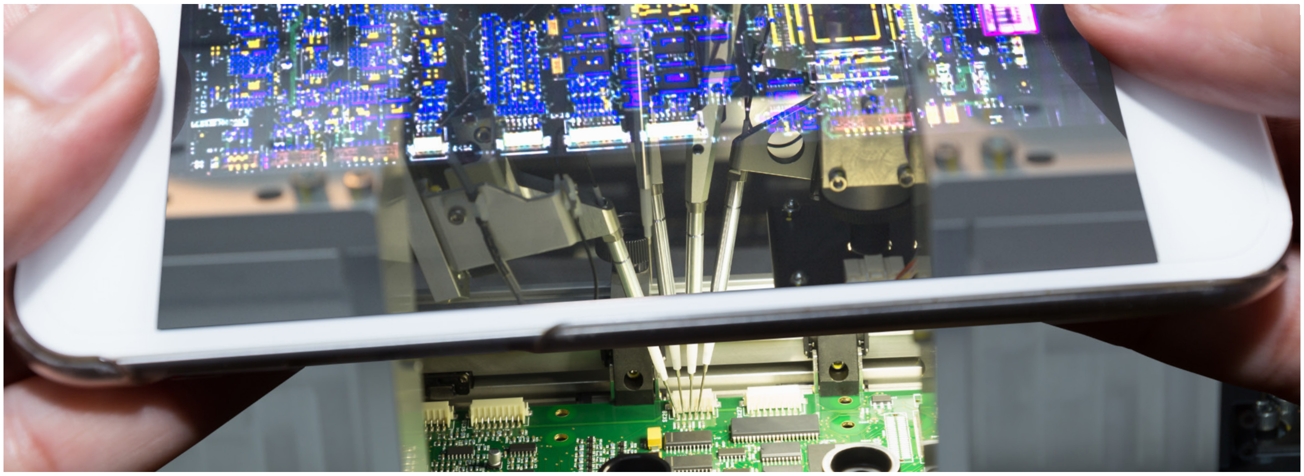
RETHINKING MOBILE TRUST

PRIVORO®

As smartphones have evolved from consumer novelties into essential tools for living and working, hackers have turned these devices into key targets of attack. In turn, makers of smartphones have implemented new ways of establishing and maintaining trust at the hardware level. But inherent limitations in the mobile architectures of these devices – especially in a world of chip-based attacks – mean that organizations must find new ways of building mobile trust.



HARNESSING THE MOBILE ECOSYSTEM

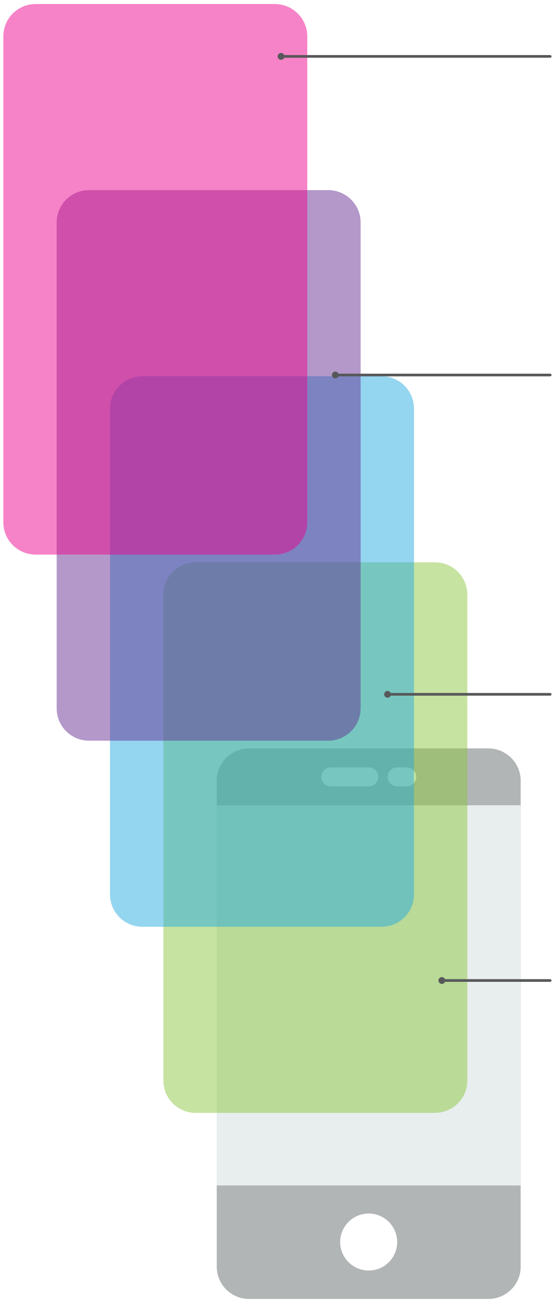
The creation of a smartphone is a delicate balancing act between a multitude of parties from across the globe, each responsible for its own sliver of functionality:

- A software engineering team develops the operating system (OS).
- A semiconductor manufacturer designs and fabricates the system on a chip (SoC).
- Hundreds of suppliers provide the hardware components for the printed circuit board assembly (PCBA).
- An electronics manufacturer assembles the device.
- A cellular provider establishes carrier settings.
- Millions of developers create apps for the public app store.

The end result is a complex mobile ecosystem, with each piece bringing its own exposure. Smartphones give threat actors a variety of entry points, from SMS phishing attacks to malicious apps. From there, any vulnerability in any component at any layer of the stack can potentially be exploited, from the individual apps at the top to the chips at the bottom. The lower in the stack a hacker is able to go, the more control is given over the rest of the mobile device.

Indeed, real-world examples abound of smartphones being attacked at every layer. As mobile devices have surpassed traditional computers as the dominant mode of computing, malicious actors have increasingly focused their efforts on these devices. The smartphone's huge attack surface – exacerbated by issues like poor implementations, insufficient security mitigations, fragmented responsibilities, supply-chain issues and delayed patching – gives hackers virtually unlimited ways to gain illicit access. From there, hackers can use tools like rootkits and remote access Trojans (RATs) to control these devices and siphon their data.

POTENTIAL SMARTPHONE VULNERABILITIES



APPS

Components

- Third-party libraries
- Data
- Permissions
- Exposed services

Key threats

- Poor authentication practices
- Multi-stage malware delivery

OPERATING SYSTEM

Components

- App sandbox
- Kernel
- Media services
- Runtime environment

Key threats

- Unpatched OS vulnerabilities
- Rootkits

FIRMWARE

Components

- Initialization code
- Bootloader
- Device drivers

Key threats

- Firmware bugs
- Supply-chain attacks

HARDWARE

Components

- Applicator processor and memory
- Baseband processor and memory
- Security module
- Subscriber identity module (SIM) card
- Peripherals

Key threats

- Speculative execution vulnerabilities
- Physical memory defects

BUILDING TRUST AT THE HARDWARE LEVEL

With so many opportunities for exploitation, smartphone makers have started building trust into the hardware at the very foundation of these devices through the use of a hardware root of trust and a trusted execution environment. Hardware-based trust is designed to enable confidence in every other layer of the stack by providing a degree of assurance that the smartphone's foundation hasn't been compromised by low-level attacks.

Hardware root of trust

A hardware root of trust (HRoT) is a set of security primitives – typically initialization code stored in read-only memory (ROM) and a unique public key based on the device's hardware identifier – providing a hardware-based, unalterable, cryptographically secure basis of trust to be leveraged by the rest of the device. Given its importance, an HRoT is typically safeguarded through trusted supply-chain processes and tamper protections.

The chain of trust manifests itself in the startup process, where security checks at each step in the process – stemming from the HRoT – validate the relying code.

After successfully booting, the HRoT may be leveraged to validate software/firmware during runtime. Tasks may include verifying the digital signatures associated with software (and creating assertions based on the results), measuring the integrity of software, managing software updates and more.

CHAIN OF TRUST

POWER ON

When the user powers on the smartphone, the application processor immediately executes the initialization code stored in read-only memory (ROM).

INITIALIZATION CODE (HARDWARE)

The initialization code, laid down in silicon during chip fabrication, uses the device's protected public key to verify that the bootloader is signed by the manufacturer before allowing it to load.

BOOTLOADER (FIRMWARE)

The bootloader cryptographically validates that each piece of firmware has been digitally signed (and is therefore unmodified by any low-level malware below the operating system, like rootkits). When finished, the bootloader verifies and runs the operating system (OS) kernel.

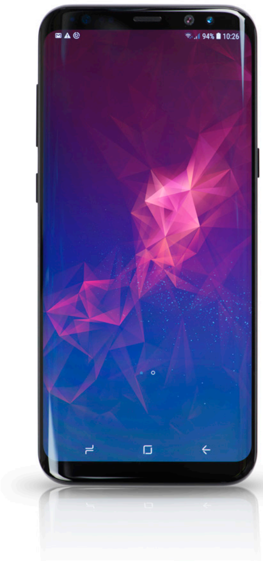
KERNEL (OPERATING SYSTEM)

The operating system ensures that all apps are digitally signed before allowing the user to run them.

APPS

TRUSTED EXECUTION ENVIRONMENT

A trusted execution environment (TEE) is an isolated execution environment that runs independently of the main, user-facing OS. Within a TEE, security-critical capabilities – such as storing cryptographic keys or running sensitive processes – are performed. Approaches for establishing a TEE vary between platforms and even within the same platform.



Android

Most Android smartphones offer some version of ARM's TrustZone technology, typically the Qualcomm Secure Execution Environment (QSEE) or Trustonic's Kinibi. TrustZone consists of two virtual processors: a "secure" world for the security subsystem and a "non-secure" world for everything else. Both virtual worlds typically run from the core processor, with hardware logic providing the separation between them. While implementations of TrustZone vary widely between different Android devices, the secure world is usually used to protect cryptographic keys and authentication mechanisms.

iPhone

Since 2013, Apple has included the Secure Enclave in all of its smartphones. The Secure Enclave is a coprocessor that's isolated from the main processor (but located on the same SoC), booting separately from the rest of the device and running its own microkernel. The main purpose of the coprocessor is to generate the device's Unique ID (UID) number and keep it segregated from the rest of iOS. Private keys are created, stored and used in Secure Enclave; other functions never handle these keys, only receiving the output of the cryptographic operations.

THE LIMITATIONS OF SMARTPHONE TRUST

Shared processing

Despite the advances in hardware-based trust from smartphone vendors, mobile devices are still multi-purpose consumer products at the end of the day. Most vendors have a mandate to fit as many components as possible into the thinnest form factor while keeping the end product affordable for most consumers. In practice, this means that critical security functions – like an HRoT and TEE – are relegated to the same application processor or SoC running non-secure software, including the user's myriad apps. Given the host of vulnerabilities affecting any smartphone at any time, attackers have a plethora of tools available to exploit these critical functions.

While vulnerabilities affecting the initialization code and bootloader have been exceedingly rare, exploits targeting various TEEs have successfully leveraged vendors' lack of mitigations. There have been numerous TEE exploits published, perhaps the most popular of which is the May 2016 discovery of a TEE vulnerability affecting about 60% of all Android smartphones. A flaw in the secure world's OS running on Qualcomm's QSEE allows an attacker running code in the non-secure world to exploit an application within the TEE, eventually gaining complete control over the entire device.

Chip-based vulnerabilities

As if smartphone vulnerabilities in the upper layers of the stack weren't bad enough, an emerging series of chip-based vulnerabilities affecting nearly every type of processor in every commercial device are poised to shatter existing security models at their core. These types of vulnerabilities threaten the isolation of hardware-based security measures, essentially putting control of the entire device in play. Chip-based vulnerabilities are particularly worrisome because they're virtually impossible to detect with existing solutions and because remediation often requires changes to the affected hardware.

Since the public disclosures of Meltdown and Spectre in January 2018, security researchers have placed more focus on these types of vulnerabilities, leading to regular discoveries of new chip flaws and variants of existing flaws. It's likely that these discoveries will continue for years to come, especially considering the lengthy development cycle for new chip architectures.

MAJOR TYPES OF CHIP-BASED VULNERABILITIES

Speculative execution

Flaws in a processor's speculative execution – in which tasks are performed based on anticipated results as a way of preventing delays – allow a rogue process to access the memory of apps and the OS.

Examples:
Meltdown (revealed January 2018)
BranchScope (revealed March 2018)

Physical memory defects

Variations of the Rowhammer attack – whereby bits can be flipped by accessing specific memory blocks inside a chip thousands of times per second – enable an attacker to alter crucial pieces of data.

Examples:
Drammer (revealed October 2016)
GLitch (revealed May 2018)

Firmware bugs

Flaws in the design and implementation of the firmware that is shipped with chipsets – typically errors in the code or a lack of security mitigations – provide an entry point for attack and privilege escalation.

Examples:
QuadRooter (revealed August 2016)
Broadpwn (revealed July 2017)

CLOSING THE MOBILE HARDWARE GAP

The looming wave of chip-based attacks puts enterprises and government agencies in a tricky situation, forced to decide whether the productivity gains enabled by smartphones are worth the countless risks of continued use.

The Privoro platform provides organizations with an alternative for trusted mobile computing: the SafeCase. An external, high-security source of trust, the SafeCase surrounds – but is functionally independent of – a user's mobile device. Like modern smartphones, the SafeCase has its own HRoT. Unlike smartphones, however, SafeCase has a number of architectural features that protect it from the threat of known and unknown chip-based vulnerabilities, including a closed-loop communication paradigm and a restricted processing schema that only allows interaction with approved, vetted and signed software. Security is central to every aspect of the case's design and manufacturing, with embedded protections for the supply chain, the provisioning process and the hardware around the chip itself. Even when the smartphone has been compromised, the SafeCase builds a baseline of trust for the broader system upon which secure services can be built.

PRIVORO®