Android Security 2017 Year In Review

March 2018

android

Contents

Overview

Smartphones and other connected devices improve lives around the world every day. People depend on connected devices to exchange messages, navigate from here to there, and take lots—and lots—of photos.

With more than 2 billion active Android devices, it's essential that Google provides the best protections for users at scale. We are committed to protecting users' privacy and security across different device types, such as smartphones, automobiles, wearables, TV, Things, and more.

> Android security made a significant leap forward in 2017 and many of our protections now lead the industry.

We measure our improvement based on our own data about the Android ecosystem. We look at metrics, such as how many devices have installed Potentially Harmful Applications (PHAs), what protections they have in place, where PHAs are coming from, as well as third-party analysis and industry signals.

Third-party data also pointed to improved overall security. Platform exploitation difficulty, as measured by vulnerability rewards program payouts, independent security researcher analyses, and premier security vulnerability contest results, signaled that Android's protections have become significantly stronger. Exploit pricing is correlated to attacker cost, which is determined by many factors, including time, people, expertise, product knowledge, product accessibility, specialized equipment, and money. Growth in exploit pricing and difficulty demonstrates that Android has achieved a strength of protection that now leads the industry.

This is Google's fourth annual report on Android security. The report details improvements to Google's security offerings for Android, new and updated

Android platform features, metrics that informed our view of Android security, and security trends for Android devices in 2017.

In 2017, we improved Android security in a variety of ways, such as reducing the number of PHAs on devices and in Google Play, improving security visibility and control for users with Google Play Protect, and reducing vulnerability exploitation with faster security updates. To make these changes, we collaborated closely with device manufacturers, system on a chip (SoC) vendors, telecom carriers, and Android researchers and academics.

We hope that sharing this information gives you more insight into the state of security in Android, and our constant efforts to keep our users and their data secure.

The best of Google security for Android

> Android devices that only download apps from Google Play are 9 times less likely to get a PHA than devices that download apps from other sources.

While all Android devices benefit from protections built into the platform, Android devices with Google Play services have an additional layer of defense to keep them safe. Google protects these devices right out of the box with Google Play Protect, our built-in device, data, and apps security scanning technology.

> With nearly 2 billion protected devices, Google Play Protect is the most widely deployed mobile threat protection service in the world.

Google Play Protect includes on-device capabilities that protect users from PHAs in real-time and cloud-based services that analyze device and app data to identify possible security concerns. Google is constantly improving our tools and methods, applying new machine learning techniques, and updating our detection and response systems to protect against new vulnerabilities and PHAs. Because Google Play Protect doesn't rely on firmware or platform updates, Android devices benefit from our innovation right away.

> Google Play Protect gives a visible home to all the security protections that have kept Android users and devices safe behind the scenes for years.

These protections include ways to find a lost device, safeguarding from deceptive websites, and systems that detect and remove PHAs—no matter where the questionable apps came from. Google Play Protect also helps users check the security state of their Android device, providing peace of mind that their device is secure.

We leverage machine and human intelligence to get the job done and keep our users safe. Our automated systems detect and classify PHAs and compare behavior to make meaningful connections across billions of data points. Our security experts analyze these findings and check suspected PHAs that our systems discover.

In 2016, the annual probability that a user downloaded a PHA from Google Play was .04% and we reduced that by 50% in 2017 for an annual average of .02%.

> In 2017, downloading a PHA from Google Play was less likely than the odds of an asteroid hitting the earth.

Up-to-date platform security

Google's protections are a real-time shield against PHAs, and those protections sit on top of core security smarts that are built directly into Android. All Android devices share a common, platform-level security model. We've enhanced this model over the years with SELinux protections, app isolation using sandboxing, exploit mitigations, and cryptographic features, like file-based encryption and Verified Boot.

In 2017, we expanded platform-level security in Android Oreo. Android Oreo increases security by making devices easier to update with Project Treble, giving apps a way to verify Android devices, reducing privilege, and mitigating sophisticated attacking techniques.

Google works closely with our device manufacturing, SoC, and carrier partners to bring the best of Android security to all devices. On top of that, the breadth

and depth of Android's ecosystem—with over 60,000 different device models—makes exploitation harder by limiting the impact of a mobile vulnerability and making it more complex to develop successful attacks. We provide compatibility resources, such as a detailed series of security requirements and a testing framework to ensure support across the diverse device ecosystem. In 2017, we also extended our security checks to proactively identify and remove preinstalled PHAs on Android devices.

In addition to our proactive compatibility resources, we work with our partners to keep Android device security up-to-date. In 2017, we improved our collaborative security maintenance programs and provided faster and easier updates across all Android devices.

> In 2017, we increased the number of Android devices that received security patches by more than 30%.

Protective power of open

One of Android's most important security strengths is its open source development approach.

As Android security has matured, it has become more difficult and expensive for attackers to find high severity exploits. This is where open source really shines. As a global, open source project, Android has a community of defenders collaboratively locating the deeper vulnerabilities and developing mitigations. This community may be orders of magnitude larger and more effective than a closed source project of similar scale. Android's defenders come from thousands of device manufacturers, SOC vendors, carriers, academic institutions, independent security researchers, and the worldwide Linux community.

> As of 2017, Google's Android Security Rewards program offers one of the highest reward values in the industry.

Another example of exploit pricing is Mobile Pwn2Own, the premier mobile hacking contest where security experts from around the world compete to find exploits in mobile devices. Mobile Pwn2Own 2017 reward values were comparable across operating system platforms. In addition, the contest did not reward any core Android platform security exploits.

Enterprise growth

Secure devices engender greater user and business confidence. Many enterprises have stringent device security requirements, so enterprise adoption and analyst reports help gauge the positive impact of Android security improvements.

For example, Gartner's December 2017 Mobile OSs and Device Security: A Comparison of Platforms report by Patrick Hevesi reviewed Android among security controls for mobile devices. These controls included iOS 9, iOS 10, iOS 11, Android 4, Android 5, Android 6, Android 7, Android 8, Samsung Knox 2.6, Samsung Knox 2.9, Google Pixel (Android 7), Google Pixel 2 (Android 8), Windows 10, and Microsoft Surface Pro.

Google continued to invest in Android's enterprise security features in 2017. One of Android's primary enterprise security capabilities is the work profile, which separates business apps and data from personal apps and data. Work profiles enable privacy for users (the business can't see the apps, data, or activity on the personal side) and improve data and network security for the business. In 2017, we established a validation process to ensure consistent, quality implementations of work profile, managed device, and dedicated device solution sets with nearly 40 Enterprise Mobility Management (EMM) solution providers. We also released managed Google Play, a curated Google Play store for enterprise customers.

> In 2017, the number of 30-day active devices running managed Google Play increased by 2000%.

We also launched the Verify Apps API, which helps administrators determine whether a device is protected by Google Play Protect, encourage users to enable Google Play Protect, and identify any PHAs that are installed on the device.

As we celebrate Android security's successes in 2017, we are far from content. We look forward to eradicating more PHAs, further enhancing privacy and security in future Android releases, and providing the most up-to-date security features across Android devices. We are deeply grateful to our ecosystem partners, developers, researchers, and the rest of the global Android community for helping to protect Android devices and users.

Google Play Protect

Google has long contributed to the security of Android devices with multiple layers of on-device and cloud-based technologies. All devices with Google Play have a set of endpoint and mobile threat protection services that protect against common threats, including network attacks, app exploits, potentially harmful applications (PHAs), and physical attacks, such as device theft.

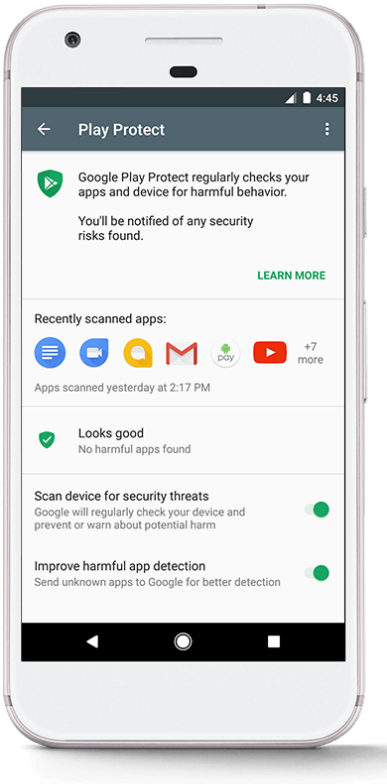
In 2017, these protections evolved to form Google Play Protect, which provides a visible home for Google's comprehensive security protections for Android. While Google Play Protect's core features have been part of Android for years, we added several features that better identify and address mobile threats in 2017, which we'll cover below.

Google Play Protect leverages the technical talent of security experts, app analysis, response tools, and machine learning advancements to detect PHAs. It also presents device security information in Settings and Google Play, providing users with comfort, ease, and control over their device's security.

> Google Play Protect is enabled on over 2 billion devices running Android 4.3+ with Google Play, and constantly works in the background to keep users' devices and data safe.

Google Play Protect regularly updates across all devices to remove new threats; it doesn't rely on releases or Over the Air updates (OTAs) to improve.

Google Play Protect in Settings



On-device protections

This table lists Google Play Protect's on-device capabilities with a brief description of how they help keep devices and data safe. Most of these services integrate with a cloud-based component that allows Google to push updates.

The following sections explain how these on-device protections work and details new features and improvements made in 2017.

Service	Protection
PHA scanning	Collection of mobile threat protections and removal options for downloaded PHAs including: — Automatic daily PHA scanning — User-initiated, on-demand scanning — Scanning for threats even when device is offline — Automatically disabling or removing PHA threats — Uploading new apps to the cloud for scanning
Find My Device	Protection for lost or stolen devices (Formerly Android Device Manager)
Safe Browsing	Protection from deceptive websites
Developer APIs	APIs that allow third-party apps to use Google's security services

PHA scanning services

Google Play Protect leverage cloud-based app-verification services to determine if apps are potentially harmful. Google Play Protect scans Android devices for evidence of PHAs. If it finds a PHA, Google Play Protect warns the user and can disable or remove particularly bad PHAs.

Daily PHA scan
Since 2014, Google Play Protect's Verify Apps service runs a periodic full-device scan that looks at apps before installation and runs regular scans on all installed apps. If a PHA is found, a notification asks the user to remove it. In cases where the PHA has no possible benefit to users, Google Play Protect can remove the PHA from affected devices and block future installs.

We have always scanned devices for PHAs about once every 6 days. (Devices that had indicators of installed PHAs or other risk factors were scanned more frequently.) In 2016, we started scanning all devices for PHAs once a day. Daily scanning allows Google Play Protect to respond quickly to a detected threat, reducing how long users could be exposed to the threat and how many devices may be affected. To conserve data, these daily scans only contact Google servers to request verification when a suspected PHA is detected.

> In 2017, daily scans led to faster identification and removal of approximately 39 million PHAs.

Though Google Play Protect works in the background, users can check when their device was last scanned and the list of scanned apps in the Google Play Protect section of their Google Play app.

On-demand PHA scan

In addition to a lightweight, daily, automatic scan, users can start a full-device scan at any time. Upon request, the device contacts Google servers for the latest information and scans all apps on the device. If a harmful app is discovered, Google Play Protect notifies the user to take action or takes action on their behalf. This visibility gives users peace of mind that they have the latest protection at all times.

Offline PHA scan

In early 2017, we investigated more PHA install patterns. Our research showed that about 35% of new PHA installations occurred when the device was offline or had lost network connectivity before Google Play Protect could determine if an app was a PHA.

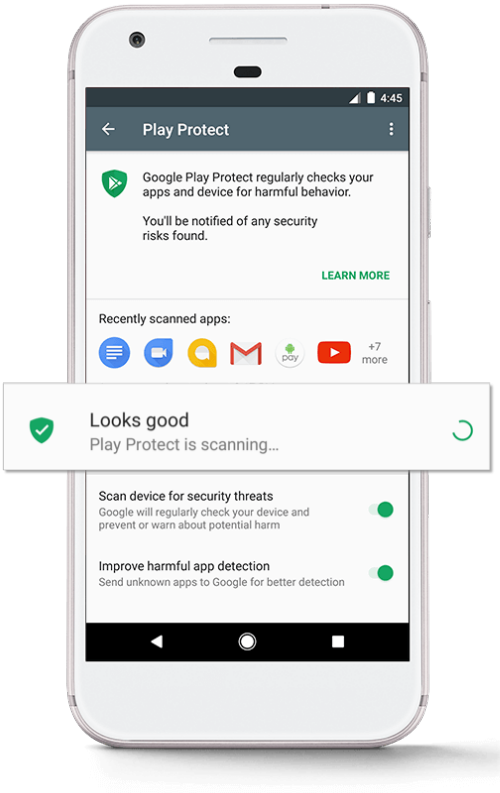
To address this, in October 2017, Google Play Protect added offline scanning, which helps prevent well-known PHAs from being installed offline. When the device regains network connectivity, it undergoes a full scan.

> Since October 2017, offline scanning blocked over 10 million harmful app installs.

Automatically disable PHAs

In November 2017, we updated Google Play Protect to disable PHAs without uninstalling them. Whenever possible, we try to leave as much choice in users' hands as possible. To walk the line between user choice and safety, when Google Play Protect detects certain kinds of PHAs, it automatically disables

User-initiated scan in Google Play Protect



the app. Users are asked to uninstall the app or re-enable it without losing their data. This mitigates the potential harm the PHA could cause while providing minimal inconvenience to the user while they decide what to do.

> In 2017, Google Play Protect automatically disabled PHAs from roughly 1 million devices.

Review apps from outside of Google Play
Because we try to protect users from PHAs regardless of the source, it is important that our systems analyze and understand as many apps as possible.

> Google reviews all apps before publishing them in Google Play.

In addition to reviewing apps submitted to Google Play, our cloud-based systems look for apps in publicly available sources. Google Play Protect also reviews the apps it finds outside of Google Play for PHAs.

Users can allow Google to review new apps by enabling “Improve harmful app detection” in Google Play Protect on their device. This feature sends apps that we haven't previously analyzed to Google. The more apps our systems analyze, the better they are at identifying and limiting the impact of PHAs for all devices.

> In 2017, Google Play Protect reviewed about 23 million new apps, up 65% from 2016.

Find My Device
Since 2013, Google has provided the Android Device Manager service to locate lost devices. In 2017, this service got more features and became Find My Device.

Find My Device is enabled by default on Android devices running Android 4.4 and above. To work, the device needs to be connected to the internet, signed in to a Google Account, and have Location enabled.

In 2017, we helped users find their lost devices with the Find My Device app and android.com/find. In addition to a better app and website, we added these new features:

- **Display last known location.** If the device isn't connected to the internet and can't report its current location, Find My Device displays the last known location of the device from the user's Google Maps location history. Users can also launch Maps' location timeline from the Find My Device app so they can retrace their steps.

- **Display last connected Wi-Fi access point.** Helps users determine the location of their lost device (such as, home or work) even if the device can't be reached to report its location.

- **Display battery level.** Helps users estimate how much longer they can reach their phone.

- **Improved usability.** It's now easier for users with multiple devices to select the one they're looking for and perform common actions, such as ring, lock, and erase.

Android Wear and Google Home also support Find My Device. Users can find their watch with their phone and their phone with their watch (as long as both devices have location enabled) or ask their Google Home to find their device by saying, "Ok Google, where is my Phone?"

Developer APIs

Since 2013, Android devices have included SafetyNet, which allows devices to contribute security-related information to Google's cloud-based services. This can include information about security events, logs, configurations, and other security-relevant details.

In 2017, SafetyNet added new APIs to allow developers to raise the security bar for their apps.

The SafetyNet attestation API helps an app evaluate whether it is talking to a genuine Android device. For more information, see the developer documentation and the SafetyNet API Samples on GitHub.

In June 2017, SafetyNet launched the reCAPTCHA API, which uses an advanced risk analysis engine to protect apps from spam and other abusive actions. If the service suspects that the user interacting with an app might be a bot instead of a human, it serves a CAPTCHA for the user to solve before continuing. Since its release, many major social, travel, and gaming companies have incorporated this API to help keep their apps safe. For more details, see the reCAPTCHA Android API blog post.

Developers and enterprises can use the Verify Apps API to determine whether a device is Google Play Protect capable, encourage users to enable Google Play Protect, and identify any known PHAs that are installed on the device. For more details, see the SafetyNet Verify Apps API blog post.

Safe Browsing

Google introduced Safe Browsing in 2007. Safe Browsing protects users against threats by allowing apps to check URLs against lists of unsafe web resources, such as social engineering sites (phishing and deceptive sites), and sites that host PHAs or unwanted software. When users attempt to visit an unsafe web resource, their Safe Browsing-supported browser displays a warning.

In 2016, Safe Browsing added a device-local Safe Browsing API and started protecting users from repeatedly dangerous websites.

In 2017, Android 8.0 added Safe Browsing as an opt-in feature. Now apps can choose to easily take advantage of the Safe Browsing API, which protects users from phishing and PHA host sites while they are browsing in the app's WebView. Together with Chrome, Safe Browsing protects over 3 billion devices and shows over a million warnings a month.

Cloud-based security analysis

Google analyzes data from over 2 billion Android devices to identify signals that indicate potential abuse or security concerns. This section describes how we updated our analysis capabilities in 2017.

Application security analysis

Before apps become available in Google Play, they undergo a review to confirm that they comply with Google Play policies. Google created an automated app risk analyzer that performs static and dynamic analysis of APKs to detect potentially harmful app behavior. If the risk analyzer discovers something suspicious, it sends the offending app to a team of security experts for manual review.

For a more details on our security analysis process, see 2016's Year in Review.

Machine learning improvements

In 2016, our systems started using machine learning to help detect and classify mobile threats. Machine learning consists of training a computer algorithm to recognize behavior by giving it hundreds of thousands of examples of that behavior. In 2017, we expanded Google Play Protect's machine learning

capabilities by exploring different techniques and leveraging new knowledge from all across Google.

Google's systems learn which apps are potentially harmful and which are safe by analyzing our entire app database. The algorithms look at hundreds of signals and compare behavior across the Android ecosystem to see if any apps are attempting anything suspicious, such as interacting with other apps on the device in unexpected ways, accessing or sharing personal data without authorization, aggressively installing apps (including PHAs), accessing phishing websites, or bypassing built-in security features. In addition to app behavior, Google Play Protect's algorithms started analyzing where PHAs come from and how they make money in 2017. PHA developers often create apps in clusters, so these new techniques help us identify new PHAs more quickly.

To better correlate PHAs, we also started designing and implementing new models based on deep neural networks. These models can take multiple signals as inputs and combine all weighted signals together to interpret the statistical interactions captured to identify the likelihood of an app being a PHA. In 2017, we created models for some major PHA categories and in 2018 we're continuing this work.

These improvements help classify apps that exhibit malicious behavior and group similar bad apps together into families. Visualizing these families helps uncover unknown apps that share similarities with known PHAs. We confirm the findings from these algorithms with manual reviews by human security experts. We then feed confirmed data back into Google Play Protect's detection and classification services, so that they can continue to improve with new information.

> Our machine learning models successfully detected 60.3% of PHAs identified by Google Play Protect in 2017.

System image scanning

As Android's platform security tools improve, they can investigate pre-installed apps in the system partition to make sure that they are not in fact PHAs. In November 2017, we started scanning for pre-installed PHAs across many software builds for devices with Google services. If our tools find a PHA, Google works with the device manufacturer partner to remove the PHA before the device or system partition update is released.

Android platform security

We improve platform security with major Android releases and monthly security updates. To be fully effective, security must be part of a product's fundamental design. Android's architecture was designed with security in mind. The Android platform controls how the operating system works and how apps interact with each other and with the various components of device hardware. Android also includes protections designed to ensure all apps operate consistently and safely. This table lists some of these protections and how they contribute to platform-level security.

Platform security feature	Protection
Encryption	Protects data from unauthorized access.
Hardware-backed security	Strengthens key storage and cryptographic services and enables strong remote authentication.
Kernel self-protections	Protects kernel against memory corruption vulnerabilities and other security flaws in kernel components and drivers.
Sandboxing	Keeps each app in a separate space, protecting its data and processing from other apps.
SELinux	Provides an auditable definition of—and enforcement for—security boundaries on all operating system and app components above the kernel.
Userspace hardening	Protects operating system and apps against memory corruption vulnerabilities and other security flaws; includes Address Space Layout Randomization (ASLR) and Data Execution Prevention (DEP).
Verified Boot	Verifies the operating system starts in a known good state.

Updates and features

In 2017, we released Android 8.0 (Oreo). This section summarizes major security features included in the Android platform and highlights their improvements in Android 8.0. For more updates, see Security Enhancements in Android 8.0.

Project Treble

One of the most important updates in Android 8.0 is Project Treble. Treble is more than a security feature—it is a series of important architectural changes that has a large, positive impact on security.

In order for a device to run Android, device manufacturers customize the software to talk with the device's hardware. In previous Android versions, these customizations were mixed with the general Android OS framework. In Android 8.0, Google worked closely with device manufacturers and system-on-chip (SoC) vendors to address this problem, resulting in the biggest change to the low-level system architecture of Android to date. Treble separates the vendor implementation—the device-specific, lower-level software written by SoC vendors—from the Android framework.

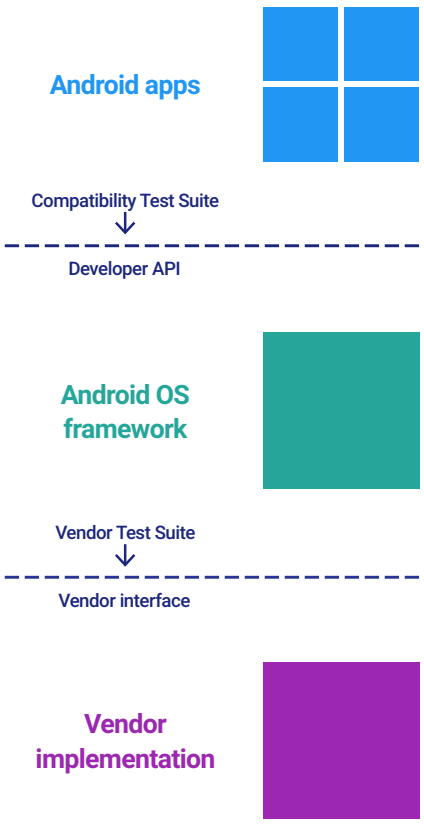
To accomplish this, Android 8.0 includes a new vendor interface between the Android framework and the SoC vendor implementation. The new vendor interface is validated by a Vendor Test Suite (VTS), which verifies forward compatibility of the vendor implementation.

This separation makes updating devices to new versions of Android much easier as it leaves the vendor implementation intact. Historically, updates were challenging, costly, and time consuming for device manufacturers because of their customizations.

> Devices with Treble are easier to update, which should mean faster security patches and Android version updates for the whole ecosystem.

Faster and easier updates aren't Treble's only security benefit. Treble's modularity is designed to improve security through increased isolation and de-privileging of vendor-provided hardware abstraction layers (HALs). In earlier versions of Android, HALs were run in-process. The process needed all the permissions required by all in-process HALs, including direct access to kernel drivers. Likewise, all HALs in a process had access to the same set

Simplified Android architecture with Treble



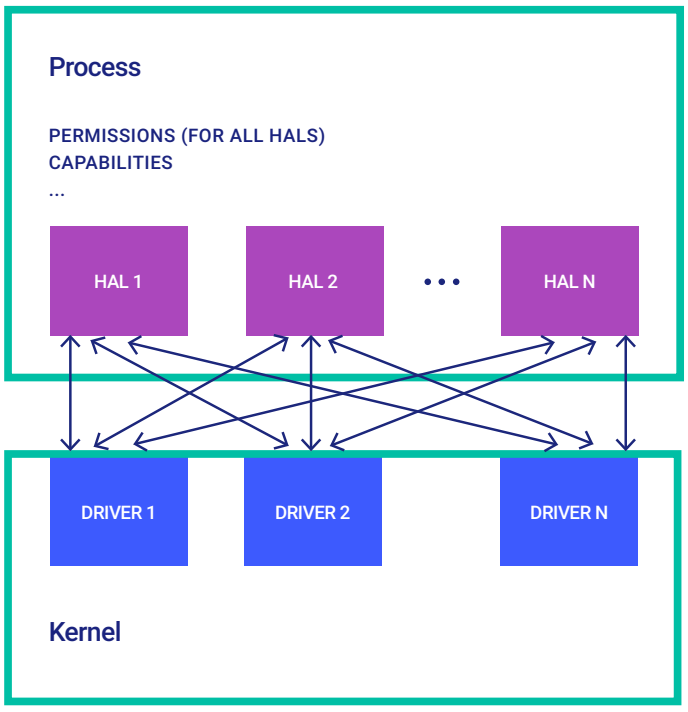
of permissions as the rest of the process, including permissions required by other in-process HALs. This resulted in over-privileged processes and HALs that had access to permissions and hardware that they shouldn't.

In Android 8.0, we moved HALs into their own processes. Isolated HALs better adhere to the principle of least privilege and provide two distinct advantages:

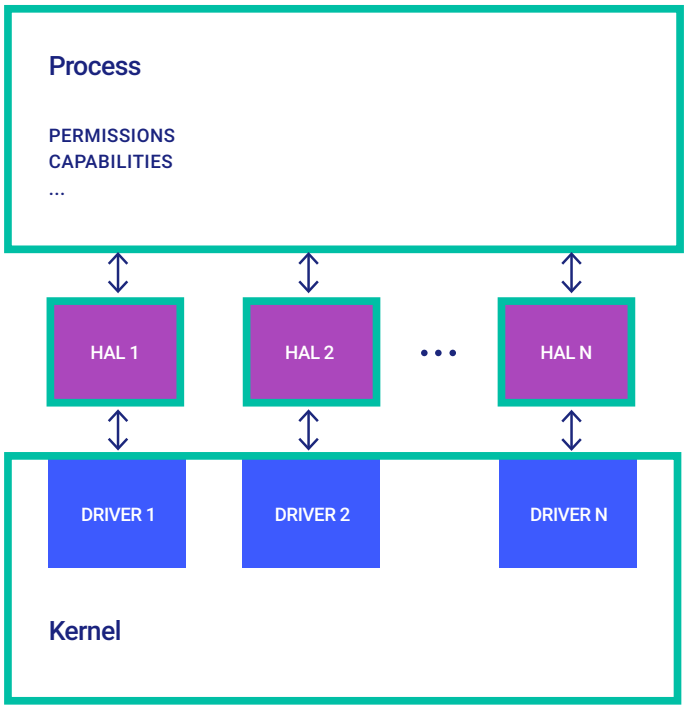
— Each HAL runs in its own sandbox and can access only the hardware driver it controls and the permissions required to do its job.

— The process loses access to hardware drivers and other permissions and capabilities needed by the HALs.

Security boundaries in Android 7.0 and earlier



Security boundaries in Android 8.0+



Security boundary

For more information on how this change helped to reduce access to privileged system permissions, remove direct hardware access from the media frameworks, and make the Linux kernel more secure, see our Shut the HAL up blog post.

Android Verified Boot 2.0

In 2017, we re-architected Verified Boot to work with Treble. Verified Boot, introduced in Android 4.4, provides a hardware-based root of trust, and confirms the state of each stage of the boot process for devices with at least 1GB of RAM. During boot, Android warns the user if the operating system has been modified from the factory version, provides information about what the warning means, and offers solutions to correct it. If the device running Android 7.0+ has corrupt data, Verified Boot either allows the boot to proceed, stops the device from booting so the user can take action on the issue, or prevents the device from booting up until the issue is resolved.

Android 8.0 includes Android Verified Boot 2.0 (AVB), which is a reference implementation of Verified Boot that works with Treble. In AVB, separate signing keys are used for individual partitions. Where previously one signing key was used to verify everything, AVB separates that responsibility by partition. Now, device manufacturers can allow other companies to update specific partitions on the device using different public keys, which should lead to faster updates.

AVB also implemented rollback protection, which prevents attackers from returning a partition to an old image with a persistent known vulnerability.

Additionally, AVB also simplified and standardized a few things across the boot process. AVB specifies a common footer format for boot and dm-verity partitions, which previously used different formats for metadata. This standardization makes updates and fetching metadata easier across partitions. AVB also reduces bootloader complexity, which lessens the possibility of vulnerabilities that occur as a result of that complexity.

Encryption

Android has supported encryption since Android 3.0. Android 8.0 supports using tamper-resistant hardware to verify a device's lock screen passcode and unlock disk encryption. Tamper-resistant hardware is a chip separate from the SoC. It includes its own flash, RAM, and other resources inside a single package, so it can fully control its own execution in a known secure environment. It can also detect and defend against physical tampering attempts. Having a dedicated security module keeps the device passcode safer and makes it even more difficult for attackers to decrypt device data without knowing that passcode. For more details on the benefits of tamper-resistant hardware, see the hardware protections in the Google Pixel 2 phones blog post.

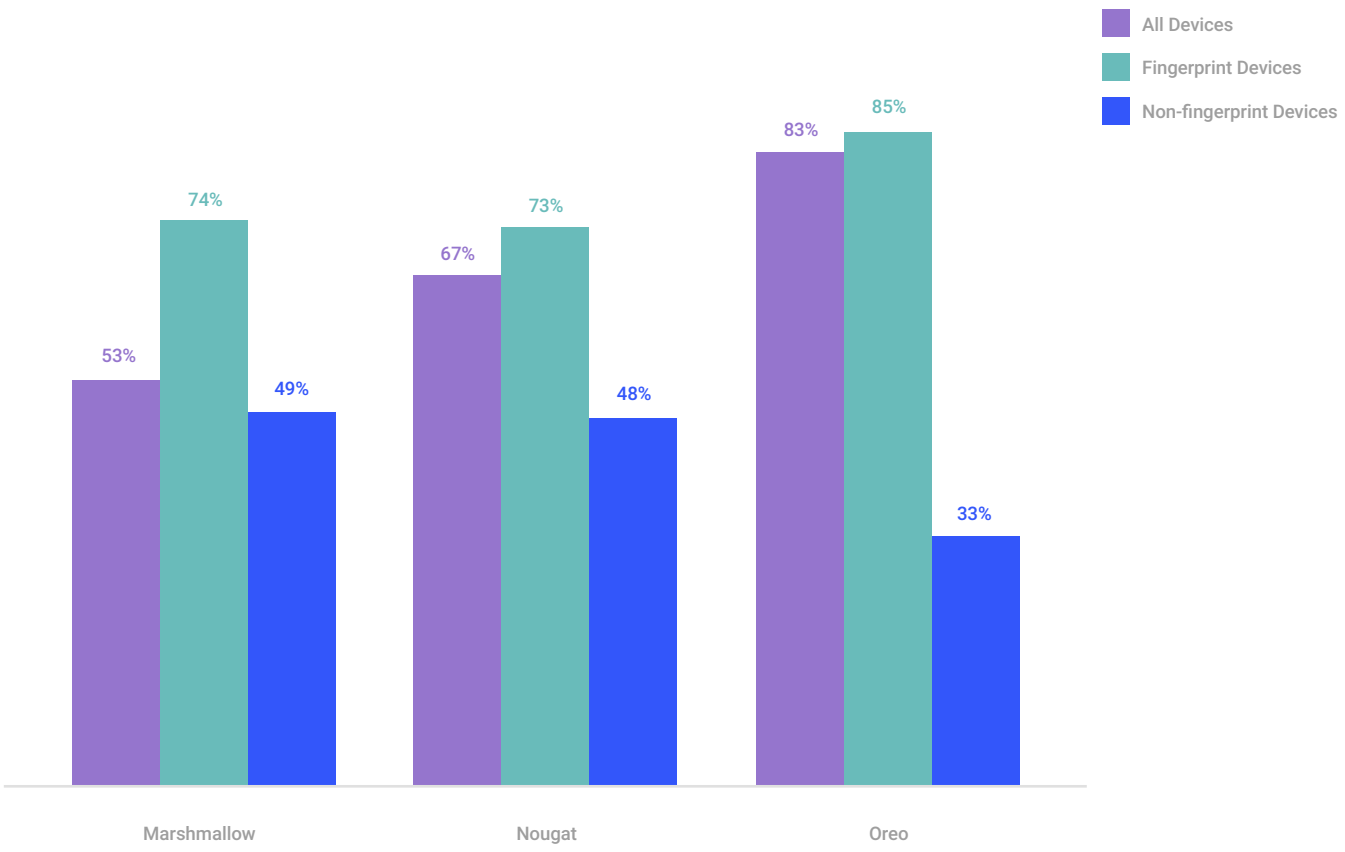
Secure lock screen

Robust encryption isn't helpful without a secure lock screen, as files are encrypted against the device's PIN, pattern, or password. If a device has

a passcode, an attacker cannot access its data without knowing the passcode first. To increase security on the lock screen and make it faster and easier for users to unlock their device, the Android platform has steadily added lock screen features.

Android 6.0 added platform support for fingerprint sensors, which encouraged more users to set a lock screen. Before fingerprint sensor support, less than half of devices had a lock screen. As time has passed, more new devices include fingerprint sensors, which make it easier for users to set a secure lock screen. As of 2017, 85% of devices with a fingerprint sensor running Android 8.0 had a secure lockscreen.

Devices that have a secure lock screen, by Android release



Keystore updates

Android's keystore helps app developers use cryptographic keys for authentication and verification. Keystore gives apps a safe place to store key information. Devices with hardware-backed security keep keys even safer by keeping key material out of the Android system entirely.

In Android 7.0, Keymaster 2 added support for key attestation and version binding. Key attestation provides public key certificates that contain a detailed description of the key and its access controls. These certificates make the key's existence in secure hardware and its configuration remotely verifiable.

Version binding ties keys to operating system and patch level version. Version binding makes it harder for an attacker who discovers a weakness in an old version of system or Trusted Execution Environment (TEE) software to roll a device back to the vulnerable version and use keys created with the newer version.

In Android 8.0, we released Keymaster 3 along with Treble. In addition to this update, Android 8.0 extends the key attestation feature to support ID attestation, which verifies hardware identifiers, such as device serial number, product name, and phone ID (IMEI / MEID).

Permissions

In Android 8.0, we tightened up permissions in a variety of ways.

The System Alert Window API allows apps to draw special windows that can display over everything to deliver system-critical messages to users. Most apps followed the rules, but some used the overlay types to create ransomware, fullscreen clickjacking ads, and other types of abuse. Starting in Android 8.0, apps can't display windows above the lock screen, the status bar, IME, and other system-critical screens. Now when an app displays a window that covers the screen, users can tap a notification to hide that window.

Android Instant Apps allow users to run an app without downloading it. Instant Apps keep devices safe by running the app in a restricted sandbox that limits permissions and capabilities, such as reading the on-device app list or transmitting cleartext traffic. Although introduced with Android 8.0, Instant Apps supports devices running Android Lollipop and newer.

Before Android 8.0, if a user wanted to install an app from outside Google Play or an OEM app store, they needed to enable the ‘Allow unknown sources’ setting for the whole device. In Android 8.0, the Install unknown apps permission makes it safer to install apps from unknown sources.

This permission is tied to the app that prompts the install—just like other runtime permissions—and asks the user to grant permission to use the install source before prompting the user to install an app. When used on a device running Android 8.0, hostile downloaders can't use pre-existing permission to trick the user into installing an app. For more details, see the Making it safer to get apps on Android O blog post.

Privacy controls

Along with tighter permissions, Android 8.0 improved privacy controls for devices and apps. Android 8.0 improved user control over device identifiers, such as usernames or the device's IMEI; limited the use of permanent device identifiers, such as the device's serial number; and updated the way that apps request account information. Android 8.0 also has an updated Wi-Fi stack that works with changes to the Wi-Fi chipset firmware used by Pixel and some Nexus devices to provide robust randomization of MAC addresses in probe requests. For more details, see the Changes to Device Identifiers blog post.

In 2017, the Google Play Console added additional privacy warnings for the pre-launch report to help developers respect user privacy. The Play Console now warns about using anomalous permissions, sending hardware identifiers off the device, and sending the Advertising ID unencrypted.

Kernel hardening

As we've worked hard to harden Android's userspace over the years, the underlying Linux kernel has become a more attractive target to attackers. As a result, more than a third of Android security bugs targeted the kernel in 2016. In Android 8.0, we further hardened the kernel to reduce the number and impact of security bugs. Changes include backported hardened usercopy, ARM Privileged Access Never (PAN) emulation, post-init read-only memory to kernel versions 3.18+, and Kernel Address Space Layout Randomization (KASLR) to the Android kernel 4.4. These hardening features mitigate the most common source of security bugs in the kernel, and make it easier for driver developers to discover common bugs during development.

Android 8.0 also takes advantage of a Linux feature called seccomp to make unused system calls (syscalls) inaccessible to apps. Because apps can't access these syscalls, they can't be exploited by PHAs. For more information about blocked syscalls, see the Seccomp filter in Android O blog post.

Sanitizers and fuzzing

Fuzzing is providing potentially invalid, unexpected, or random data as an input to a program to help find bugs. While Android has supported fuzzing tools for many releases, Android 8.0 includes more fuzzing support, tighter fuzzing tool

integration in the Android build system, and greater dynamic analysis support on the Android kernels.

LLVM, the compiler infrastructure used to build Android, contains multiple components that perform static and dynamic analysis. We use these various components to try to shake the bugs out of Android. Some of the components are covered below.

AddressSanitizer
AddressSanitizer (ASan) helps detect various memory errors in C/C++ code at runtime. In Android 8.0, all of Android can be built and run with ASan enabled, allowing for more effective debugging and fuzzing sessions. Google is using these improvements today, as are many researchers who submit ASan crash logs with their bug reports to the Android Security Rewards program.

SanitizerCoverage
SanitizerCoverage is a simple code coverage capability provided with LLVM. It instruments code to be analyzed with additional user-definable callbacks to allow for coverage reporting, visualization, and analysis. In Android 8.0, SanitizerCoverage was updated to allow for edge coverage instrumentation and more user-friendly callbacks.

UndefinedBehaviorSanitizer
UndefinedBehaviorSanitizer (UBSan) checks for various undefined behaviors and unsigned integer overflows. UBSan has been useful in discovering any latent integer-overflow vulnerabilities in the mediaserver components.

LibFuzzer
Android's build system supports fuzzing through LLVM's libFuzzer project. LibFuzzer is a library that is linked with the function under test, and handles all actions applicable to fuzzing, such as input selection, mutation, and crash reporting. LibFuzzer's memory corruption detection and code coverage metrics are implemented with the LLVM Sanitizers previously described. In Android 8.0, we added upstream improvements to libFuzzer, allowing for better corpus management and more efficient and directed fuzzing sessions.

KernelAddressSanitizer
Android also includes the Kernel Address Sanitizer (KASAN). KASAN is a combination of kernel- and compile-time modifications, which make it easier to discover bugs and analyze their root-cause. It can detect various memory violations in the kernel, including out-of-bound reads and writes, use-after-free, and double frees.

Similar to ASAN, KASAN combines memory-function instrumentation at compile time with shadow memory to track memory accesses at runtime. KASAN dedicates an eighth of the kernel memory space to shadow memory, which determines if a memory access is valid or not.

KASAN is supported on x86_64 and arm64 architectures. It has been part of the upstream kernel since 4.0, and has been backported to Android 3.18-based kernels and newer. With Android 8.0, KASAN-enhanced kernels boot and run on all Pixel devices that have at least a 3.18-based kernel. This has made kernel debugging and kernel fuzzing easier and more efficient.

Vulnerability rewards and updates

While we update Android platform security with major releases, Google also oversees many programs to provide more frequent security updates to Android devices. In 2017, Google and Android partners worked together to keep hundreds of millions of device secure.

Android Security Rewards program

2017 was the second full year of the Android Security Rewards (ASR) program. Since June 2015, the program has awarded over 177 different researchers with almost 1,000 rewards for discovering and reporting vulnerabilities on Nexus and Pixel devices and the Android platform.

> In 2017, the Android Security Rewards program paid out $1.28 million, totaling $2.2 million since the start of the program.

Of the amount awarded in 2017, researchers donated $26,500 of their rewards to charity, which Google matched bringing the total to $53,000. We wish to thank these researchers for their generosity.

In June 2017, the ASR program increased its top-line payouts for exploits that compromise TrustZone or Verified Boot from $50,000 to $200,000 and remote kernel exploits from $30,000 to $150,000. In the December Android security bulletin, Guang Gong claimed the first larger payout, receiving $105,000 (and an additional $7,500 from the Chrome team) for a report, which is detailed in his guest blog post.

Security research competitions

In addition to the ongoing ASR program, the Android security team participated in external vulnerability discovery and disclosure competitions, such as Mobile Pwn2Own at PacSec.

> At the 2017 Mobile Pwn2Own competition, no exploits successfully compromised Google Pixel devices.

Of the exploits that were demonstrated against devices running Android, none could be reproduced on a device running unmodified Android source code from the Android Open Source Project (AOSP).

Zero days

In 2017, no critical security vulnerabilities affecting the Android platform were publicly disclosed without an update or mitigation available for Android devices. This was possible due to the ASR program, enhanced collaboration with the security researcher community, coordination with industry partners, and built-in security features of the Android platform.

An example of this coordinated effort to protect Android users was our response to the Blueborne vulnerabilities (CVE-2017-0781, CVE-2017-0782, CVE-2017-0783, and CVE-2017-0785). These vulnerabilities were privately reported to Google by security researchers at Armis through the ASR program. Google worked with the researchers and industry partners to address their findings and create updates for devices running Android and other operating systems before releasing a coordinated industry-wide disclosure.

We followed a similar process for the Spectre and Meltdown vulnerabilities, which we mitigated on Android devices using ARM processors before public disclosure as part of the January 2018 Android security updates.

We also saw Android's built-in security features mitigate the impact of vulnerabilities, such as KRACK. The Android security platform was designed so that the security of the network should not have any impact on device or data security and thus Android shouldn't need to rely on network-level security protections, such as WPA. This means that if a user joins an untrusted network at a coffee shop, they should be protected by default. As a result, the KRACK vulnerabilities have a reduced impact on Android and had no effect on encrypted traffic.

Android security updates program

Google mitigates security vulnerabilities discovered through the ASR program and additional engagements through regular Android security updates. In 2017, we continued to work with Android device manufacturers, mobile network operators, and SoC vendors to increase the number of devices receiving regular security updates.

In October 2017, we launched the Pixel/Nexus security bulletins to bring additional transparency and clarity to the security and functional updates provided to Google Pixel and Nexus devices. The Pixel / Nexus security bulletins contain details of security vulnerabilities and functional improvements affecting supported Google Pixel and Nexus devices that are recommended for compliance with a security patch level.

Other Android device manufacturers are following our lead by including these optional updates on their devices, enhancing their security as well as the overall security of the Android ecosystem.

Android device update metrics

Through our combined efforts, the Android ecosystem has made significant progress in releasing the latest security updates. In fact, the majority of the deployed devices for over 200 different Android models from over 30 Android device manufacturers are running a security update from the last 90 days.

> As of December 2017, over 90% of deployed Google Pixel 2 devices were running a security update from the last 90 days.

Increasing security update adoption

Beyond supporting Treble, Android 8.0 made a number of changes to make it easier for devices to receive and apply security updates.

Seamless System Updates apply security updates in the background without interrupting the user and also stream the update to avoid disk space issues. This increases the likelihood of update adoption because the user only needs to restart the device to finish installing the update. For extra safety, the device can always fall back to the previous working version if there are any issues with an update.

Most OTAs have a phased rollout, which means the update reaches a small set of users first and gradually makes it to everyone. On devices running Android 8.0, if a user checks for an available security update, they can get it immediately, even if it hasn't rolled out to all devices.

These features come standard on Google Pixel devices and make sure that updates are fast and easy. Google continues to work with the Android ecosystem to bring these improvements to all Android devices.

App Security Improvements Program

In addition to working with device manufacturers and keeping the platform up-to-date, Google also works with app developers to improve the security of their apps. The App Security Improvements (ASI) program notifies developers about vulnerabilities affecting their apps in Google Play. As vulnerable apps are identified, the ASI program contacts developers by email and the Google Play Console with guidance for remediation. To encourage prompt security fixes, developers are given 90 days to provide a fixed version of the app. After 90 days, any update to their app must include the fix and new apps with the vulnerability are not accepted in Google Play.

In September 2017, we launched a new warning for a Path Traversal Vulnerability.

> Since 2014, the ASI program has supported the discovery of 27 different types of vulnerabilities and fixed nearly 383,000 vulnerable apps in Google Play.

Google Play Security Reward Program

To encourage finding and fixing more vulnerable apps, we launched the Google Play Security Reward Program for researchers to submit vulnerability reports for popular apps in Google Play. In collaboration with the independent bug bounty platform, HackerOne, Google and 18 other developers are offering rewards. We are actively growing the program and inviting more developers of popular Android apps to join.

If a security researcher finds a vulnerability in any participating app, they submit a report to the app developer. After that vulnerability is addressed, the researcher submits a report to the Play Security Reward Program to receive a $1000 bounty from Google Play. Google then analyzes the bug and scans all the apps in Google Play for the same vulnerability. If the issue is present in other apps, the affected developers are notified through the ASI program. This way, the research benefits all affected apps, the larger Google Play ecosystem, and most importantly the users who get their apps from Google Play.

Ecosystem data

This section covers details and trends for Potentially harmful application (PHA) categories in the Android ecosystem, both inside and outside of Google Play. To gather this information, a variety of systems, such as SafetyNet and Verify Apps, work together. For example, SafetyNet looks at the larger picture over time by gathering anonymized device telemetry to track PHAs in the ecosystem and provide insight on security-related services usage. In contrast, Verify Apps is useful in the moment by scanning apps for PHAs before users install them, regardless of their origin.

Potentially harmful applications

Potentially harmful applications are apps that could put users, user data, or devices at risk. Common PHA categories include trojans, spyware, or phishing apps.

Apps that weaken or disable Android's built-in security features are potentially harmful but can also provide functionality that users find useful and desirable. In our statistics, we represent these apps differently from other PHA categories because users intentionally download the apps. To make sure that users are aware of the risks, Google Play Protect still displays a warning when they try to install these kinds of apps. For example, we warn users about apps that disable Android security features like SELinux or root the device with user consent. Power users can proceed with installation while users who were not aware of the dangers can make more informed decisions about altering their device. We generally discourage any changes that lower Android's built-in security protections, but we believe in letting individuals choose what risks they are comfortable taking with their devices.

In 2017, we extended the definition of hostile downloaders to cover any app that downloads and installs other apps if the percentage of PHA among those

downloaded apps exceeds 5%. We also changed how we handle apps that download and install other apps. If a PHA attempts to install another app, a warning is shown to the user. If a known hostile downloader or trojan attempts to install another app, the install attempt is silently blocked without user notification.

This table contains the categories that we use to classify PHAs. We warn users when they attempt to install an app that falls into one of these categories. Darker rows signify a change to the definition in 2017.

PHA	Definition
Backdoors	An application that allows the execution of unwanted, potentially harmful remote-controlled operations on a device that would place the app into one of the other PHA categories if executed automatically. In general, the backdoor is more a description of how a potentially harmful operation can happen on a device and is therefore not completely aligned with PHA categories like billing fraud or commercial spyware apps.
Commercial spyware	Any application that transmits sensitive information off the device without user consent and does not display a persistent notification that this is happening. Commercial spyware apps transmit data to a party other than the PHA provider. Legitimate forms of these apps can be used by parents to track their children. However, these apps can be used to track a person (a spouse, for example) without their knowledge or permission if a persistent notification is not displayed while the data is being transmitted.
Denial of service	An application that, without the knowledge of the user, executes a denial-of-service attack or is a part of a distributed denial-of-service attack against other systems and resources. This can happen by sending a high volume of HTTP requests to produce excessive load on remote servers.
Hostile downloader	An application that is not potentially harmful by itself, but downloads other potentially harmful applications. An app may be a hostile downloader if: — There is reasonable cause to assume that the app was created to spread PHAs and the app has downloaded PHAs or contains code that could download and install apps. — At least 5% of apps downloaded by the app are PHAs with a minimum threshold of 500 observed app downloads (25 observed PHA downloads). Major browsers and file sharing apps are not considered hostile downloaders as long as: — They don't drive downloads without user interaction — All PHA downloads are initiated by consenting users

PHA	Definition
Mobile billing fraud	An application that charges the user in an intentionally misleading way. Mobile billing fraud is divided into SMS fraud, Call fraud, and Toll fraud based on the type of fraud being committed.
SMS fraud	An application that charges users to send premium SMS without consent, or tries to disguise its SMS activities by hiding disclosure agreements or SMS messages from the mobile operator notifying the user of charges or confirming subscription. Some apps, even though they technically disclose SMS sending behavior introduce additional tricky behavior that accommodates SMS fraud. Examples of this include hiding any parts of disclosure agreement from the user, making them unreadable, conditionally suppressing SMS messages the mobile operator sends to inform user of charges or confirm subscription.
Call fraud	An application that charges users by making calls to premium numbers without user consent.
Toll fraud	An application that tricks users to subscribe or purchase content via their mobile phone bill. v includes any type of billing except Premium SMS and premium calls. Examples of this include: Direct Carrier Billing, WAP (Wireless Access Point), or Mobile Airtime Transfer. WAP fraud is one of the most prevalent types of Toll fraud. WAP fraud can include tricking users to click a button on a silently loaded transparent WebView. Upon performing the action, a recurring subscription is initiated, and the confirmation SMS or email is often hijacked to prevent users from noticing the financial transaction.
Non-Android threat	An application that contains non-Android threats. These apps are unable to cause harm to the user or Android device, but contain components that are potentially harmful to other platforms.
Phishing	An application that pretends to come from a trustworthy source, requests a user's authentication credentials and/or billing information, and sends the data to a third party. This category also applies to apps that intercept the transmission of user credentials in transit. Common targets of phishing include banking credentials, credit card numbers, or online account credentials for social networks and games.
Privilege escalation	An application that compromises the integrity of the system by breaking the application sandbox, or changing or disabling access to core security-related functions. Examples include: — An app that violates the Android permissions model, or steals credentials (such as OAuth tokens) from other apps. — An app that prevents its own removal by abusing device admin APIs. — An app that disables SELinux. Note: Privilege escalation apps that root devices without user permission are classified as rooting apps.

PHA	Definition
Ransomware	An application that takes partial or extensive control of a device or data on a device and demands payment to release control. Some ransomware apps encrypt data on the device and demand payment to decrypt data and/or leverage the device admin features so that the app can't be removed by the typical user. Examples include: — Ransomware that locks a user out of their device and demands money to restore user control. — Ransomware that encrypts data on the phone and demands payment, ostensibly to decrypt data again. — Ransomware that leverages device policy manager features and cannot be removed by the user.
Rooting	A privilege escalation app that roots the device. There is a difference between malicious rooting apps and non-malicious rooting apps. Non-malicious rooting apps let the user know in advance that they are going to root the device and they do not execute other potentially harmful actions that apply to other PHA categories. Malicious rooting apps do not inform the user that they will root the device, or they inform the user about the rooting in advance but also execute other actions that apply to other PHA categories.
Spam	An application that sends unsolicited commercial messages to the user's contact list or uses the device as an email spam relay.
Spyware	An application that transmits sensitive information off the device. Transmission of any of the following without disclosures or in a manner that is unexpected to the user are sufficient to be considered spyware: — contact list — photos or other files not owned by the application — content from user email — call log — SMS log — web history or browser bookmarks of the default browser — information from the /data/ directories of other apps. Behaviors that can be considered as spying on the user can also be flagged as spyware. For example: recording audio or recording calls made to the phone, stealing application data, etc.
Trojan	An application that appears to be benign, such as a game that claims only to be a game, and performs undesirable actions against the user. This classification is usually used in combination with other categories of harmfulness. A trojan will have an innocuous app component and a hidden harmful component. For example, a tic-tac-toe game that, in the background and without the knowledge of the user, sends premium SMS messages from the user's device.

User-intended rooting

Some users choose to root their phones to access functionality that is not available in the standard Android configuration. Because these apps are frequently intentionally installed by a user to customize their device, apps that root the device with full disclosure and user consent are tracked separately from apps that root the device without user disclosure or consent. We do this because rooting a phone removes some fundamental security protections and we want to monitor how much of the ecosystem is in this intentionally weakened state.

As previously discussed, Google's SafetyNet service provides an attestation feature that checks for unexpected changes to the security state of the device. This API is invoked about 260 million times a day and provides an approximate number of rooted devices. Worldwide, 94.4% of all Android devices report passing the basic system integrity check, from which we conclude that these devices are free from tampering. The remainder includes devices that were rooted by the user, sold as a rooted device, or were rooted by a PHA.

Google Play Protect follows the ratio of all app installs to user-intended rooting installs. In 2017, user-intended rooting app installs comprised 0.011% of all installs, with fewer than 0.0001% of installs coming from Google Play.

Mobile Unwanted Software (MUwS)

Google uses the concept of "unwanted software" (UwS) as a way to deal with apps that are not strictly considered PHA, but are generally harmful to the software ecosystem. In 2016, Android took a similar approach with mobile, introducing Mobile Unwanted Software (MUwS).

MUwS are prohibited by Google Play's policies, but even outside of Google Play they are harmful to the Android ecosystem and unwanted by most users. An example of common MUwS behavior is overly aggressive collecting of device identifiers or other metadata. Before 2016, some MUwS were categorized as PHAs, especially apps formerly described as Data Collection. Since 2016, MUwS are defined as apps that collect at least one of the following without user consent:

- Device phone number New in 2017
- Primary email address New in 2017
- Information about installed apps
- Information about third-party accounts
- Names of files on the device

To address this, Google works with developers to remove this behavior from their apps or disclose their data collection practices to users. As of late 2017, we have expanded coverage of MUwS apps from Google Play to apps from outside of Play as well.

Changes in methodology

Since 2016, we changed some of our methodology, which led to some variances in numbers in this report compared to last year. These changes include:

- Widened PHA definitions for hostile downloaders. This resulted in more apps categorized as PHAs than last year.
- Ignored PHA installations that users explicitly want. If users make a conscious effort to install and keep a PHA after we warn them, then their device is no longer considered to have a PHA.
- Don't warn me again. If a user decides to keep an app by choosing "Don't warn me again" on the notification from Google Play Protect, then the app is excluded from our device hygiene metrics but the app is still considered a PHA.
- Standardized all metrics to use only 28-day active devices. We previously used different periods in different metrics.

Device and ecosystem hygiene

The broadest statistic we use to measure device hygiene is how frequently a routine full-device scan detects PHAs. Since we began to measure device hygiene in late 2014, less than 1% of devices have PHAs installed on average. This trend continued in 2017.

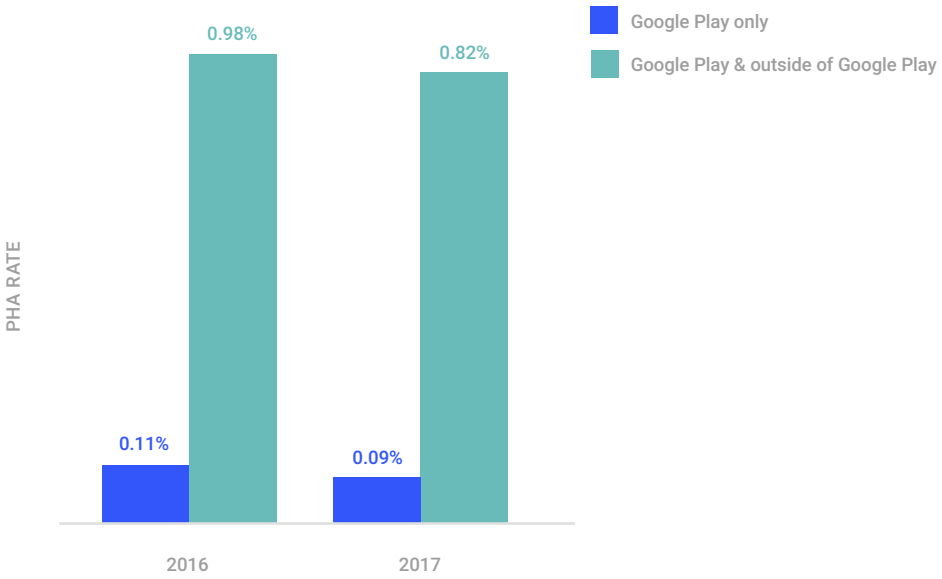
> In 2017, 0.56% of all Android devices scanned by Google Play Protect had installed a PHA, compared to 0.77% in 2016.

Apps available on Google Play must adhere to published policies and are reviewed to verify their compliance. Google Play security continued to improve in 2017. On average, only 0.09% of devices that exclusively used Google Play had one or more PHAs installed, which is less than the 0.11% of affected devices in 2016.

The security of devices that installed apps outside Google Play improved as well. On average, 0.82% of devices that installed apps from outside of Google Play were affected by one or more PHA in 2017. In 2016, this number was 0.98%.

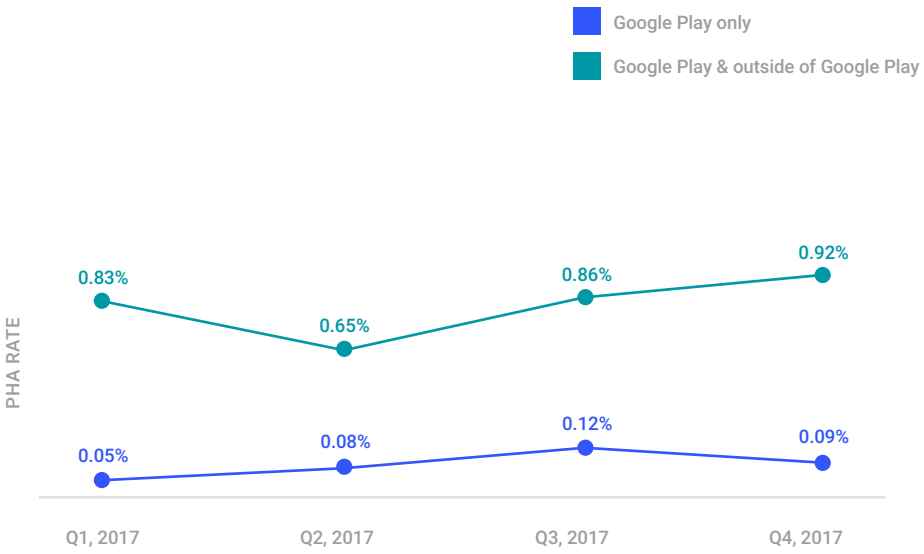
> Devices that installed apps from outside of Google Play were nine times more likely to be affected by PHAs.

Devices with PHA installed, 2016 and 2017



This graph shows the percentage of devices that were affected by PHAs over time. The blue line represents devices that only download apps from Google Play while the light teal line represents devices that also install apps from other sources.

Devices with PHA installed in 2017, by quarter



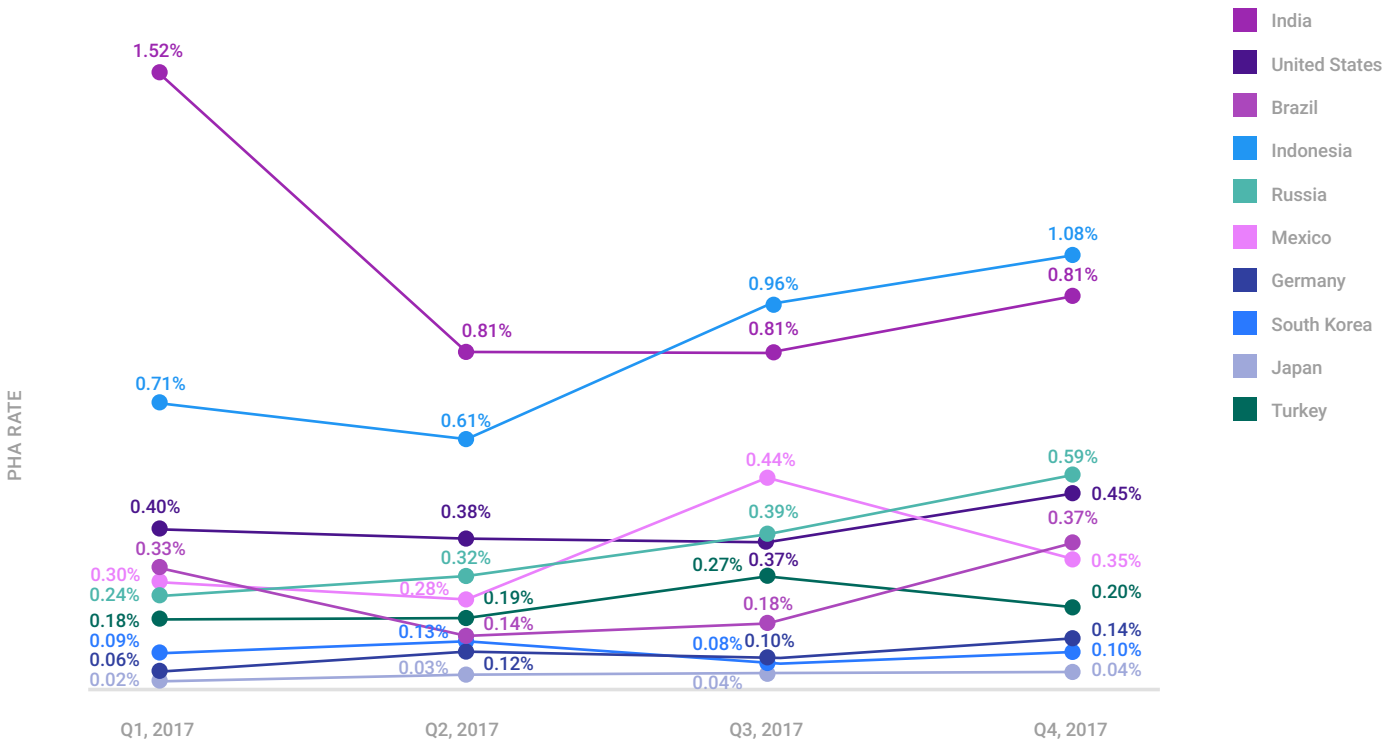
Device hygiene by country

While Google Play and the overall Android ecosystem got safer in 2017, there was some variance based on the country of the device. The following sections discuss PHA rates in the largest overall Android markets and the markets with the highest and lowest prevalence of PHAs. In this year's report, the country-level analysis focuses on the primary location of Android devices instead of install location, like last year.

Largest Android markets

In 2017, India, the United States of America (USA), and Brazil were the three countries with the largest number of Android devices with Google Play Protect. The number of affected devices in varied across these countries. With 1.00% of affected devices, PHAs are more common in India compared to the global average of 0.56%. The USA is slightly safer than the rest of the world with 0.40% of devices affected with PHAs while Brazil is less affected at 0.26%.

Device hygiene of the top 10 countries, by the largest Android markets in 2017



This table shows the device hygiene rates in the 20 largest Android markets.

Country	PHA Rate	%pt Change since 2016	% of all devices
India	1.00%	0.31% ↓	11.0%
United States	0.40%	0.10% ↓	10.6%
Brazil	0.26%	0.10% ↓	6.8%
Indonesia	0.86%	0.06% ↓	4.6%
Russia	0.37%	0.07% ↑	3.9%
Mexico	0.35%	0.07% ↑	3.9%
Germany	0.11%	0.04% ↑	2.9%
Korea	0.10%	0.01% ↑	2.7%
Japan	0.03%	0.01% ↑	2.6%
Turkey	0.21%	0.07% ↑	2.3%

Country	PHA Rate	%pt Change since 2016	% of all devices
France	0.29%	0.09% ↓	2.1%
Spain	0.19%	0.15% ↓	2.2%
Great Britain	0.39%	0.09% ↓	2.1%
Thailand	1.14%	0.32% ↑	2.1%
Italy	0.14%	0.05% ↑	2.1%
Vietnam	0.40%	0.12% ↓	1.9%
Argentina	0.23%	0.11% ↑	1.7%
Philippines	1.52%	0.10% ↑	1.5%
Colombia	0.33%	0.08% ↑	1.4%
Egypt	0.60%	0.23% ↓	1.3%

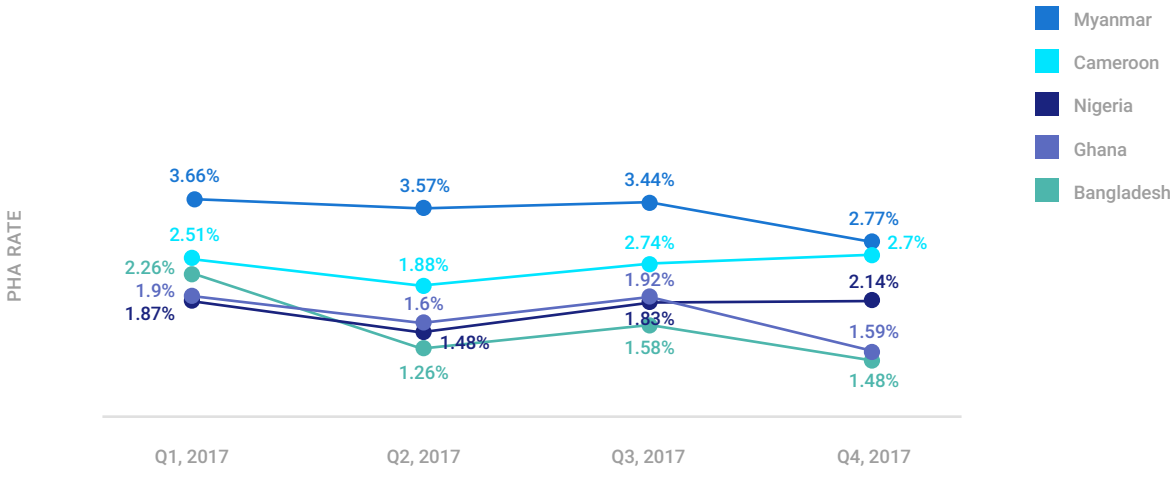
India continues to be affected by trojans, such as Ghost Push and Hummingbad, which we described in the 2016 Year in Review. The spike in Q1 2017 was caused by a legitimate video player from outside of Google Play that downloaded and installed PHAs on user devices. We believe that the developer used an advertising network that pushed PHAs and did not know about this behavior.

The situation looks different in the USA. Many of the PHA installations come from popular rooting tools and an app that fakes GPS coordinates to cheat at Pokémon Go. We don't remove these apps, but still warn users that these apps may degrade device security.

The PHA profile of Brazil looks different from India and the USA. Major contributors to Brazil's PHA rate were two pre-installed apps that send SMS to premium-rate SMS numbers.

Highest percentage of PHA devices
Among countries with more than one million Android devices, PHAs were most prevalent in Myanmar, Cameroon, and Nigeria.

Device hygiene of the 5 countries with the highest percentage of PHA devices in 2017



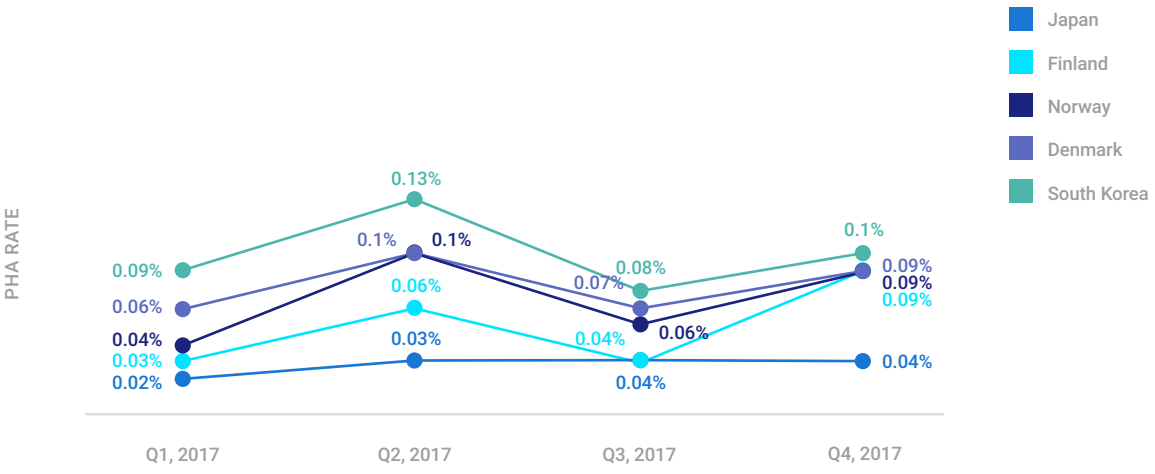
With 3.39% of all Android devices containing PHAs, Myanmar is the most affected country in the world. Three of the top ten PHAs are rooting tools or other power-user utilities that lower device security levels. The others are a mix of different PHA families similar to those found in other South-East Asian countries.

With 2.50% of all Android devices affected by PHAs, Cameroon is second among the most affected countries with at least one million Android devices. Most PHAs in Cameroon are acquired by installing apps from sources other than Google Play. Power-user tools and preinstalled PHAs play a smaller role in Cameroon.

The situation in Nigeria is different from Myanmar and Cameroon. In Nigeria, four of the ten most common PHAs came preinstalled on devices. This hints at a device ecosystem that is not as quality-controlled as other countries or perhaps that financial incentives for device manufacturers and mobile carriers outweigh the privacy and safety concerns of Android users. Of the other six most common PHA apps, five belonged to various dangerous trojan families and one was a cleaner app that disables SELinux.

Lowest PHA percentage of PHA devices
Japan (0.03%), Finland (0.06%), and Norway (0.07%) are the countries with the lowest PHA prevalence among countries with at least one million Android devices. PHAs in the Scandinavian countries are primarily power-user tools that lower device security levels intentionally while Japan has a mix of power user tools and malicious PHA, albeit at a very low level. The most common PHA there, a trojan app, affected just over 9,000 devices.

Device hygiene of the 5 countries with the lowest percentage of PHA devices in 2017

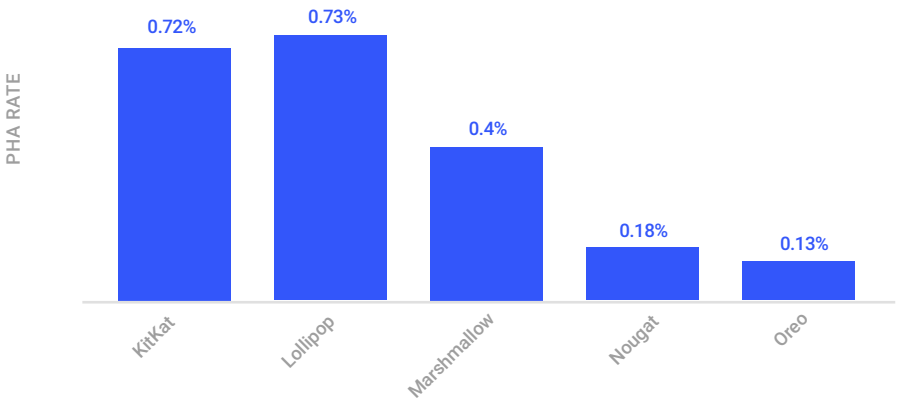


Device hygiene by Android version

Device hygiene by Android version shows that newer versions of Android are much less affected by PHAs. Nougat and Oreo have PHA rates that are a fifth and a tenth of older Android versions, respectively.

This could be attributed to many factors, such as platform hardening, API hardening, and app security and developer training to reduce apps' access to sensitive data. In particular, newer Android versions, such as Nougat and Oreo, are more resilient to privilege escalation attacks that had previously allowed PHA to gain persistence on devices and protect themselves against removal attempts.

Percentage of devices with PHA installed in 2017, by Android version



PHA distribution analysis

The device hygiene metric in the previous section provides a way to track how many devices have installed one or more PHAs. This section focuses on how those PHAs are distributed inside and outside of Google Play.

Many devices install apps from both inside and outside of Google Play, so device hygiene is a blended average of all distribution paths. Device hygiene varies across the Android ecosystem based on the number of apps users install, which ranges from zero to several hundred apps per device. To provide more insight into the root cause of changes in device hygiene, we also analyze individual install events and distribution paths. This section compares PHA installations in 2017 to 2016.

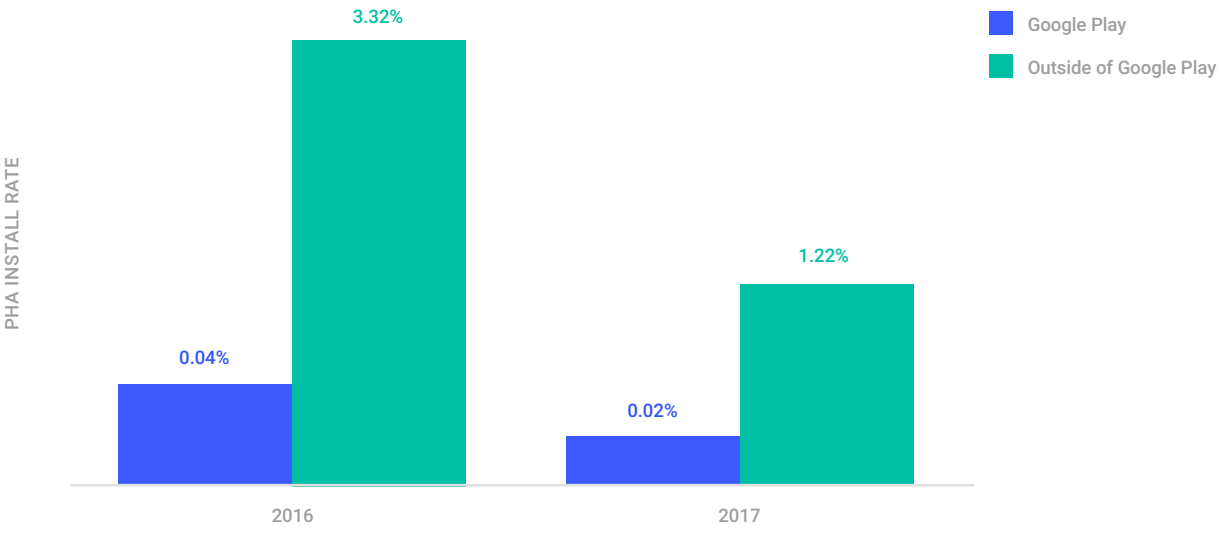
PHA install rates from Google Play was halved from 0.04% in 2016 to 0.02% in 2017 (-50% YoY). PHA install rates from outside of Google Play also showed drastic reduction from 3.32% in 2016 to 1.22% in 2017 (-63% YoY).

Google Play Protect can remove any app identified to have PHA behavior from the Google Play store. For apps outside of Google Play this is not the case. Because of this difference, in addition to actual installs, we also track installation attempts for apps installed from outside of Google Play. Not all installation attempts result in actual installs: if a user heeds the warning that an app is a PHA, they may not install the app. In particularly harmful cases, such as ransomware or banking phishing apps, Google Play Protect just blocks the installation in addition to warning the user.

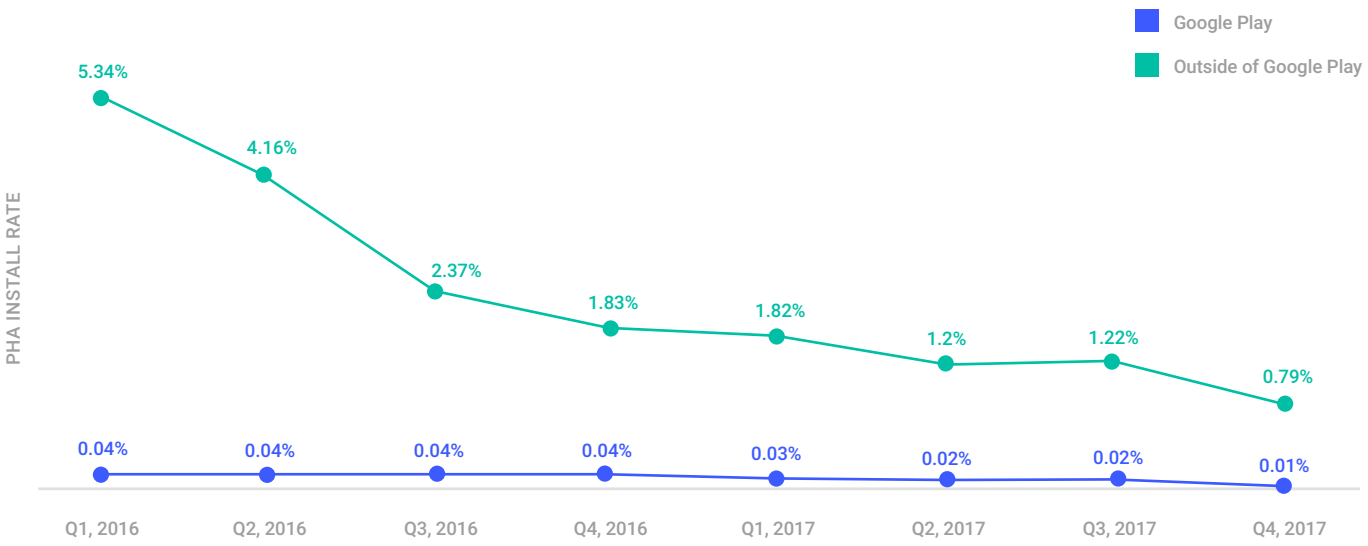
Compared to 2016, the total number of app installation attempts observed by Google Play Protect grew by about 25% in 2017. Installation attempts of legitimate apps grew much faster and installation attempts of PHAs declined. Compared to 2016, PHA installation attempts from outside of Google Play fell by a third.

Outside of Google Play, the PHA installation attempt rate dropped from 8.5% to 4.7% in 2017. We successfully prevented 74% of those PHA installation attempts with warnings and protections from Google Play Protect. Only 1.22% of all app install attempts from outside of Google Play ended with a PHA installation. In 2016, this number was 3.32%.

PHA install rates, 2016 and 2017



PHA install rates, by quarter



Looking just at the effectiveness of Google Play Protect, we see another positive trend. In 2017, Google's mobile threat protection services stopped 74% of all PHA installation attempts from outside of Google Play compared to 55% in 2016. The remaining 26% are a combination of PHAs that were installed before we had identified them as PHAs and users who decided to ignore Play Protect's warning to complete the PHA installation.

> Google Play Protect prevented Android users from installing a PHA from outside of Google Play about 1.6 billion times in 2017.

Google Play: PHA trends

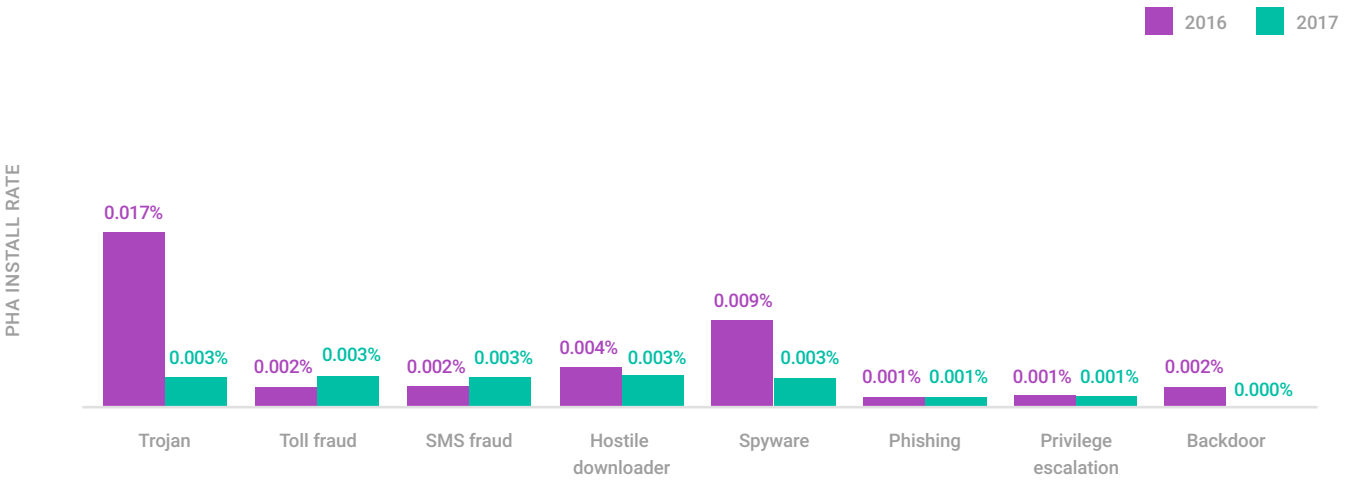
The overall health of Google Play has improved over the years. In 2017, 0.02% of all downloads from Google Play were PHAs. In 2016, the number was nearly double. The biggest contributor to this decline was the 80% drop in trojan installations coming from Google Play compared to 2016. Over the past twelve months, large trojan families were less successful at entering Google Play compared to previous years.

The decrease in trojans had another effect. In previous years, trojans represented a much larger percentage of PHAs in the Android ecosystem. In 2017, trojans barely maintained the lead, as the toll fraud, SMS fraud, and hostile downloader categories accounted for approximately 0.003% of all app downloads from Google Play.

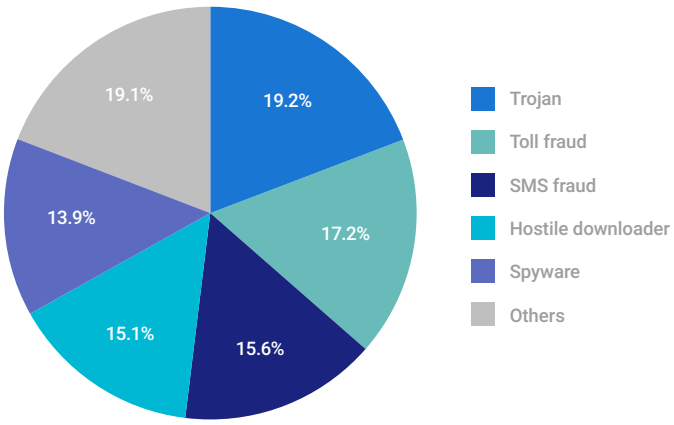
Top PHA categories

These charts show all PHA categories broken down by percentage against other PHAs in 2016 and 2017. For example, in 2016 trojan installs represented 0.017% of all installs and in 2017 they represented 0.003%

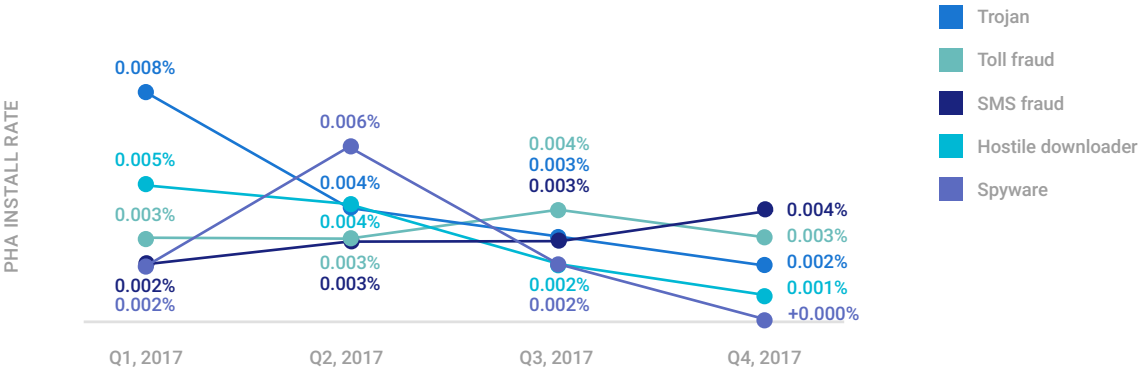
PHA install rate by category in Google Play, 2016 and 2017



PHA categories in Google Play, 2017



Top PHA categories in Google Play, by quarter



Google Play: Trojan

In 2017, trojans remained the top PHA category in Google Play with 0.003% of all app installs (2016: 0.017%) and 19.2% of all PHA downloads from Google Play.

Based on the number of installs, eight of the top ten trojan apps belonged to the Gaiaphish PHA family. This family uses Google account authentication tokens on Android devices to fraudulently manipulate parts of Google Play, such as star ratings or comments. Gaiaphish apps also seems to target other Google properties, such as YouTube or Google+, with fake content.

This report includes more technical details for Gaiaphish in the PHA families section.

Google Play: Toll fraud

Toll fraud was a new top PHA category on Google Play. In 2017, toll fraud PHAs accounted for 0.003% of all app installs (2016: 0.002%) and 17.2% of all PHA installs from Google Play.

Toll fraud abuses payment services provided by mobile carriers to put unwanted charges onto user phone bills without their consent or knowledge. As mobile billing is not standardized, most users who download and run toll fraud apps are not affected. Only users in countries and on mobile carriers that offer direct billing services can be abused by the fraudulent apps.

The rise of toll fraud on Google Play can be attributed to a single app monetization SDK that was included in many low-quality apps before it was found and removed from Google Play.

The company that created the SDK is ostensibly based in Seychelles. However, the website domain has been registered by Chinese companies for years and we assume that the SDK company exists to hide the true origin of the fraudulent code.

Google Play: SMS fraud
In 2017, the third largest PHA category in Google Play by install rate was SMS fraud. SMS fraud apps accounted for 0.003% of app installs (2016: 0.002%) and 15.6% of all PHA installs from Google Play.

No individual family of SMS fraud apps was particularly prevalent among the top SMS apps. The largest among them is the BreadSMS family. This family tricks users through generically branded apps such as camera tools, disk cleaner tools, or other potentially useful apps.

This report includes more technical details for BreadSMS in the PHA families section.

Google Play: Hostile downloader
In 2017, the third largest PHA category in Google Play by install rate was hostile downloaders. Hostile downloaders represent 0.003% (2016: 0.004%) and 15.1% of all PHA installs from Google Play fall into this category.

In late 2017, Google Play policies changed to forbid all dynamic code loading, which affected this PHA category in Google Play. Google Play can now remove apps that have code for dynamic downloading, before we have access to the actual downloaded code.

No individual family of hostile downloaders was particularly prevalent. Among the top hostile downloader apps on Google Play were two Hummingbad apps from January 2017, a device tracker that upgraded itself with commercial spyware features, and an otherwise random assortment of apps. The only commonality is their effort to circumvent Google Play security scanning by moving problematic functionality into a second stage that would later be downloaded and executed on user devices.

Outside of Google Play: PHA trends

Overall, the health of the Android ecosystem for devices downloading apps from outside of Google Play improved. In 2017, the ratio of PHA installs to total installs decreased by about two thirds from 2016. While in 2016, PHA installations made up 3.32% of all apps from outside of Google Play, in 2017 only 1.22% of these apps are PHA.

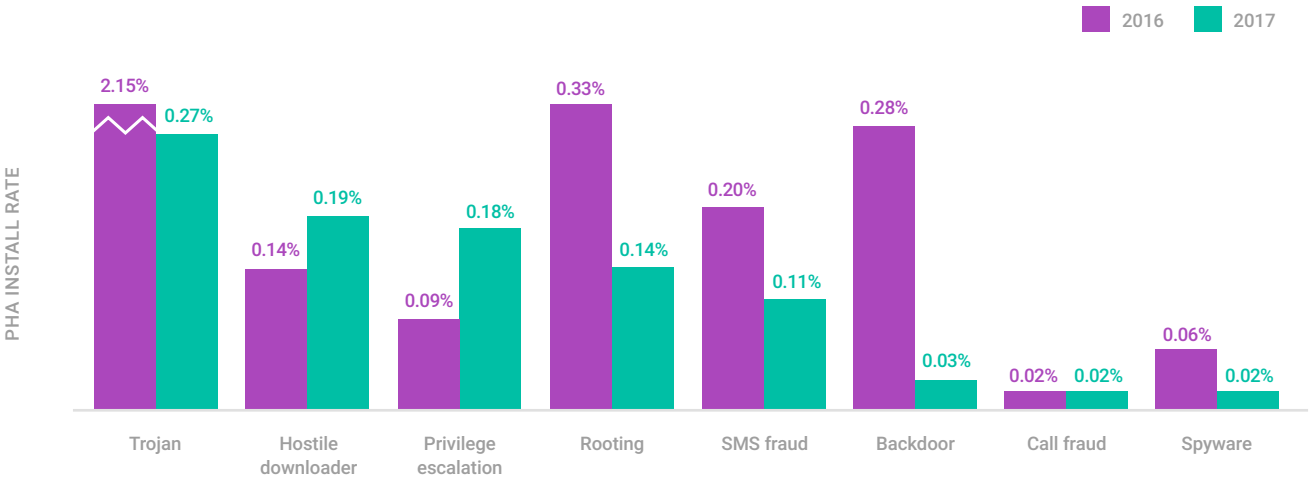
The substantial drop in PHA installs outside of Google Play can be attributed to the disappearance of huge PHA families like Ghost Push, which we have reported on since 2015. These families used to have sophisticated distribution and infection channels that allowed them to infect devices in many different ways. We did not see any PHA families with similar levels of impact in 2017.

The most prevalent PHA categories outside of Google Play looked similar to those inside of Google Play. Along with trojans and hostile downloaders, privilege escalation apps were the most common PHAs.

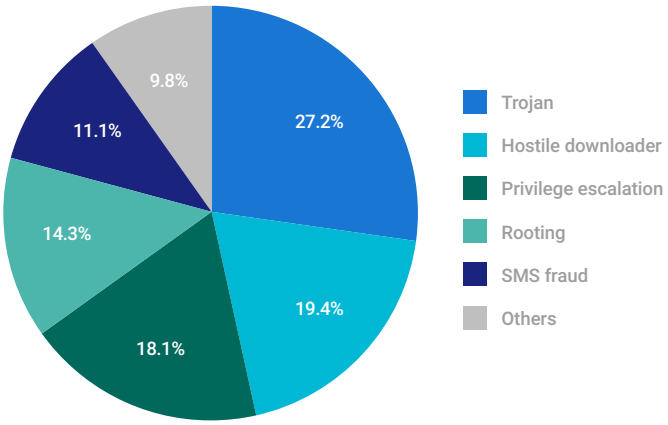
Top PHA categories

These charts show all PHA categories broken down by percentage against other PHAs in 2017. For example, in 2016 trojan installs represented 2.15% of all installs and in 2017 they represented 0.27%.

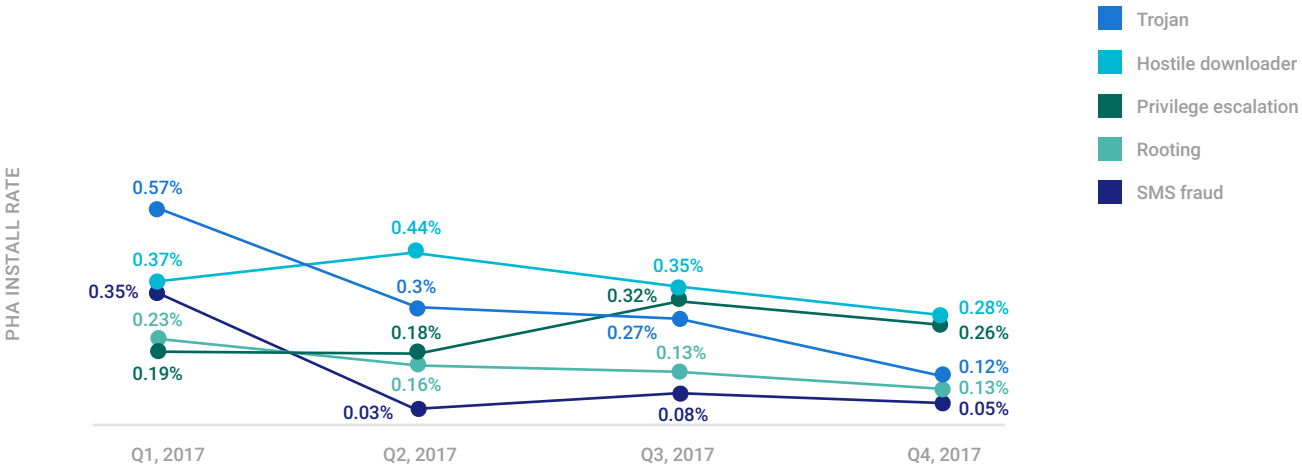
PHA install rate by category outside of Google Play, 2016 and 2017



PHA categories outside of Google Play, 2017



Top PHA categories outside of Google Play, by quarter



Outside of Google Play: Trojan

In 2017, trojans were still the dominant PHA category outside of Google Play where they made up 27.2% of all PHA installs (down from 87.3% in 2016). Compared to 2016, their importance diminished sharply. Year over year, the prevalence of trojans outside Google Play dropped drastically from 2.15% of all app installs to just 0.27%. This huge drop was led by the near disappearance of major trojan families such as Ghost Push, Gooligan, and Hummingbad, which dominated the PHA ecosystem in 2015 and 2016.

In 2017, the only trojan family that saw widespread distribution was KoreFrog, to which 8 of the top 10 most installed trojan PHAs belonged.

This report includes more technical details for KoreFrog in the PHA families section.

Outside of Google Play: Hostile downloader
Up from third place in 2016, hostile downloaders were the second most prevalent PHA category outside Google Play. In 2017, they made up 0.19% of all app installs (2016: 0.14%) and 19.4% of all PHA installs from outside of Google Play.

However, the reason for their prevalence changed. In 2016, hostile downloaders were created as part of the distribution chain of large trojan families. In 2017, the major hostile downloaders were legitimate apps that also downloaded PHAs. As of 2017, our systems categorize hostile downloaders as any app where more than 5% of the apps downloaded are PHAs.

The most popular of these is a device cleaner app from China where up to 30% of app installs it initiates are PHAs. Due to our policy of flagging all apps where more than 5% of downloads are PHA apps, we warn users about this app.

The second most popular app was a Chinese app store that served up to 25% PHA apps. In addition to warning Android users about the apps offered by this store, we worked with the company to remove the most downloaded PHAs from the store.

Outside of Google Play: Privilege escalation
In 2017, privilege escalation apps accounted for 0.18% of all app installs (2016: 0.09%) and 18.1% of all PHA installs from outside of Google Play.

Privilege escalation apps' rise among PHA categories outside of Google Play can be explained by just one app: Lucky Patcher. Lucky Patcher is an app that can be used to circumvent license checks on Android apps for piracy reasons. Because Lucky Patcher contains code for disabling security settings of Android devices, we flag the app as privilege escalation. Most Lucky Patcher users probably install the app knowingly and intentionally, but we feel it is important to remind users of the security trade-offs presented by the app.

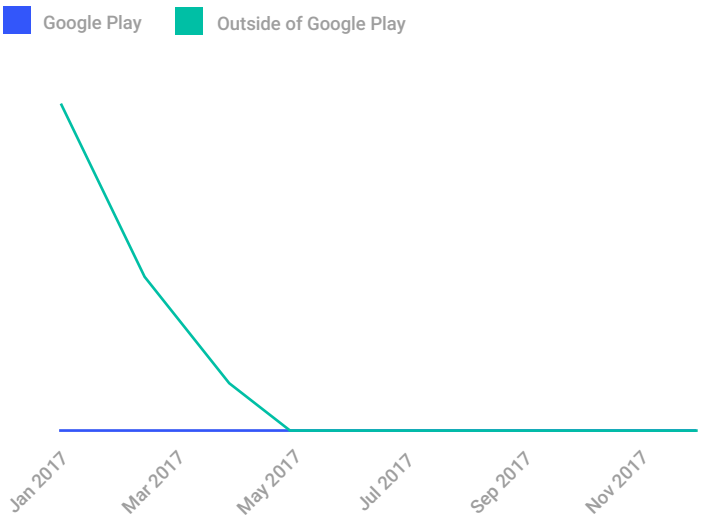
PHA family highlights

Chamois

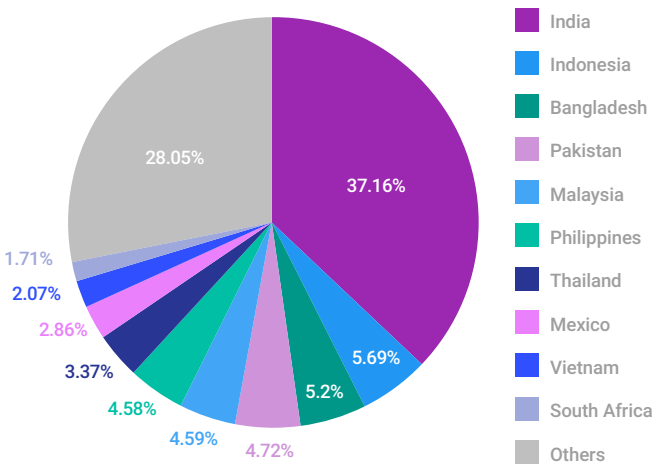
Chamois is one of the largest PHA families in Android to date and is distributed through multiple channels. While much of the backdoor version of this family was cleaned up in 2016, a new variant emerged in 2017. To avoid detection, this version employs a number of techniques, such as implementing custom code obfuscation, preventing user notifications, and not appearing in the device's app list. Chamois apps, which in many cases come preloaded with the system image, try to trick users into clicking ads by displaying deceptive graphics to commit WAP or SMS fraud.

India had the most Chamois downloads in 2017 (25% of the total) and practically all downloads came from outside of Google Play. For more details, see our blog post on detecting and eliminating Chamois.

Chamois installs, by month



Chamois installs, by country



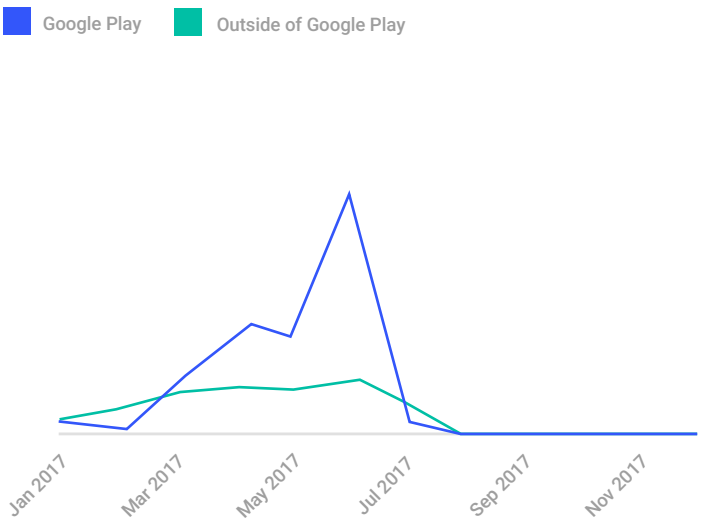
IcicleGum

IcicleGum is a spyware PHA family whose apps rely on versions of the Igexin ads SDK that offer dynamic code-loading support. IcicleGum apps use this library's code-loading features to fetch encrypted DEX files over HTTP from command-and-control servers. The files are then decrypted and loaded via class reflection to read and send phone call logs and other data to remote locations.

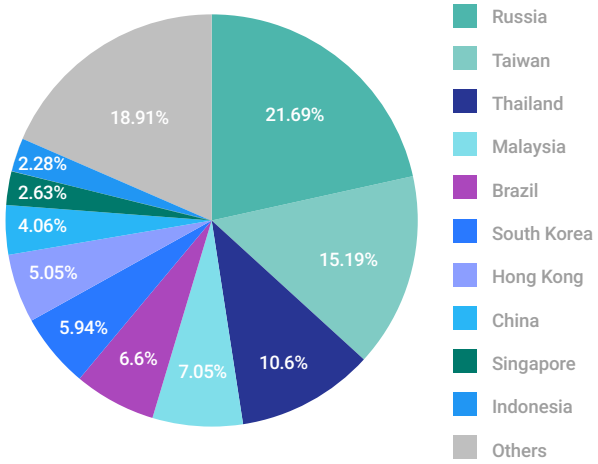
Some developers who include Igexin in their apps may be unaware of the SDK's risks and may not be in control of the malicious code being loaded and executed—this is dynamically decided by the command-and-control server based on configuration information received from each device.

Russia had the most IcicleGum downloads in 2017 (21%+ of the total) and 70% of all IcicleGum downloads came from Google Play.

IcicleGum installs, by month



IcicleGum installs, by country

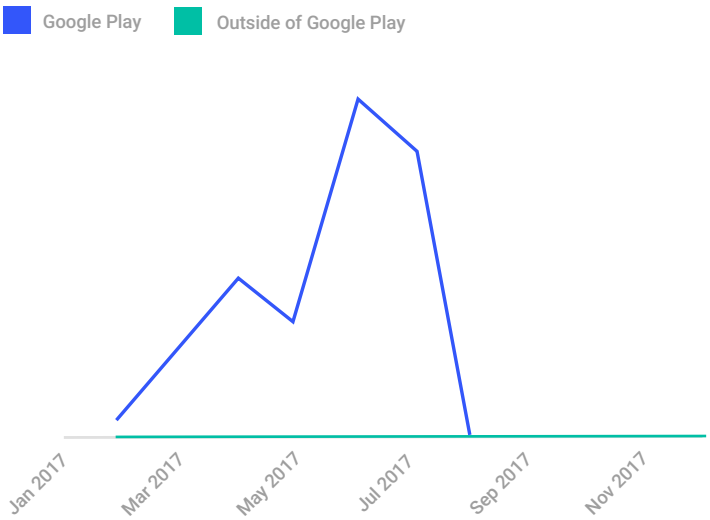


BreadSMS

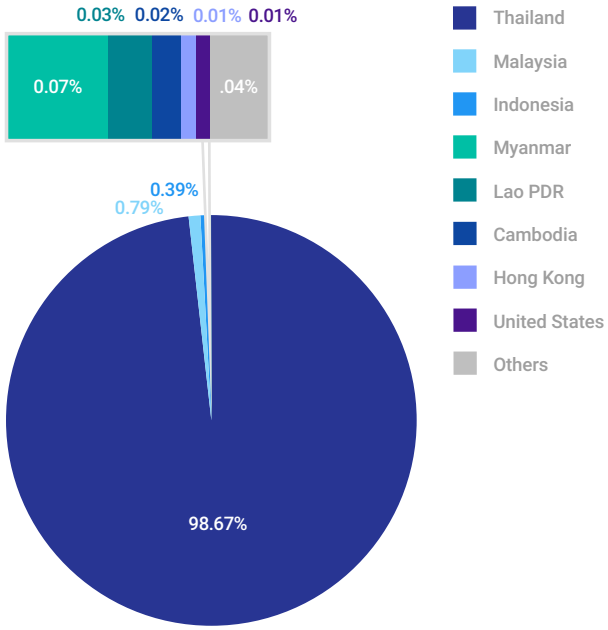
BreadSMS is a large SMS-fraud PHA family that we started tracking at the beginning of 2017. These apps compose and send text messages to premium numbers without the user's consent. In some cases, BreadSMS apps also implement subscription-based SMS fraud and silently enroll users in services provided by their mobile carriers. These apps are linked to a group of command-and-control servers whose IP addresses change frequently and that are used to provide the apps with premium SMS numbers and message text.

Thailand had the most BreadSMS downloads in 2017 (98% of the total) and most downloads came from Google Play.

BreadSMS installs, by month



BreadSMS installs, by country

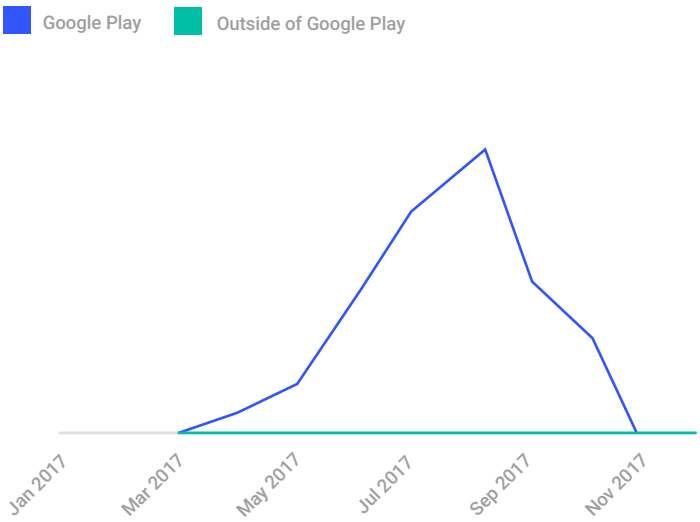


JamSkunk

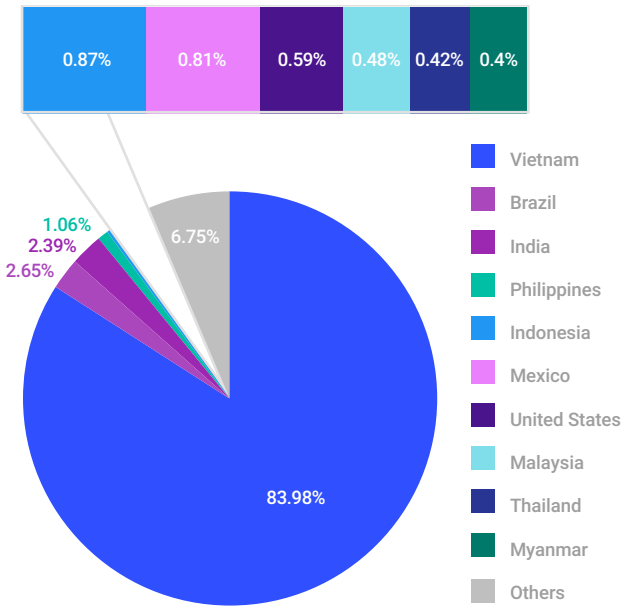
JamSkunk is a toll-fraud PHA family composed of apps that subscribe users to services without their consent. These apps disable Wi-Fi to force traffic to go through users' mobile data connection and then contact command-and-control servers to dynamically fetch code that tries to bypass the network's WAP service subscription verification steps. This type of PHA monetizes their abuse via WAP billing, a payment method that works through mobile data connections and allows users to easily sign up and pay for new services using their existing account (i.e., services are billed directly by the carrier, and not the service provider; the user does not need a new account or a different form of payment). Once authentication is bypassed, JamSkunk apps enroll the device in services that the user may not notice until they receive and read their next bill.

Vietnam had the most JamSkunk downloads in 2017 (83% of the total) and most downloads came from Google Play.

JamSkunk installs, by month



JamSkunk installs, by country

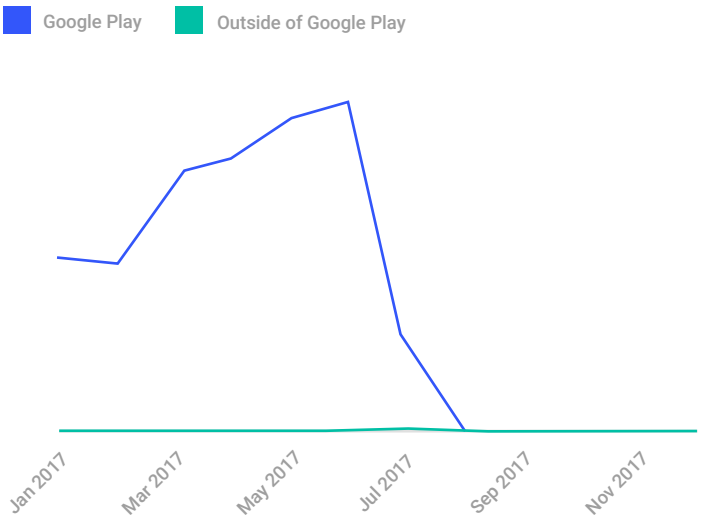


Expensive Wall

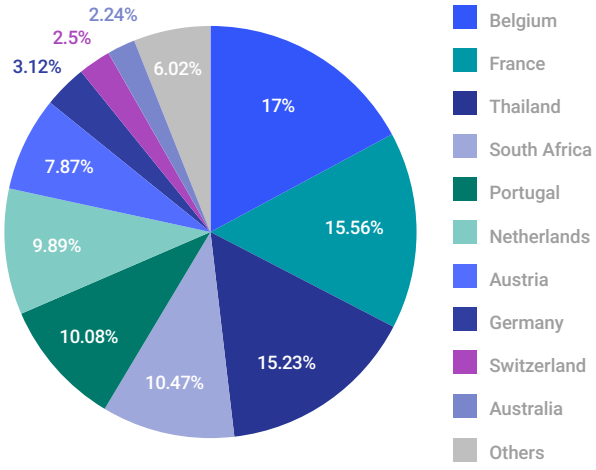
Expensive Wall is a family of SMS-fraud apps that affected a large number of devices in 2017. Expensive Wall apps use code obfuscation to slow down analysis and evade detection, and rely on the JS2Java bridge to allow JavaScript code loaded inside a Webview to call Java methods the way Java apps directly do. Upon launch, Expensive Wall apps connect to command-and-control servers to fetch a domain name. This domain is then contacted via a Webview instance that loads a webpage and executes JavaScript code that calls Java methods to compose and send premium SMS messages or click ads without users' knowledge.

Belgium and France had the most Expensive Wall downloads (17% and 16% of the total, respectively) and most downloads came from Google Play.

Expensive Wall installs, by month



Expensive Wall installs, by country

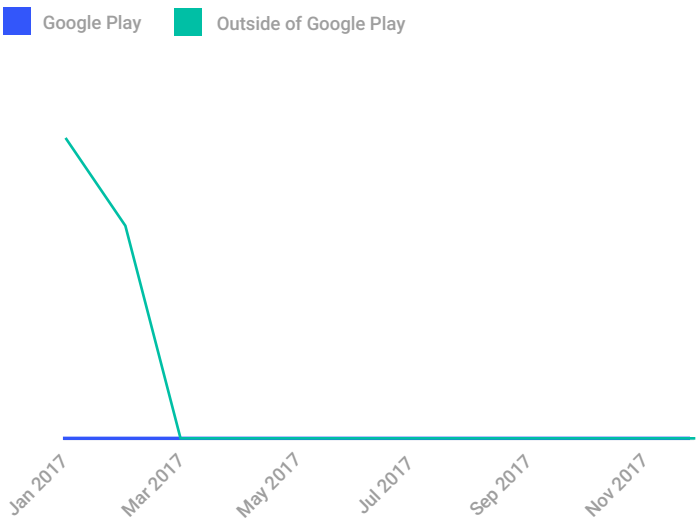


BambaPurple

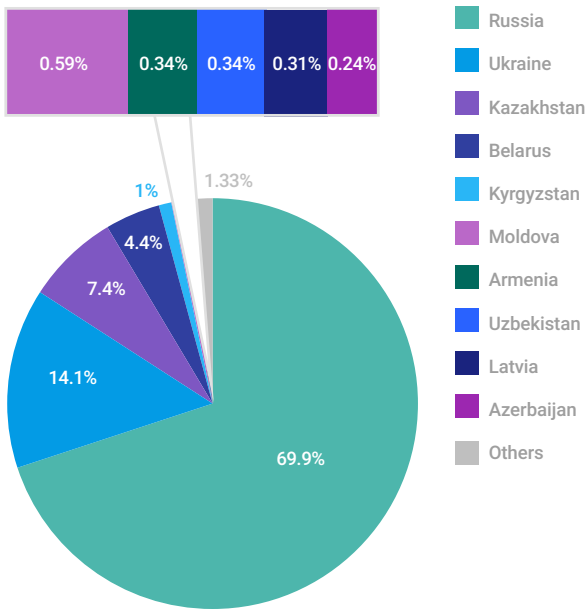
BambaPurple is a two-stage toll-fraud PHA family that tries to trick users into installing it by disguising itself as a popular app. After install, the app disables Wi-Fi to force the device to use its 3G connection, then redirects to subscription pages without the user's knowledge, clicks subscription buttons using downloaded JavaScript, and intercepts incoming subscription SMS messages to prevent the user from unsubscribing. In a second stage, BambaPurple installs a backdoor app that requests device admin privileges and drops a .dex file. This executable checks to make sure it is not being debugged, downloads even more apps without user consent, and displays ads.

Russia had the most BambaPurple downloads in 2017 (63% of the total) and all downloads came from outside of Google Play.

BambaPurple installs, by month



BambaPurple installs, by country

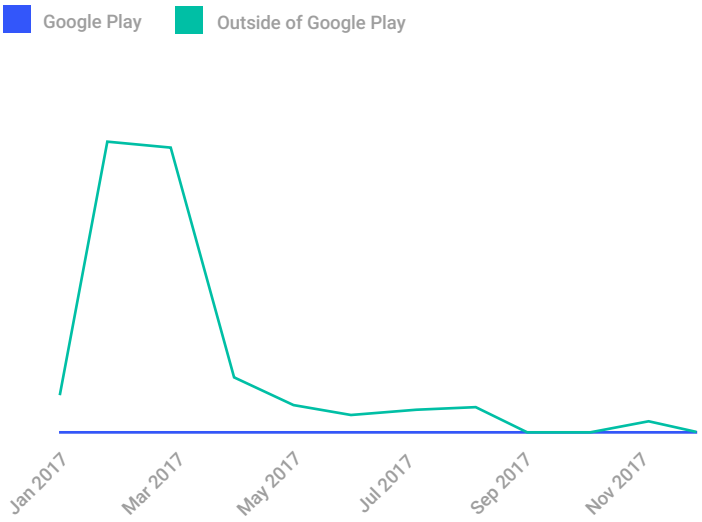


KoreFrog

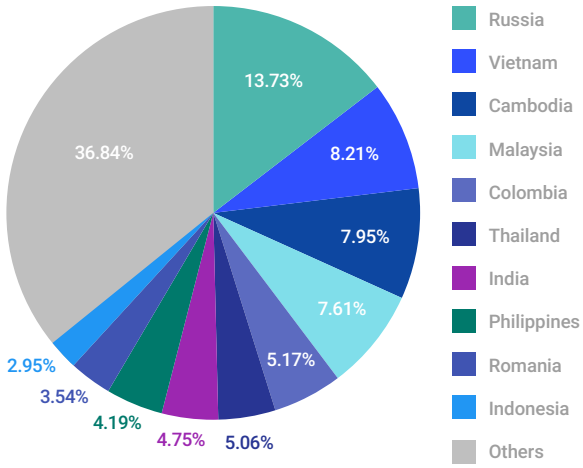
KoreFrog is a family of trojan apps that request permission to install packages and push other apps onto the device as system apps without the user's authorization. System apps can be disabled by the user, but cannot be easily uninstalled. KoreFrog apps operate as daemons running in the background that try to impersonate Google and other system apps by using misleading names and icons to avoid detection. The KoreFrog PHA family has also been observed to serve ads, in addition to apps.

Russia and India had the most KoreFrog downloads in 2017 (11% and 7% of the total, respectively) and most downloads came from outside of Google Play.

KoreFrog installs, by month



KoreFrog installs, by country

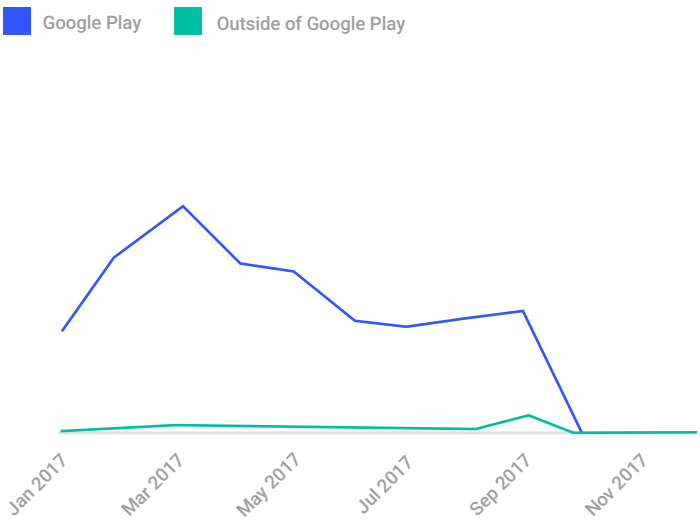


Gaiaphish

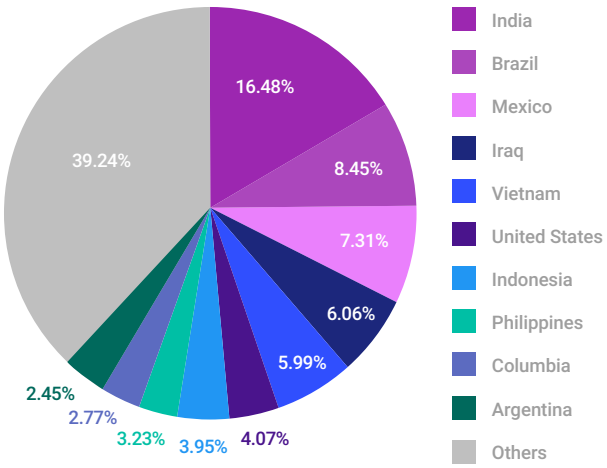
Gaiaphish is a large family of trojan apps that target authentication tokens stored on the device to abuse the user's privileges for various purposes. These apps use base64-encoded URL strings to avoid detection of the command-and-control servers they rely on to download APK files. These files contain phishing apps that try to steal GAIA authentication tokens that grant the user permissions to access Google services, such as Google Play, Google+, and YouTube. With these tokens, Gaiaphish apps are able to generate spam and automatically post content (for instance, fake app ratings and comments on Google Play app pages).

India had the most Gaiaphish downloads in 2017 (23% of the total). Installs came from Google Play as well as from outside of Google Play (51%, 49%, respectively).

Gaiaphish installs, by month



Gaiaphish installs, by country



Acknowledgments

This report—and all of the hard work that it represents—isn't the product of a single team or company.

Thank you to the Google teams, our Android partners, and the external researchers who contribute to the security of Android throughout the year.

Your hard work, effort, and commitment to security makes the entire Android ecosystem safer and protects Android users across the world.

android