Android Security 2016 Year In Review

March 2017

android

Contents

Overview

Google is committed to protecting the security and privacy of all Android users. Keeping more than 1.4 billion devices safe starts with a strong foundation—the core Android platform—which is strengthened by regular security updates for the platform, applications, and devices and constantly evolving security services that monitor and protect the ecosystem.

In 2016, Google worked closely with device manufacturers, system on a chip (SoC) providers, and telecom carriers to release security patches to more devices than ever before. We made key security features like data encryption and verified boot the standard for over one hundred million users. In addition to making devices more secure, we actively protected users from application threats by reducing the impact of Potentially Harmful Applications (PHAs) inside and outside of Google Play and improving the quality of security in hundreds of thousands of applications. Overall, devices, apps, and users are safer than ever.

Looking forward to 2017, we're working to increase the number of patched Android devices and accelerate adoption of key platform security features. We believe that advances in machine learning and automation can help reduce PHA rates significantly in 2017, both inside and outside of Google Play.

This is Google's third annual report on Android's security protections. The report covers new and updated features, provides metrics that informed our view of Android security, and discusses trends around security for Android devices in 2016.

Google security services for Android

Devices with Google Mobile Services (GMS) are protected straight out of the box by a complete set of endpoint security and antivirus services. This set includes both cloud-based and pre-installed on-device services that use

real-time data from the Android ecosystem to understand the security environment. Because Google's security services generally don't require firmware or platform-level patches to update, they provide a first line of defense against evolving security threats.

> By Q4 2016, fewer than 0.71% of devices had Potentially Harmful Applications (PHAs) installed and for devices that exclusively download apps from Google Play, that number was even smaller at 0.05%.

These small numbers are thanks in part to Google's responsive security services.

Google regularly enhances its security services for Android. In 2016, we used machine learning and statistical analysis to further automate and speed up detection of PHAs and other threats. Enhancements to the Safe Browsing service, which protects users from phishing sites and websites hosting malware, improved PHA device-scanning capabilities and enabled third-party developers to leverage the power of Safe Browsing in their own applications. Third-party developers took advantage of the security services offered through SafetyNet APIs, such as SafetyNet Attest, which serves nearly 200 million requests per day.

Android platform security

All Android devices share a common, platform-level security model. This model has been enhanced over multiple years with SELinux protections, application isolation using sandboxing, exploit mitigations, and cryptographic features, like file-based encryption and Verified Boot.

In 2016, Android expanded platform-level security with the launch of Android 7.0. We streamlined our boot-up process to make it easier to install over-the-air (OTA) security updates. To support this faster boot up, we implemented file-based encryption, which also better isolates and protects individual users and profiles on a device. We re-architected the mediaserver stack to address Stagefright-type vulnerabilities by adding integer overflow protections and compartmentalizing mediaserver's components into individual sandboxes with minimal privileges. We also increased the degree of randomness in address space layout randomization (ASLR), making some attacks more difficult.

Ecosystem security programs

Android promotes security best practices in a variety of ways. The Android Compatibility Definition Document (CDD) and Compatibility Test Suite (CTS) provide a detailed series of security requirements and a testing framework to verify compatibility. Google works with device manufacturers to keep devices secure and quickly adopt security updates and features on all supported devices.

Google Play encourages application developers to adopt security best practices. We launched 18 campaigns to notify application developers about vulnerabilities or recommended security improvements in their apps in Play, resulting in security upgrades to over 275,000 apps.

As promised in 2015, we released monthly security bulletins and patches to the Android Open Source Project (AOSP). We worked closely with device manufacturers, SoC providers, and carriers to ship security updates, and introduced a freshness test for security patch levels in CTS. By Q4 2016, over half of the top 50 devices worldwide had a recent security patch.

> Several manufacturers, including Samsung, LG, BlackBerry, and OnePlus, regularly deliver security updates to flagship devices on the same day as Google's updates to Nexus and Pixel devices, thereby providing their customers with the most up-to-date security available.

Openness strengthens security

Android has been open source since its launch. Because all Android source code is publicly available, individuals and companies can create their own versions of Android and even add security features.

Open source code means that Android is subject to more scrutiny and creates more opportunities for research. We consider this a strength of the platform as it allows security researchers to directly examine the code for weaknesses. To encourage this, Google offers a security bug bounty program for researchers who find and report vulnerabilities to Google. This allows us to fix the reported vulnerabilities and improve the overall health of Android devices. In this way, Android leverages the expertise of the security community as a whole.

Over 100 security researchers made public contributions to Android in 2016, for a total of nearly 1 million dollars in security rewards.

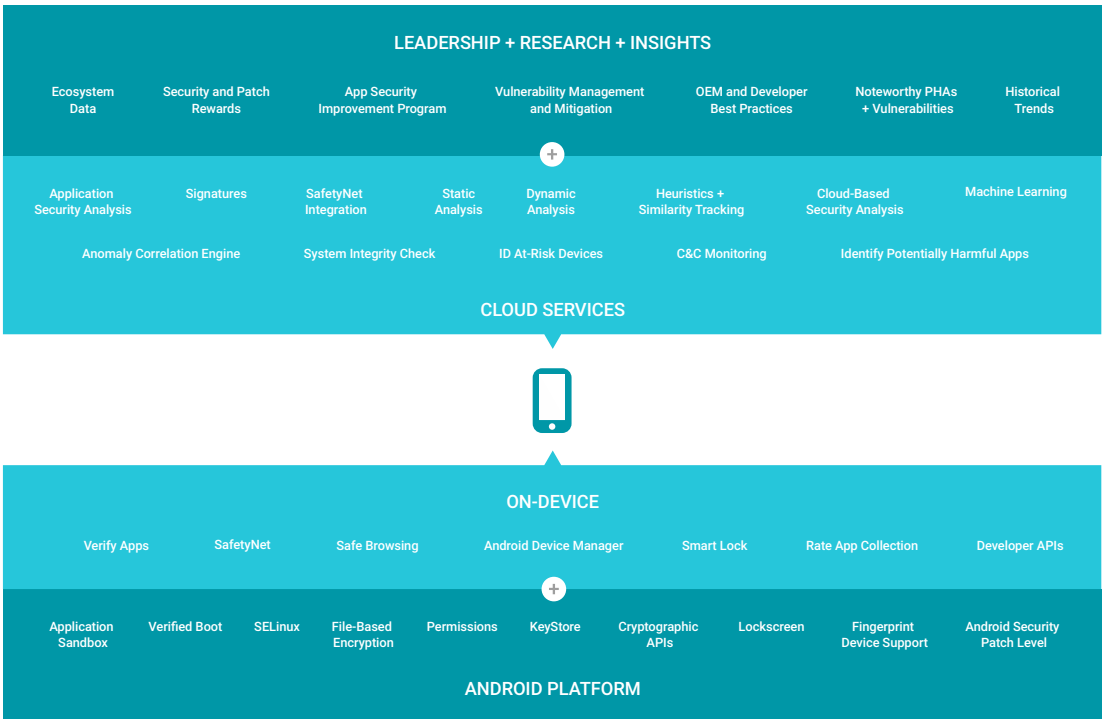
We continue to iterate and innovate upon Android's security features. In 2016, we protected Android users—both on and off their devices—by improving our cloud security services, updating the Android platform, and investing in our ongoing ecosystem security programs.

Google Security Services for Android

Google protects the Android ecosystem with pre-installed cloud-based and on-device services, providing multiple layers of security protections to devices. All devices with GMS have a complete set of endpoint and antivirus services that protect against common threats including network attacks, application exploits, Potentially Harmful Applications, and physical attacks, such as device theft.

Google's security services for Android can be updated independently of device or carrier implementations. This autonomy facilitates quick responses to emerging security threats, allowing us to block or minimize their impact of newly discovered vulnerabilities.

This diagram shows the range of different security services and technologies provided by Google for Android.



In 2016, Google's security services conducted over 790 million device security scans daily, protecting Android phones, tablets, smartwatches, and TVs. The goal is to provide the right protection at the moment it is needed by the user.

On-device services

This table lists the on-device protections offered in 2016, along with a brief description of their roles in device and/or data protection.

Service	Protection
Verify Apps	Antivirus protection and removal options for downloaded PHAS
SafetyNet	Protection from network and application-based threats
Safe Browsing	Protection from deceptive websites
Developer APIs	Allows third-party applications to use Google's security services
Android Device Manager	Protection for lost or stolen devices
Smart Lock	Encourage lock screen adoption by reducing friction around device unlock

All of these services integrate with a cloud-based component that allows Google to push updates to the device.

In the following section, we provide a description of these services, along with new features and improvements made to these on-device protections in 2016.

Verify Apps

Verify Apps uses a cloud-based service to determine if applications are potentially harmful. It scans applications before installation and blocks installs of PHAs. It also runs regular scans on all installed apps. If a PHA is found, Verify Apps prompts the user to remove it. In cases where the PHA has no possible benefit to users, Verify Apps can remove the PHA from affected devices with a notification to the user. Future installs of the PHA will be blocked.

In 2016, we made Autoscan faster and more robust. While all devices are scanned at least once every 6 days, devices with indicators of installed PHAs or other risk factors are scanned more frequently. This feature leverages the new Safe Browsing API on Android that pushes information to devices when new risk indicators are found. If the device matches a risk indicator, then Verify Apps starts a full scan to check that all installed apps are behaving in a safe manner.

Rare app collection

Verify Apps protects users against applications that are installed from any source—whether they come from Google Play or not—so it is important that our systems understand as many applications as possible. All applications that are submitted to Google Play undergo a review prior to publication. Similarly, Google's cloud-based systems review all applications they can find on public websites.

Users can send applications directly from their device to Google for review by enabling the "Improve harmful app detection" feature in Settings. The more applications that Verify Apps analyzes, the more accurate it is at identifying PHAs. In 2016, approximately 1.8 million rare applications were uploaded by Verify Apps, up 87% from 2015.

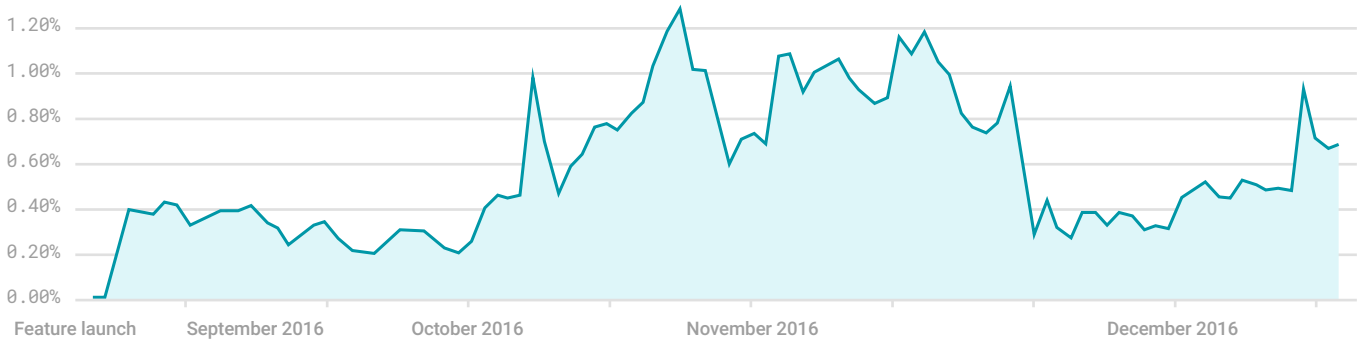
Harmful secondary installations

Some harmful apps attempt to install other applications without user knowledge or consent. These applications can be benign, but 37% of the time they are a PHA. To address this, we updated Verify Apps to automatically block install attempts initiated by an installed PHA in September 2016.

> Verify Apps blocks between 0.4% and 1.2% of all secondary install attempts each day and prevents PHAs from benefitting from these potential secondary app installs.

This chart shows the trend of blocked installation attempts made by PHAs as a portion of all app installs.

Blocked harmful secondary install attempts



SafetyNet

In 2013, we introduced 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 information. Before 2016, only users that installed apps from unknown sources were prompted to enable SafetyNet's protection. In 2016, SafetyNet is enabled by default on all Android devices with Google Play; users can still opt out of SafetyNet's extended protection in Settings.

SafetyNet integrations

In addition to the changes to SafetyNet's default settings for consumer protection, we also updated its APIs and documentation to encourage developer and enterprise adoption. The SafetyNet Attestation API, launched in 2015, helps developers assess the security and compatibility of the Android environments in which their apps run. It determines the integrity of the device and the application, and is commonly used as a signal in anti-abuse systems.

In 2016, we added the basicIntegrity field to the API response to help developers assess a broader spectrum of devices than the existing ctsProfileMatch. If SafetyNet Attestation returns true for basicIntegrity, then the device exhibits the properties of a functional Android device with a working security model, though it might not pass Android compatibility testing. If ctsProfileMatch is also true, then the profile of the device running the developer's app matches the profile of a device that has passed Android compatibility testing (CTS). Device manufacturers submit their CTS test results to Google as part of the certification process for Google's applications; Google believes that devices that return `ctsProfileMatch` fulfill Android's security and compatibility requirements.

The SafetyNet Attestation API gathers information about the state of devices globally. This table shows the percentage of devices that match an unaltered CTS profile certified by Google (`ctsProfileMatch`) and devices that report passing the basic integrity checks (`basicIntegrity`) for the 20 countries with the largest number of active users of Google Play.

Country	CTS profile match	Basic integrity
Argentina	85%	91%
Brazil	93%	96%
Canada	92%	94%
France	92%	96%
Germany	93%	95%
Great Britain	94%	97%
India	86%	96%
Indonesia	79%	89%
Italy	90%	95%
Japan	97%	97%
Korea	97%	97%
Mexico	82%	91%
Russia	80%	93%
Saudi Arabia	90%	94%
Spain	83%	90%
Taiwan	94%	95%

Country	CTS profile match	Basic integrity
Thailand	65%	95%
Turkey	79%	87%
United States	94%	96%
Vietnam	79%	89%

To make integration easier for developers, we published updated documentation and released sample code for Android and server-side verification on GitHub. This continued effort in developer advocacy resulted in SafetyNet attestation adoption by major entertainment, enterprise, and financial applications. SafetyNet attest served nearly 200 million requests per day in 2016, an increase of about 25% over 2015.

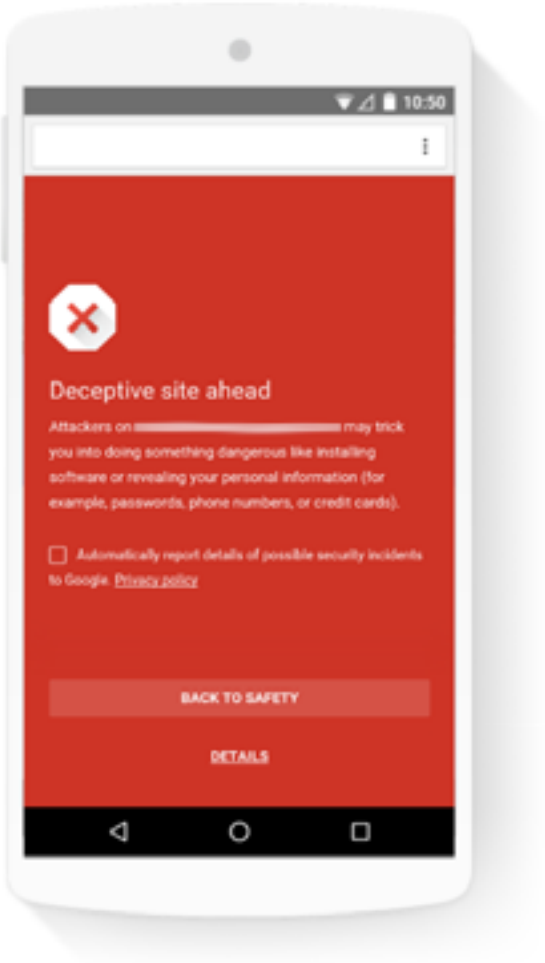
Checking device certification with Google Play
In late 2016, we updated the Google Play Store app to show whether a device is certified by Google when the device preloads Google applications. Users, retailers, carriers, and device manufacturers can see a device's certification status in Play Settings.

Safe Browsing

Google introduced Safe Browsing in 2005. Safe Browsing protects users against threats by allowing client applications to check URLs against lists of unsafe web resources, such as social engineering sites (phishing and deceptive sites) and sites that host malware or unwanted software. When a user attempts to visit an unsafe web resource, their Safe Browsing-supported browser displays a warning.

Approximately a billion users take advantage of Safe Browsing every day. For every million page views across all platforms, Safe Browsing shows around 125 warnings: 80% of which are phishing or social engineering and 20% are malware.

Safe Browsing warning



Safe Browsing protects Chrome desktop users, as well as other popular desktop web browsers. In December 2015, Google Play Services incorporated an API that extended Safe Browsing's protections to the Chrome browser on Android devices.

In mid-2016, we released the Safe Browsing API to third-party developers, which allows their apps to use Safe Browsing's database of known harmful URLs with little additional work on their part.

This allows all apps to use the same protections as the Chrome Browser while being considerate of the user's data plan, network bandwidth, and privacy.

Safe Browsing sometimes flags legitimate websites that have been taken over by a hostile attacker. Once these legitimate websites remove the harmful code and are restored to a safe state, Safe Browsing removes the warning. Some harmful websites take advantage of this by temporarily removing the harmful behavior to get the warning lifted. Once the warning is removed the website reinstates the harmful behavior.

To mitigate these tactics and better protect our users, we adjusted our policies to classify these sites as Repeat Offenders, in 2016. Repeat Offenders are websites that switch between compliant and policy-violating behavior to obtain a successful review and have warnings removed. Repeat Offender websites receive a Safe Browsing warning for at least 30 days and the site's webmaster cannot request a review to remove the warning until the 30 days has passed.

Android Device Manager

User data is more often at risk from lost or stolen devices than from PHAs. To help solve this, Google introduced the Android Device Manager (ADM) service in 2013. Users can find their lost device by using the ADM website or downloading the ADM app to a different Android device. With either approach, users can see their device's location, make it ring, set up a lock screen, or wipe all their personal data and accounts from their device.

ADM is available to all Android users who sign into their Google accounts on their devices. Users who also enable location services can find their devices with ADM. ADM is enabled by default on devices running Android 4.4 and above. In 2016, we improved ADM by:

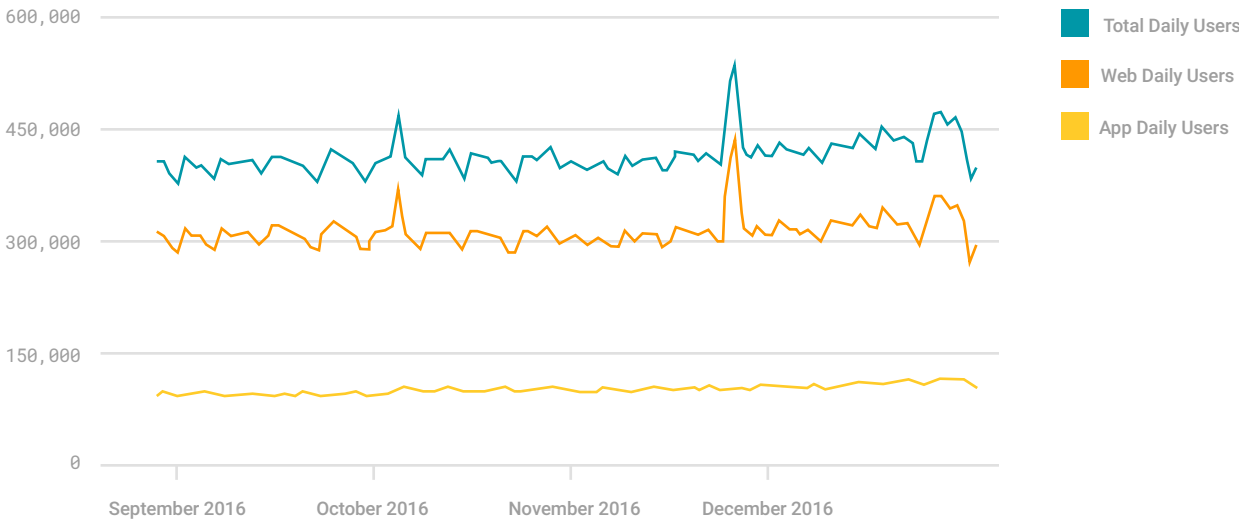
— Translating the ADM website into 31 additional languages, for a total of 79.
— Releasing the ADM app to Android Wear devices, so users can use their watch to find their phone.

Most users access ADM by going to the website or searching for the phrase "find my phone."

> In 2016, approximately 380,000 people used ADM to find their phones each day.

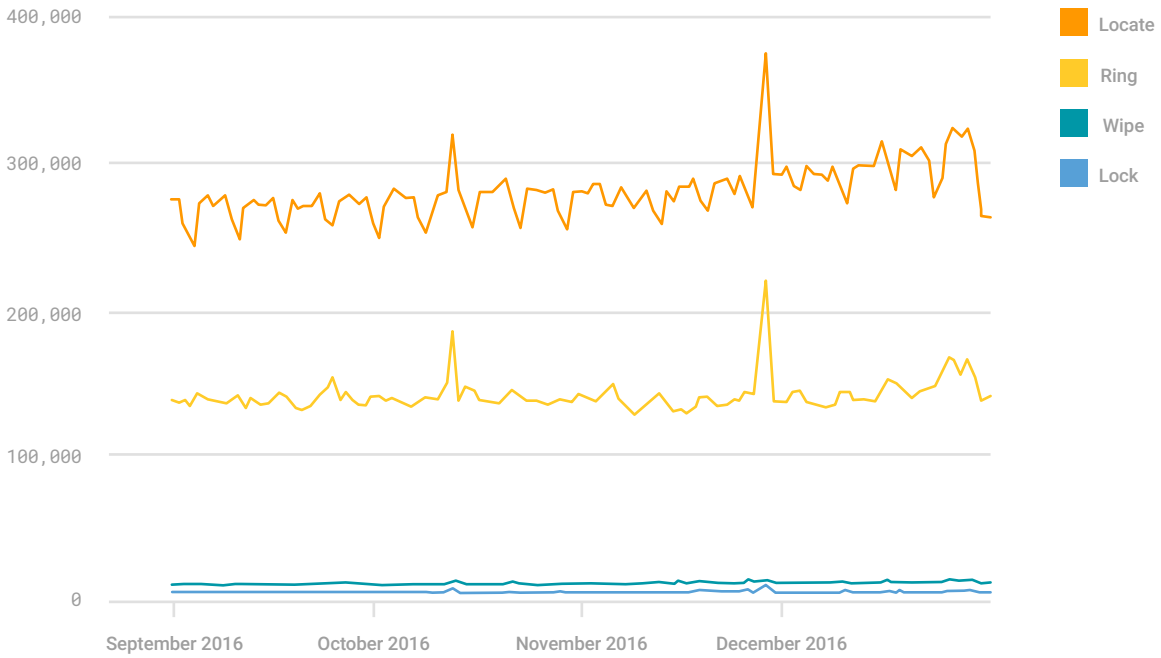
We started saving ADM usage data in September 2016. This graph shows ADM trends since then.

Android Device Manager—Daily users



Locate and Ring are the most commonly used ADM features. Far fewer users take protective measures to lock or wipe their devices, suggesting that most users are able to recover their lost devices.

Android Device Manager—Actions



To perform any of these actions, the device must be on and have network access. ADM contacts the phone in real time to determine its location. If the device is off, location services are off, or it can't connect to the network, ADM won't be able to find it. Improving protections for lost or stolen devices that are not able to connect to the network, such as by strengthening device encryption, is an active area of research and development.

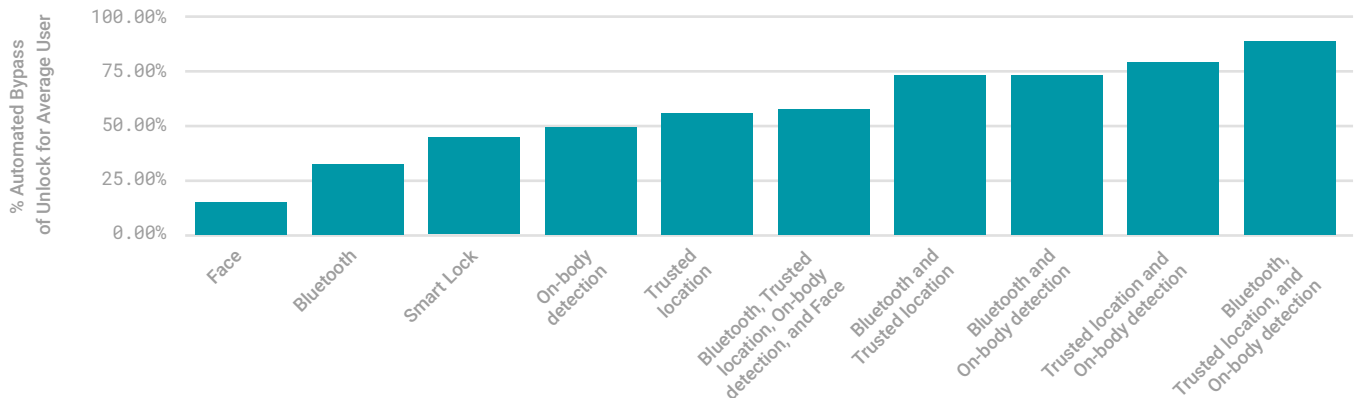
Smart Lock

Lock screens greatly increase user privacy and security. Many users choose not to use a lock screen because manually unlocking their device dozens or even hundreds of times a day is frustrating. In 2014, Android 5.0 introduced Smart Lock, which allows a user's device to remain unlocked as long as it is in their possession. This is determined by certain security signals, such as facial recognition; trusted places, like a user's home or office; and paired Bluetooth devices, such as a smartwatch or car. Enterprises can manage these security signals via API to suit their IT policy.

Smart Lock added other security signals including voice recognition and on-body detection, which keep the phone unlocked while it's on the user's body. Devices running Android 7.0 and above prompt users to set a lock screen and enable Smart Lock's on-body detection to remove the friction of entering a PIN or password. This reduces the number of times that a user needs to manually unlock their device and encourages adoption of a more secure lock screen.

Smart Lock users manually unlock their device about half as often as before they enabled the feature. And users who configure Smart Lock with multiple unlock mechanisms experience even better results—the combined use of trusted Bluetooth devices, trusted places, and on-body detection reduces the number of manual unlocks by about 90%. In 2016, Smart Lock's daily active users increased nearly 175% over 2015.

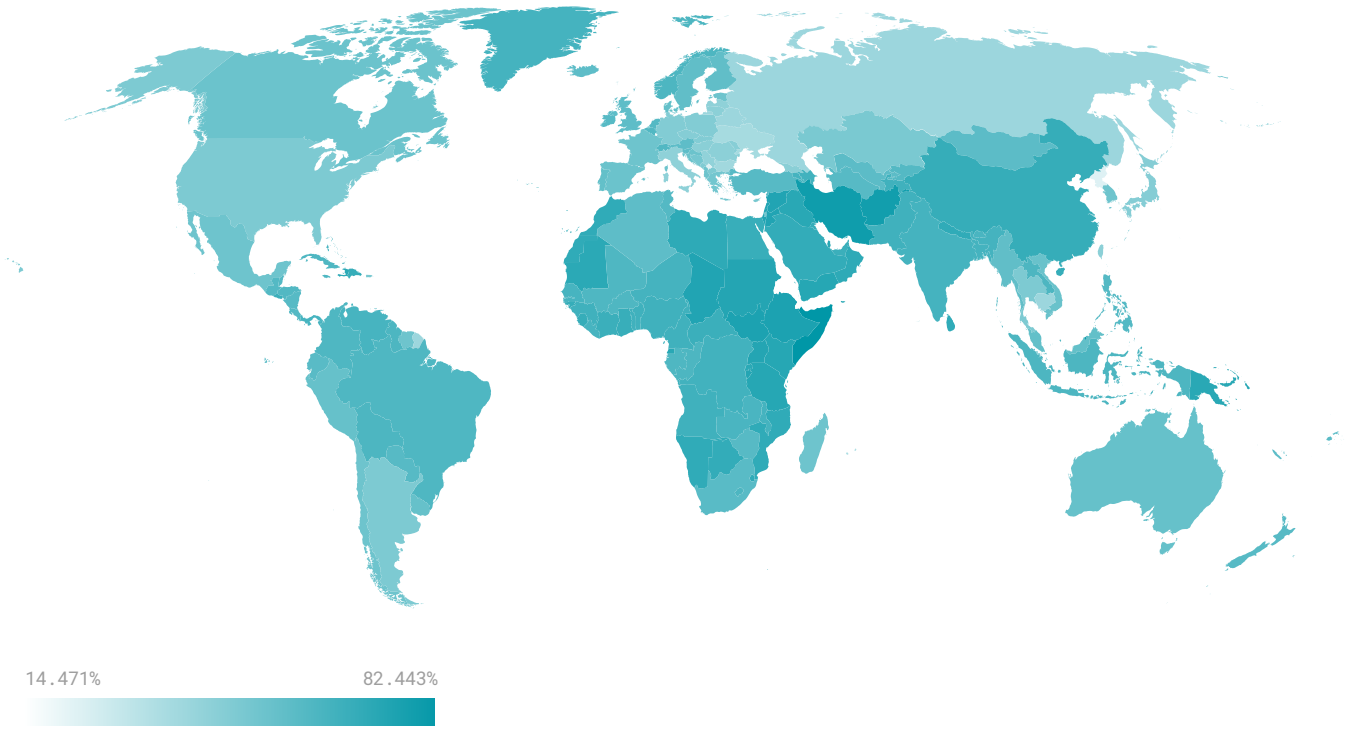
Smart Lock—Manual unlock reduction, by type



Adoption of Secure Lockscreen

Worldwide, 48.9% of devices—across all form factors—have enabled a secure lockscreen, such as a PIN, pattern, password, or other unlock mechanism. This map shows secure lock screen adoption by country. Somalia has the highest adoption at 82%, followed by Samoa and Iran at 78% and 77%, respectively. On the other end of the spectrum, countries with the lowest rate of lock screen adoption include San Marino (14.47%), Ukraine (27.10%), and Bulgaria (28.98%).

Lock screen adoption, by country



Cloud-based security analysis

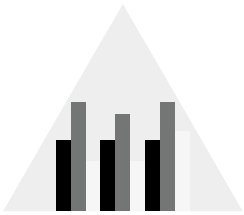
The Android ecosystem includes over 1.4 billion devices. Google applies large-scale analysis to a sizable pool of data to identify signals that indicate potential abuse or security concerns. This section describes additions and updates to analysis capabilities in 2016.

Android application security analysis

Before applications become available in Google Play they undergo an application review process to confirm they comply with Google Play policies. Google has developed an automated application risk analyzer that performs

static and dynamic analysis of APKs to detect potentially harmful app behavior. When Google's application risk analyzer discovers something suspicious, it flags the offending app as a PHA, and refers the PHA to a security analyst for manual review if needed.

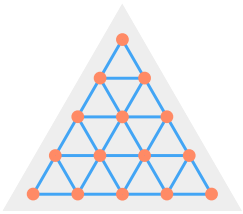
Here are some of the ways that our machines learn what is good and what is bad:



Static analysis
We analyze application code without running the app. Application features are extracted and analyzed against expected good behavior and potential bad behavior.

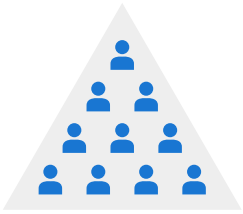
Dynamic analysis
We run applications to identify interactive behavior that cannot be seen with static analysis. This allows reviewers to identify attacks that require connection to a server and dynamic downloading of code.

Third-party reports
We cultivate active relationships with industry and academic security researchers. These independent security researchers also evaluate applications in a variety of ways and will often let us know if they see something amiss.

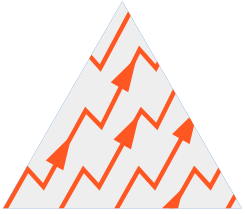


Developer relationships
We analyze non-code features to determine possible relationships between applications and to evaluate whether the developer that created the application may have previously been associated with creation of Potentially Harmful Applications.

Signatures
We use signatures to compare apps against a database of known bad apps and vulnerabilities.



SafetyNet
A privacy preserving sensor network spanning the Android ecosystem, identifying apps and other threats that cause harm to the device.



Heuristic and similarity analysis
We compare applications with each other to find trends that lead to harmful apps.

Static analysis
Static analysis examines the source code without executing it to determine if the app attempts certain types of behavior. For example, static analysis can be used to determine if an app is able to send sensitive data off the device, or to delete SMS messages when they are received—a behavior that could be used to evade automated notifications of account behaviors, such as password changes. This type of analysis is useful for catching specific behaviors that can be identified by reading through the code.

Dynamic analysis
The goal of dynamic analysis is to detect PHA behavior by running the app in a sandboxed virtual environment. As part of the process, a dynamic analysis engine simulates a user clicking through the application and observes what behavior is triggered. Because apps undergoing dynamic analysis actually execute the app, dynamic analysis is able to detect some types of behavior that static analysis may not identify.

In 2016, Google increased the speed and capacity of its analysis system infrastructure. Our analysis engine is now more reliable, three times faster, and exposes more application behavior than ever before. In addition, improvements to the automated event injection engine allow for more app paths to be tested and greater coverage to be achieved, resulting in a three-fold increase in the number of apps with suspicious behavior being flagged for human review.

PHA authors are aware that companies, such as Google, perform dynamic analysis. They attempt to evade identification by not displaying harmful behavior if the app detects it's running in a simulated environment, or virtual machine (VM). Due to the diversity of Android devices, it's difficult for PHA authors to successfully implement any kind of cloaking strategy, but to further prevent successful evasion, we built a system to automatically detect application code that could be used to identify a VM and implemented additional anti-cloaking technologies to evade analysis detection by the apps under analysis. PHA authors will continue to iterate on methods of attempting to evade detection, so this will continue to be an important—and evolving—area.

SafetyNet integration
SafetyNet provides information about device security in the real world. Starting in 2014, we began to use this data to identify potentially harmful behavior that might not occur within our emulated environment. We then used SafetyNet results to identify applications that tried to abuse SMS based on users' responses to warnings about premium SMS. In 2015, we started integrating data from the Anomaly Correlation Engine (ACE) to detect rooting applications and other PHAs. In 2016, we expanded how we use ACE and

added the Dead or Insecure (DOI) scorer to identify applications that appear to make user devices stop working.

Anomaly Correlation Engine (ACE)

> SafetyNet gathers ecosystem security telemetry from over 1.4 billion Android devices to build a picture of the Android ecosystem.

In late 2015, we created the Anomaly Correlation Engine (ACE) to extend SafetyNet's ability to detect and identify PHAs. ACE monitors for changes in key device security indicators, then examines which applications changed since the device was healthy. By gathering this information across a large number of devices, we can predict which application caused the security change and investigate. This allows us to protect our users by identifying, blocking, and removing new PHAs from the Android ecosystem before they can spread widely.

In 2016, ACE focused on identifying apps from outside of Google Play that root devices without user knowledge and consent. We built a machine learning model to separate malicious and user-intended rooting apps and flag the apps that were most likely to attempt rooting without user knowledge and permission. This model has been successful in flagging multi-stage PHAs, which are sometimes not caught by more traditional analysis mechanisms. ACE shows high positive predictive value (precision) in identifying PHAs, raising the signal-to-noise ratio in the review process and thereby allowing human analysts to review apps more efficiently. For example, in a sample from May 2016, ACE flagged malicious rooting apps with over 90% precision.

Analysis of anomalous devices

In April 2016, we began analyzing periodic check-in reports from Verify Apps to identify devices that are Dead or Insecure (DOI). This program observes device behavior after downloading one or more applications to verify that the device continues to check in with Verify Apps. Devices that check in are retained. Devices that stop checking in with Verify Apps after downloading applications are considered DOI. A DOI device generally stops sending SafetyNet data because it has stopped functioning or is experiencing interference from a PHA.

We use statistical analysis to establish a baseline metric of retained devices and look for applications where the retention rate deviates significantly from the norm after install. This deviation is a signal to apply additional scrutiny to these apps. This signal identified members of the HummingBad, Ghost

Push, and Gooligan PHA families and helped protect users from these apps.

We use multiple different scorers to identify potentially harmful applications, but the DOI scorer has been particularly effective at identifying certain families of PHA. This table shows the percentage of apps that the DOI scorer flagged for each family and the percentage of install attempts blocked by the DOI scorer relative to all install attempts for the family. The data spans April 2016 to December 2016.

PHA family	Flagged apps in the family	Blocked install attempts
HummingBad	7%	99.96%
Ghost Push	24%	93%
Gooligan	75%	92%

Based on this success, we're exploring more ways to use device behavior data to identify other anomalous behaviors that may negatively affect users or devices.

Additional machine learning research

In 2016, we used machine learning to track PHAs by observing their install patterns. Traditional machine learning generally focuses on analyzing PHA code and requires a significant sample to compare against. Because the Android ecosystem is disproportionately clean—significantly more apps are non-PHA than PHA—we opted for a semi-supervised approach.

Using SafetyNet data, we trained a neural network to automatically group apps based on install patterns. The network uses document analysis and clustering techniques to group PHAs that push similar payload apps or target similar devices. If an app belongs to a cluster that has a disproportionately high amount of PHA apps it is flagged for review. This allows us to:

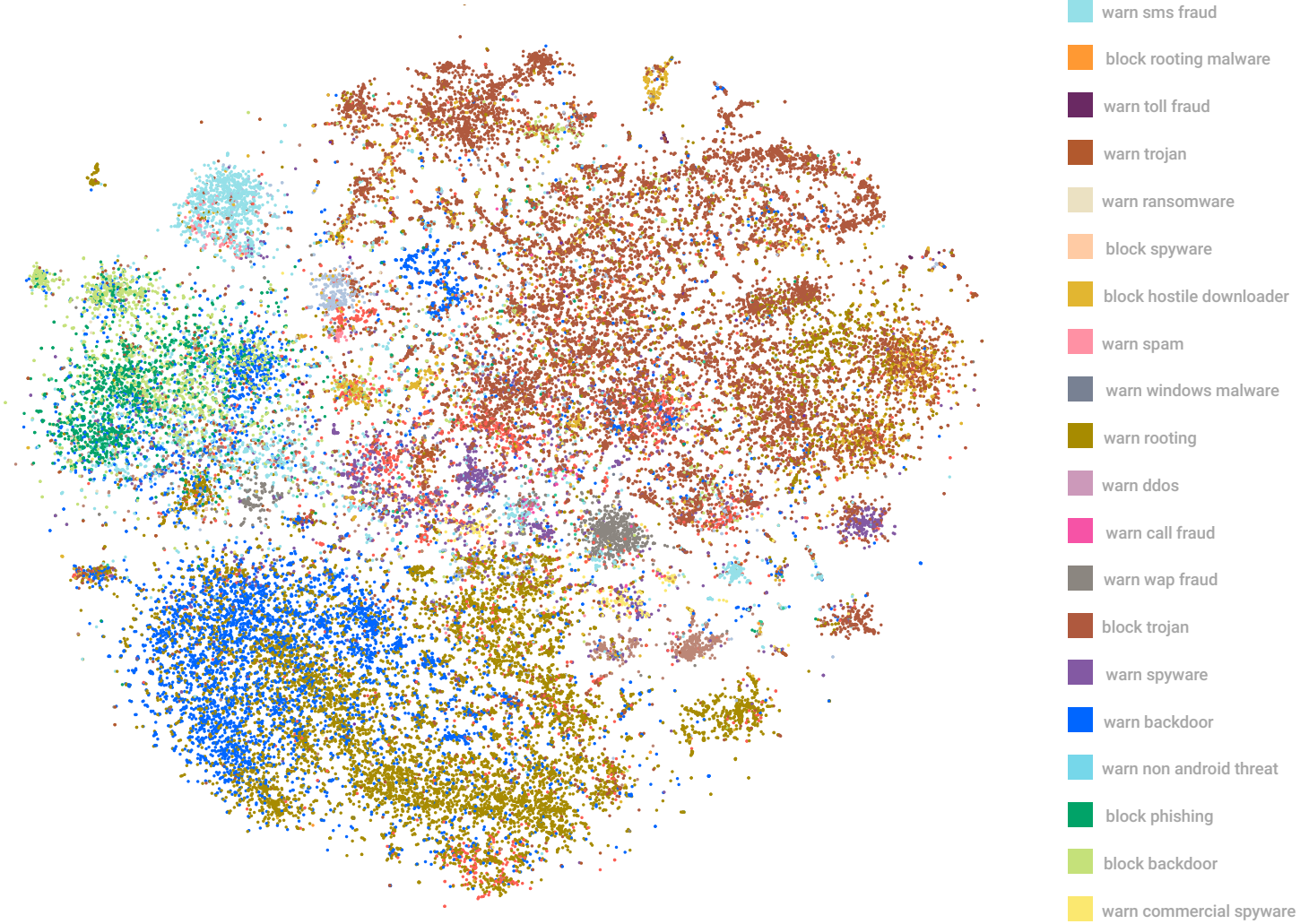
- detect new variants of existing PHA families. Often new variants make slight changes in code but behave similarly.
- detect inconsistencies in previous reviews. If an app marked as SMS fraud is present in a cluster that predominantly contains rooting trojans, we can flag the app for a second review.

— suggest classifications for a previously unseen app based on which clusters are closest to it.

Along with detecting inconsistencies, this research helps Google identify previously unknown PHAs.

This graph shows a visual representation of how PHAs are grouped based on behavior. Closely related apps indicate that a category is well defined and potentially has a few families dominating it. Looser clustering shows that some seemingly unrelated apps have been given the same classification. For example, generic malware tends to be loosely clustered, but the trojan family has tight clusters relating to Ghost Push, a family of Trojans that was one of the most common PHAs in 2015 and early 2016. The relative proximity of apps can help us find new PHAs and improve our scorers and classification systems.

PHA behavioral clusters



Android Platform Security

To be fully effective, security must be part of a product's fundamental design. The Android platform has a number of security features designed into its architecture. The Android platform controls how the operating system works, how applications interact with each other and with the various components of device hardware, such as memory management, camera and microphone, networking, and other system-level features. Android implements a number of protections so that the different system components work with applications in a safe, consistent manner. This table lists some of these protections and how they contribute to the platform-level security.

Platform security feature	Protection
Encryption	Protects data from unauthorized access.
Hardware-backed security	Protects data from unauthorized access.
Kernel self-protections	Protects kernel against memory corruption vulnerabilities and other security flaws in kernel and drivers.
Sandboxing	Keeps each application in a separate space, protecting its data and processing from other applications.
SELinux	Provides an auditable definition of—and enforcement for—security boundaries on all operating system and application components above the kernel.
Userspace hardening	Protects operating system and applications 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.

We upgrade the security of the platform with each major Android release and the monthly security updates.

Updates and features

In 2016, we released Android 7.0 (Nougat). This section summarizes major security features included in the Android platform and highlights where they were updated in Android 7.0. For a list of more updates, see Security Enhancements in Android 7.0.

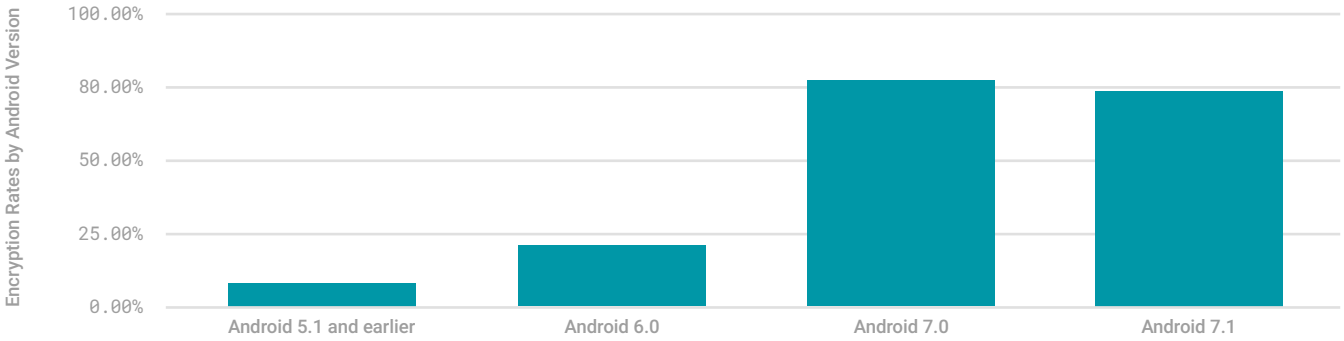
File-based encryption and Direct Boot

Encryption was added to Android in version 3.0, and has continuously evolved since that time. Starting with Android 6.0, all capable[1] devices were required to support encryption and that encryption extended to removable storage, such as SD cards. Many devices, like the Nexus 5X and 6P also use keys that are accessible only with trusted hardware, such as the ARM TrustZone. In 7.0, all new capable Android devices must also have hardware support for key storage and provide brute force protection while verifying the user's lock screen credential before these keys can be used. This way, all data can only be decrypted on that exact device by the user.

1 As defined in the Compatibility Definition Document, for devices that support a lock screen and data storage encryption with Advanced Encryption Standard (AES) crypto performance above 50MiB/sec, the data storage encryption MUST be enabled by default at the time the user has completed the out-of-box setup experience.

This graph compares the rates of devices with encryption enabled by Android version.

Encryption rates, by Android version



Users can check if their device is encrypted under the Security section in Settings. If their device is unencrypted, they can also enable it there.

In previous versions of Android, users with encrypted devices generally needed to enter their PIN, pattern, or password during the boot process to decrypt

their storage area and finish booting. Android 7.0 updated the underlying encryption scheme and streamlined the boot process to speed up rebooting devices. Now many device features, like the phone app and alarm clock, are ready right away. We call this feature Direct Boot.

Under the hood, file-based encryption enables this improved user experience. With this new encryption scheme, the system storage area and each user profile storage area are all encrypted separately. Unlike full-disk encryption, where all data was encrypted as a single block that required user credentials to decrypt, file-based encryption enables the system to reboot normally into a functional state using just device keys. Essential apps can opt-in to run in a limited state after reboot, and when the user unlocks their device, these apps get access to the user's data to provide full functionality.

Verified Boot

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. 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. Depending on device implementation, Verified Boot will either allow the boot to proceed, stop the device from booting so the user can take action on the issue, or prevent the device from booting up until the issue is resolved. As of Android 6.0, per the Compatibility Definition Document, Verified Boot is required for device implementations with Advanced Encryption Standard (AES) crypto performance above 50MiB/second. Starting from Android 7.0, devices with Verified Boot will not boot a corrupt boot image or will boot in a limited capacity with user consent. Android 7.0 also improved dm-verity robustness by introducing forward error correction, which makes the operating system more resistant to data corruption.

Platform hardening

In Android 7.0, we hardened the platform by re-architecting mediaserver, reducing friction around system updates, establishing a core set of trusted certificate authorities, and adding memory protections and reducing the attack surface in the kernel.

Mediaserver improvement

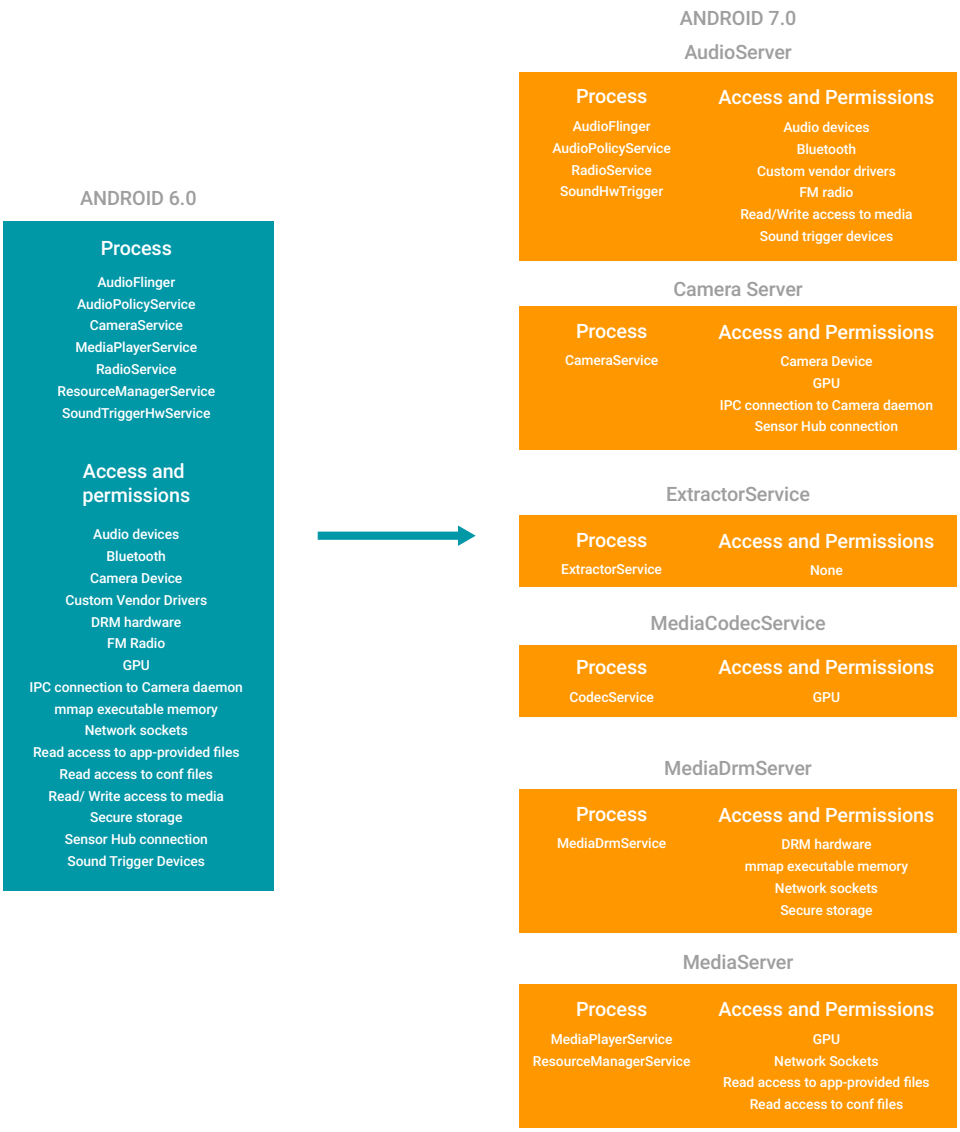
Following the discovery of vulnerabilities in mediaserver's libstagefright library, we added new features that enhanced the existing security model and provided additional defense-in-depth in Android 7.0. These features can prevent the exploitation of unsigned integer overflows and protect the system by de-privileging and isolating components that handle untrusted content. We modified the build process to provide safer default behavior

for integer overflows in security-critical components, such as mediaserver, to prevent exploitation of vulnerabilities that rely on integer overflow.

In addition to the compiler changes, we redesigned mediaserver by moving the logic for parsing file formats into unprivileged sandboxes, and by splitting up components that require sensitive permissions. Now each component is in its own sandbox and has only the minimum permissions necessary. For example, cameraserver only has access to the camera. This way, a potential vulnerability's reach is limited in scope and—depending on the nature of the vulnerability—it may not allow access to anything that would seriously compromise the device's security.

This diagram illustrates the sandboxing and splitting of sensitive permissions.

Mediaserver re-architecture



Improved system updates
In December 2016, Android 7.1.1 improved the system update process by increasing its speed and transparency. Devices using this feature are automatically kept up-to-date with the latest version of the system software. To do this, devices have two system images: one for the currently active system and one to receive an updated image. When an update is available, the device downloads the new system image in the background. The device seamlessly switches to the new software update the next time it reboots. If the system update has trouble, the device can fall back to the previous, working image.

This feature launched late in 2016 and is primarily found on Pixel phones. As more new phones are sold with Android 7.1.1, this feature will become available on a wider variety of devices.

Certificate Authorities
Certificate authorities (CA) are a vital component of the public key infrastructure used in establishing secure communication sessions via TLS. Establishing which CAs should be trusted—and by extension, which digital certificates signed by a given CA should be trusted—is critical for secure communications over a network. In Android 7.0, we changed how Android handles trusted CAs to provide safer defaults.

Beginning with Android 7.0, compatible devices trust only the standardized system CAs maintained in AOSP. By default, applications targeting Android 7.0 and above no longer trust user- or admin-added CAs, which reduces the application attack surface. Apps can choose to bundle additional CAs to trust. Apps can also choose to trust all user- or admin- added CAs. Trust can be specified across the whole app or only for connections to certain domains.

Kernel updates
Android 7.0 introduced several new Linux kernel defenses. These additions are in two main areas: memory protections and attack surface reduction.

Memory protections help prevent attacker-controlled code from being inserted into executable memory. To better protect memory from attacks such as buffer overflows, Android 7.0 incorporates a Linux feature that logically segments kernel memory so that sections with executable code allow read-only and execute access, but are not writable. Android 7.0 also prevents the kernel from directly accessing userspace memory, thus putting the memory upon which the kernel relies further from the control of potential attackers. We also extended stack buffer overflow protections to more array types.

Attack surface reduction aims to reduce the number of potential entry points into the system to a bare minimum while allowing legitimate functionality to operate smoothly. As part of this, we changed the default behavior for perf, a tool used to measure performance in the kernel, to be blocked by default, since this functionality is primarily used during development and is not typically needed by Android users. Developers still have access through developer mode settings. This change reduces risk to users while leaving the functionality available for developers to use. We also implemented ioctl command whitelisting.

First implemented in Android 5.0, seccomp provides an additional sandboxing mechanism that allows a process to restrict the syscalls and syscall arguments available by using a configurable filter. Android 7.0 devices with Linux 3.8 kernels and above must include seccomp support.

Vulnerability rewards and updates

In addition to updating the Android platform with major releases, Google oversees many programs to provide more frequent security updates to Android devices. Over the course of 2016, Google and Android partners made great strides in improving the security of the Android ecosystem and keeping hundreds of millions of device secure.

Android Security Rewards Program

In June 2015, we established the Android Security Rewards Program (ASRP). Since then, the ASRP has awarded over 125 different researchers with nearly 500 rewards. In 2016, we paid almost a million dollars to researchers who found vulnerabilities on the Android platform, Nexus, and Pixel devices.

In June 2016, we increased the payouts and offered 50% more for proof of concepts, which led to higher quality bug reports. We continue to encourage researchers to provide a proof of concept, patch, and CTS tests to maximize the reward amount.

If a researcher chooses to donate their vulnerability discovery reward to charity Google matches their donations. Our researchers have generously donated $14,500, resulting in a total of $29,000 donated to charity from the ASRP program.

Thank you to the many researchers who made Android better in 2016; we look forward to working with new and returning researchers in 2017.

Additional security research programs

In addition to our ongoing Vulnerability Rewards program, we participated in external vulnerability discovery and disclosure competitions, including Mobile Pwn2Own at PacSec, PwnFest, and the Project Zero Prize. Google sent representatives to both Mobile Pwn2Own and PwnFest, where they worked with the contest winners to understand the vulnerabilities used and immediately start work on a fix. This collaboration led to patches checked in within 24 hours of the contest and fixes available to users in under a month.

Mobile Pwn2Own is an annual hacking contest hosted at the PacSec security conference. At the 2016 competition, the Tencent Keen Security Lab Team successfully demonstrated an attack chain that used a hijacked web browser to trigger Google Play's remote app installation feature to install an arbitrary app from Play. The web browser vulnerability was patched and pushed to user devices less than a month after the contest. We also updated the remote app installation feature to prompt the user to enter their password, which makes this form of attack difficult or infeasible in the future.

PwnFest is a similar hacking contest, which is hosted at the POC security conference. Vulnerability researchers from Qihoo 360 used an exploit chain that took advantage of Google Play's remote app installation feature to install a rogue app. The changes to remote app installation mentioned above also work to disrupt an attacker who might attempt to use this method to install a rogue app.

Google's Project Zero announced a contest in September 2016 to find a vulnerability or bug chain that achieves remote code execution on multiple Android devices knowing only the devices' phone number and email address. While the Project Zero Prize is a hacking contest, it complements the efforts of the existing Android Vulnerability Rewards Program.

Zero days

The combination of regular monthly security updates and fast responses by Android device manufacturers significantly mitigated the impact of zero day vulnerabilities against the Android platform. For example, CVE-2016-5195 (also known as Dirty Cow) was publicly disclosed on October 19, 2016. As the

patch was available from upstream Linux, some device manufacturers, such as BlackBerry, deployed a fix in time for the November 2016 security update. We created a special patch string (November 06, 2016) for devices to indicate the vulnerability had been fixed. A fix was required for the December 01, 2016 security patch level.

Android Security update improvements

> In 2016, Google collaborated with SoC vendors, device manufacturers, and mobile network operators to improve the Android security update processes.

We worked closely with SoC vendors, such as Qualcomm, Broadcom, MediaTek, and Nvidia, to address security vulnerabilities in their components and streamline the delivery of fixes to downstream device manufacturers. In addition to issues that were disclosed throughout 2016, Android security bulletins addressed 86 Qualcomm security vulnerabilities that were shared with Google prior to 2016.

We also helped device manufacturers build and expand processes to deliver monthly security updates to customers. Over the course of the year, Android device manufacturers became more efficient at delivering monthly security updates, including expanding their security programs to accept and address security vulnerabilities specific to their devices.

> We also teamed up with mobile network operators in many countries to reduce friction and increase the speed at which customers receive monthly security updates on their devices.

For example, we helped drive an expedited approval and sign-off process for monthly security updates that has reduced approval times from over one month to less than one week. Android partners also made significant investments in discovering security vulnerabilities in their products and

making that information public. In 2016, Qualcomm launched a paid vulnerability rewards program that offers rewards of up to $15,000 to encourage security researchers to responsibly disclose vulnerabilities in Qualcomm products. Samsung and LG also launched security-focused websites discussing their security programs and providing details of fixes to security vulnerabilities specific to their devices.

Android security patch level

In 2015, we introduced the Android security patch level. This patch level allows users and enterprise customers to verify their Android device contains the most recent security updates. Our monthly public security bulletins document newly patched security vulnerabilities and the security patch level that contains all of these fixes. By checking the security patch level, users can verify their device has the fixes for the issues described in our bulletins. In 2016, we updated the monthly public security bulletins to include two security patch levels in order to provide Android partners with the flexibility to fix a subset of vulnerabilities that are similar across all Android devices more quickly.

Security updates program

All of the vulnerabilities found by the ASRP and additional engagements make their way into the monthly Android security bulletins and security updates. In 2016, these addressed 655 vulnerabilities—broken down into 133 Critical, 365 High, 154 Moderate, and 3 Low severity fixes. This represented a greater than 275% increase from 2015, attributable in large part to efforts such as the ASRP.

These regular monthly security releases provide security patches to manufacturers so they can update their devices. In 2016, manufacturers demonstrated a significant and increased commitment to regularly update their devices.

Here are some of the Android devices that attained an update rate of 60% to 95% by the end of 2016:[2] Google Pixel, Google Pixel XL, Motorola Moto Z Droid, Oppo A33W, Nexus 6P, Nexus 5X, Nexus 6, OnePlus OnePlus3, Samsung Galaxy S7, Asus Zenfone 3, bq Aquarius M5, Nexus 5, Vivo V3Max, LGE V20, Sony Xperia X Compact, BlackBerry PRIV.

Much like the device manufacturers, we also saw increased commitment and effort from mobile network operators to promptly deliver security updates to user devices. In the United States, over 78% of active flagship Android devices[3] on the four major mobile network operators reported a security patch level from the last three months.

2 Percentage is calculated as percentage of Android devices running 4.4.4 or higher that checked in with Google Play Services between 12/3/2016 and 12/31/2016 and reported a security patch level of October 1, 2016 or greater.

3 Galaxy S7, Galaxy S7 Edge, Galaxy S7 Active, Galaxy S6, Galaxy S6 Edge, Galaxy S6 Edge+, Galaxy S6 Active, Galaxy Note5, Galaxy Note4, Galaxy Note Edge, Galaxy A5(2016), LG G5, LG G4, LG G3, V10, Moto X Play, Moto X Style, Moto X Force, DROID MAXX 2, DROID Turbo 2, Mate 8, Mate S, P8, P9, Xperia Z4, Xperia Z5, Xperia Z5 Compact, Xperia Z5 Premium

In Europe over 73% of active flagship Android devices on the major mobile network operators reported a security patch level from the last three months.

Over the course of 2016, Google continued providing security patches for Android 4.4 and higher. The percentage of Android devices running Android 4.4 or higher increased from 70.8% of active devices at the beginning of 2016 to 86.3% of active devices at the end of 2016. As of December 2016, 735 million Android devices report a 2016 security patch level.

These 735 million devices are spread across over 200 device manufacturers and represent over 2,000 Android models and over 3,400 SKUs and represent a step forward for the Android ecosystem to help keep users safe and secure.

App Security Improvements program

In addition to working with device manufacturers and keeping the platform up to date, Google also works with application developers to improve the security of their apps. The App Security Improvement (ASI) program identifies apps in Google Play that have security vulnerabilities in their own code or in third-party libraries they include. To do this, we scan apps uploaded to Google Play for known vulnerabilities. As vulnerable apps are identified, the ASI program contacts developers by email and the Play Developer Console with guidance to fix the vulnerabilities.

In 2016, the ASI program added 18 new security vulnerabilities, up from 8 in 2015.

The program notifies developers of 26 vulnerabilities overall and remediated vulnerabilities in over 275,000 apps in Google Play.

Four of these campaigns are based on advanced static analysis algorithms. Google continues invest and make advances in state-of-the-art in program analysis to protect users against vulnerabilities. To the best of our knowledge, Google Play is the first app store to employ such technologies for screening apps for vulnerabilities.

This table lists the campaigns initiated in 2016:

Campaign	Started
AdMarvel	Feb 8, 2016
Libupup (CVE-2015-8540)	Feb 8, 2016
TrustManager	Feb 17, 2016
Airpush Ad SDK	Mar 31, 2016
MoPub Ad SDK	Mar 31, 2016
OpenSSL ("logjam" and CVE-2015-3194, CVE-2014-0224)	Mar 31, 2016
Libpng	Jun 16, 2016
Libjpeg-turbo	Jun 16, 2016
Vpon Ad SDK	Jun 16, 2016
Supersonic Ad SDK	Sep 28, 2016
Fragment Injection	Nov 29, 2016
Insecure Hostname Verification	Nov 29, 2016

We also launched six campaigns that warn developers about a potential security issue, but have no remediation deadline at this time. These campaigns ask developers to investigate and resolve a potential security risk that may not be an immediate risk to their users.

This table lists these warn-only campaigns:

Campaign	Started
Developer URL Leaked Credentials	Jun 16, 2016
Embedded Google Refresh Token OAuth	Jul 28, 2016
In-app billing interception	Jul 28, 2016
Embedded Facebook OAuth Token	Nov 28, 2016
Embedded Foursquare OAuth Token	Nov 28, 2016

To encourage prompt security fixes, we began imposing remediation deadlines in 2015. 90 days after the first notification, app updates and new apps containing the vulnerability are not accepted in Google Play. Any app that was already in Play and exceeds the 90-day remediation period without a fix continues to be available on Google Play. However, if the developer wants to upload a new version after the remediation period, the new version must include fixes for the disclosed vulnerabilities.

> In order to better support developers, in December, 2016 we launched the App Security Improvement web site.

This web site gives an overview of the program, provides a list of all active campaigns, and links to help for each campaign with details about each vulnerability and how to remediate it.

Public and developer outreach

In 2016, we tried to regularly communicate how we improved Android by blogging about updates and giving presentations to the security community.

Blog posts

We published a variety of security-related blog posts on many topics in 2016. Here are some samples:

- Keeping Android safe: Security enhancements in Nougat
- Changes to Trusted Certificate Authorities in Android Nougat
- Protecting Android with more Linux kernel defenses
- Strictly Enforced Verified Boot with Error Correction
- Hardening the media stack
- One Year of Android Security Rewards
- Enhancing App Security on Google Play
- Protecting against unintentional regressions to cleartext traffic in your Android apps
- More Safe Browsing help for webmasters
- Inline Encryption: Better, Faster, Stronger

Conference presentations

In addition to blog posts, many Android engineers presented at conferences. Here are some samples:

- Linux Security Summit: Android: Protecting the Kernel
- Blackhat: The Art of Defense: How Vulnerabilities Shape Security Features and Mitigations in Android
- Virus Bulletin: Android Security Small Talk
- Qualcomm Security Summit: Overcoming Stagefright—Integer Overflow Protections in Android
- Qualcomm Security Summit: Rooting for fun and profit
- Kapersky Security Analysts Summit: Protecting Android users against harmful apps
- Samsung Dev Conference: Developing secure Android for Work apps
- RSA conference: Building an Android Scale Incident Response Process
- Botconf: Hunting Droids From The Inside
- Wired: Some Thoughts on “Emerging Threats”
- Trustech 2016: Making the Android ecosystem safer
- Google I/O: What’s New in Android security

Ecosystem Data

This section provides data on the overall state of the Android ecosystem in 2016 with details and trends for Potentially Harmful Applications categories, both inside and outside of Google Play. SafetyNet gathers ecosystem security telemetry, which allows us to get an overview of the utilization of security-related services and track PHAs in the ecosystem. Verify Apps scans user-installed applications at install time for PHAs, regardless of the apps' origin.

Potentially harmful applications

Potentially harmful apps (PHAs) are applications that could put users, user data, or devices at risk. Commonly discussed categories of PHAs include trojans, spyware, or phishing apps.

Applications that weaken Android's built-in security features are potentially harmful but can also provide functionality that users (typically power users) find useful and desirable. We still warn users when they try to install these types of apps, but represent these types of apps differently in our statistics than classic "malware" PHAs. For example, we warn users about applications that disable Android security features like SELinux or root the device with disclosure and user consent. Power users can proceed with installation while users who were not aware of the dangers can be more informed about the decision to alter their system. 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.

We are also less strict in our definition of certain PHAs than some users expect.A classic example is advertising spam, which we define as an app that pushes advertising to the user in an unexpected way, such as on the device home screen or lock screen. While advertising spam is annoying and negatively impacts a user's Android experience, this kind of behavior is not classified

as a PHA by the Android Security Team as it doesn't put Android users, user data, or devices at risk.

In 2016, we changed some of our PHA definitions. We shifted classifications within the billing fraud category to better track abuse trends. Of particular note, we renamed WAP fraud to toll fraud and widened its definition.

This table contains the PHA category definitions that we use to warn users when they attempt to install a PHA. Darker rows signify a change to the definition in 2016.

User-disclosed rooting

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 malware categories if executed automatically. In general, the backdoor is more a description of how 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.
Data collection	*Reclassified as Mobile Unwanted Software (MUwS).* Any application that collections at least one of the following without user consent: — Information about installed applications — Information about third-party accounts — Names of files on the device This includes collecting the actual list of installed applications as well as partial information like information about currently active apps.
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.

PHA	Definition
Hostile downloader	An application that is not in itself potentially harmful, but downloads other potentially harmful apps. For example, a gaming app that does not contain malicious code, but persistently displays a misleading "Security Update" link that installs harmful apps.
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. Toll Fraud 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 administrator 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 administrator 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 applications. 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.

Some users choose to root their phones in order to get access to 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 track this because rooting a phone does remove some security protections and we want to monitor how much of the ecosystem is in this intentionally weakened state.

As previously discussed, Google's SafetyNet security service provides a feature called Attestation that can check for signs of rooting. This API is invoked over 200 million times per day and gives us an approximation for the number of devices that are rooted. Worldwide 94.4% of all Android devices report passing the basic system integrity check, from which we conclude that these devices are not rooted. The remainder includes devices that were rooted by the user, sold as a rooted device, were unintentionally rooted by a PHA, or that do not match expected characteristics of an intact security model.

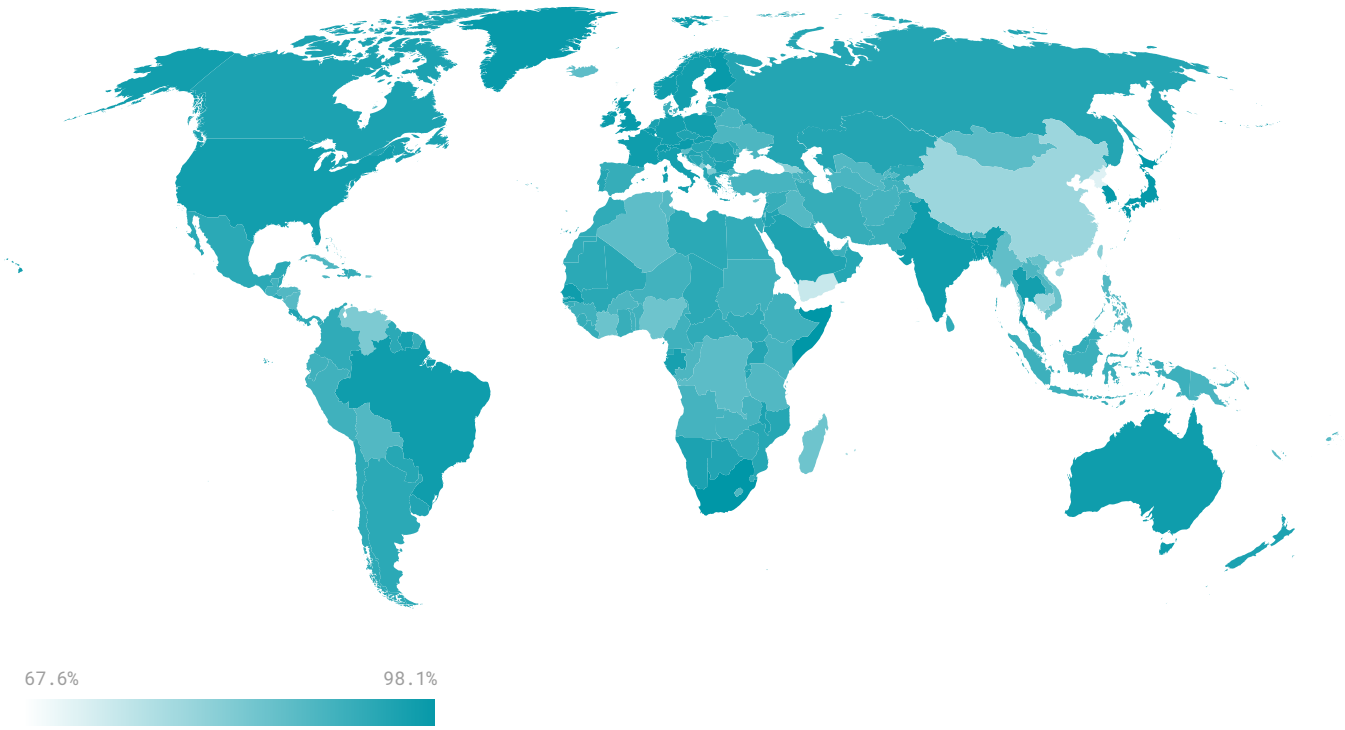
Verify Apps tracks the ratio of all app installs to user-intended rooting. In 2016, user-intended rooting installs comprise 0.3461% of all installs, with fewer than 0.0001% of installs coming from Google Play.

Apps that root devices without disclosure to and permission from the user are significantly more rare.

In 2016, malicious rooting apps accounted for 0.00233% of all installs.

Most devices are either rooted by the user or the manufacturer. This map shows the distribution by country of devices that pass the basic system integrity check. The darker the color, the higher the percentage of devices that pass the basic system integrity check.

Devices that pass system integrity check, by country



Mobile Unwanted Software (MUwS)

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

This type of app has long been prohibited by Google Play's policies—but even outside of Google Play it is 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. Previously, we categorized some of these apps as PHAs, but to improve the clarity of our classifications we're now classifying them as MUwS. As part of this change we've moved the apps formerly described as "Data Collection" into the MUwS category. In 2016, we defined MUwS as apps that collect at least one of the following without user consent:

- Information about installed applications
- Information about third-party accounts
- Names of files on the device

This includes collecting the actual list of installed applications, as well as partial information like information about currently active apps. We expect the number of behaviors classified as MUwS will increase in 2017, to align with our cross-platform approach.

Google identifies MUwS applications and works with developers to remove overly aggressive data collection behavior from their apps or disclose the data collection to users. In 2016, this resulted in the removal of data-collection code from applications used by tens of millions of users.

Device and Ecosystem Hygiene

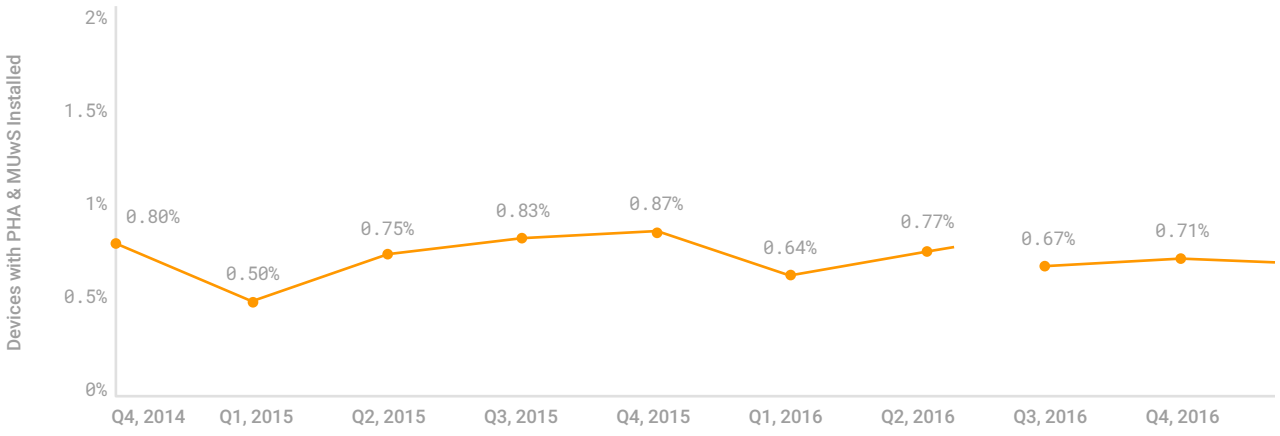
The broadest statistic we use to measure device hygiene is the frequency with which PHAs are detected during a routine full-device scan. This graph shows the level of device hygiene for the entire Android ecosystem in 2016.

> Since we began to measure device hygiene in late 2014, we have seen on average less than 1% of devices have PHAs installed. This continued in 2016.

Throughout 2016, PHA rates were lower than in the second half of 2015.

Devices with PHA & MUwS installed[4] (except user-initiated rooting)

4 The break in trend lines at the start of Q3 2016 is due to a change in how Verify Apps counts unique devices.



Ecosystem data: Inside and outside of Google Play

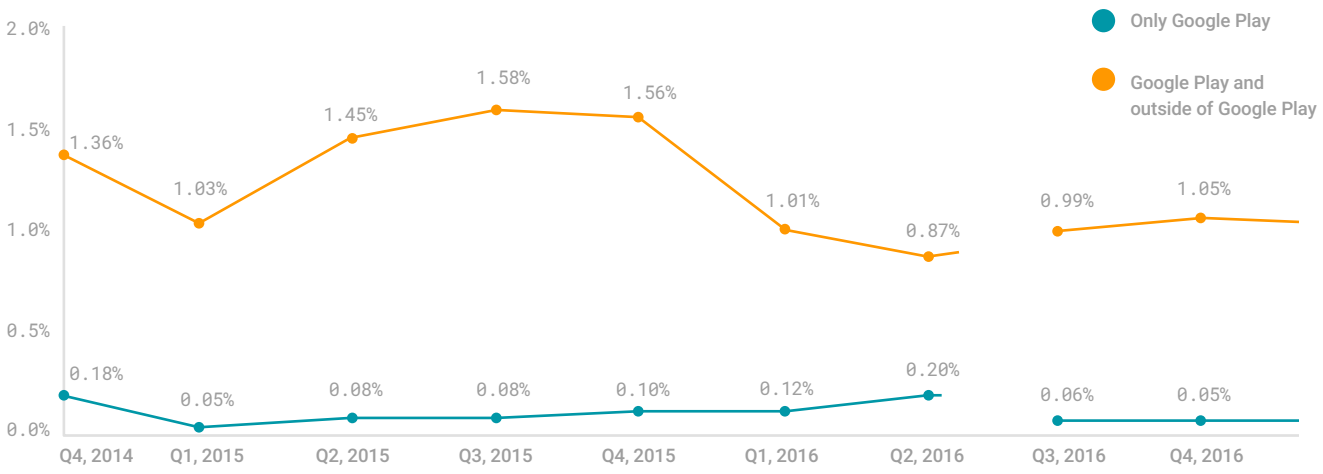
Apps available on Google Play must adhere to a set of published policies and are reviewed to verify their compliance. While Google Play has one of the most effective systems for catching PHAs and is constantly evolving to address new threats, no review process is perfect. With just over one million applications in Google Play, a small number of PHAs still manage to creep in. But this number is small, as PHAs accounted for 0.16% of all apps published to Google Play in 2016.

By contrast, a user was ten times more likely to download a PHA from outside of Google Play in 2016.

This graph shows the percentage of devices that installed a PHA. The blue line represents devices that only download from Google Play and the yellow line represents devices that install applications from outside of Google Play, whether exclusively or in addition to Play apps. Both percentages are given relative to the total number of Android devices.

Devices with PHA installed (except user-intended rooting), inside and outside of Play [5]

5 The break in trend lines at the start of Q3 2016 is due to a change in how Verify Apps counts unique devices.



PHA distribution analysis

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

Many devices install applications from both Google Play and outside of Google Play. For devices that allow apps from other markets, the device hygiene metric is a blended average of all distribution paths. Device hygiene varies considerably across the ecosystem based on the number of applications users install—which ranges from 0 to several hundred, with a mean of 21 user-installed applications and a mode of 9 applications per device. To provide more insight into root cause of changes in device hygiene, we also analyze the individual install events and distribution paths.

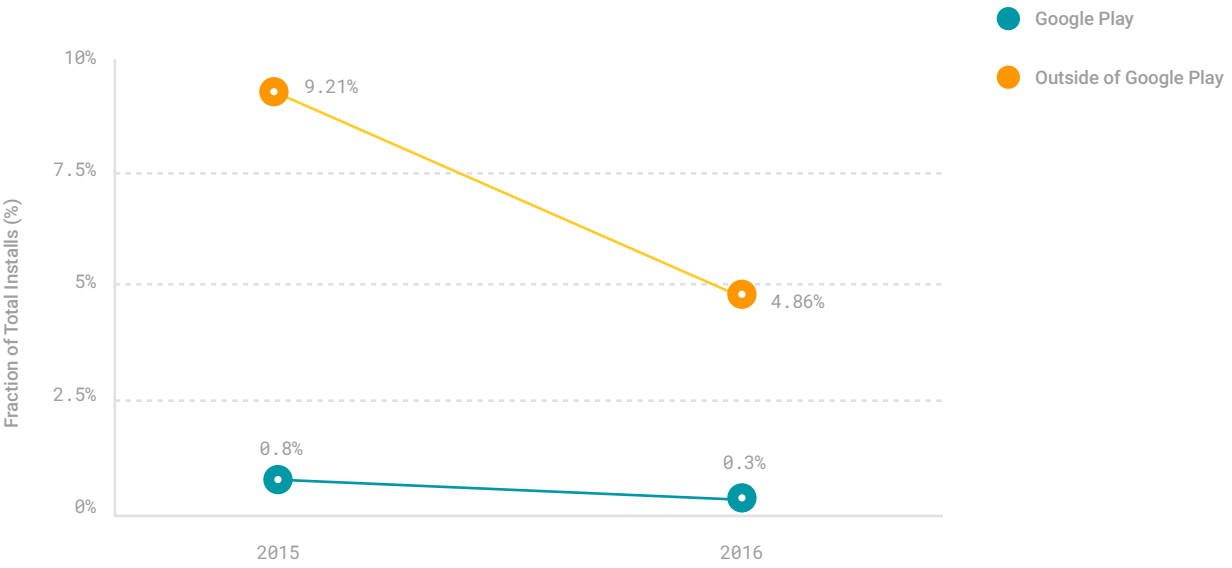
We also track when a user is not warned at install time that an application is a PHA. If a warning is not presented on an app that is later determined to be a PHA, we call this a false negative. Finding false negatives requires re-evaluating apps based on new information gained from monitoring application behavior. The false negative rate changes as our understanding of current PHAs evolves. By definition, 0% of false negatives are known on the day of installation, and that number increases over time. In 2016, only .02% of installs downloaded from Google Play were discovered to be false negatives within 90 days and the percentage of false negatives remains consistent beyond 180 days. For installs outside of Google Play, the numbers are higher. The average false negative rate is 2.6% within the first 90 days and it continues to increase to 4.9% (or +2.3%) by end of the 180-day window.

This section compares PHA installations in 2016 against previous years.

Overall, the rate of PHA installs—inside and outside of Google Play—dropped in 2016 compared to 2015.

Overall, the rate of PHA installs—inside and outside of Google Play—dropped in 2016 compared to 2015, primarily as a result of improved detection of large families such as Ghost Push. For more information about Ghost Push, please refer to our 2015 Android Security Year in Review.

PHA and MUwS install rates

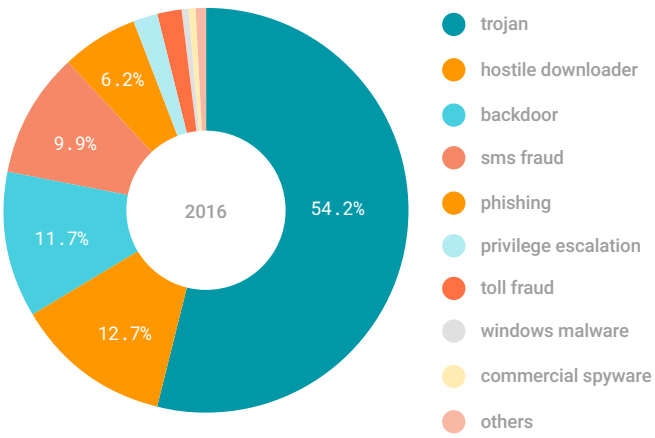


Trojans comprise the majority of installs both on and off Play, but the percentage of installs varies by market. Overall, the number of trojan PHA installs dropped in 2016, largely due to a reduction in Ghost Push installs.

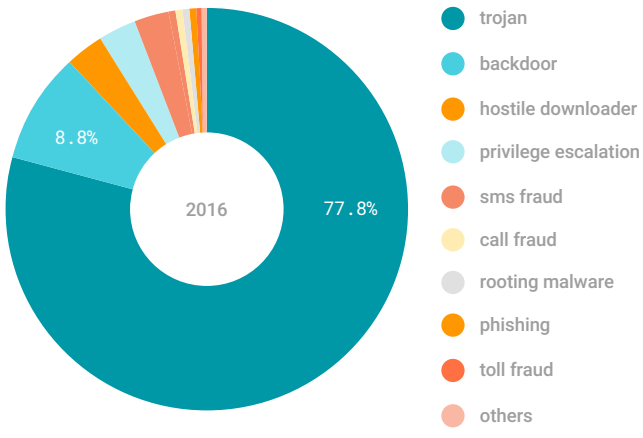
Top PHA categories

These charts show all PHA categories broken down by percentage against other PHAs in 2016.

GOOGLE PLAY



OUTSIDE OF GOOGLE PLAY



PHA install rate

These tables show changes in PHA category between 2015 and 2016. Column one lists the PHA category and column two shows the relative size of installs compared to other PHA categories for 2016. Column three compares the change between 2015 and 2016 for PHA installs against the total install base. Column four shows the percentage of PHA installs for that family out of all application installs in 2016 and column five shows the changes to that percentage since 2015. For example, in 2015 trojan installs represented .03348% of all installs and this year it represents 0.01623%.

Google Play

PHA category	2016 share in PHA Category	2015-2016 change in PHA installs	2016 percent of total installs	2015-2016 percentage point change of total installs
trojan	54.2%	-51.5%	0.01623%	-0.01725
hostile downloaders	12.7%	-54.6%	0.00380%	-0.00458
backdoor	11.7%	-30.5%	0.00351%	-0.00154
sms fraud	9.9%	282.2%	0.00296%	+0.00219
phishing	6.2%	-73.0%	0.00185%	-0.00501
privilege escalations	2.5%	-77.6%	0.00076%	-0.00263
toll fraud	2.0%	592.8%	0.00060%	0.00051
commercial spyware	0.4%	-45.3%	0.00012%	+0.00004
call fraud	0.3%	-50.4%	0.00008%	-0.00008
ransomware	0.002%	-92.9%	0.000001%	-0.000009

Outside of Google Play

PHA category	2016 share in PHA Category	2015-2016 change in PHA installs	2016 percent of total installs	2015-2016 percentage point change of total installs
trojan	77.8%	31.3%	2.58579%	-0.23835
backdoor	8.8%	229.8%	0.29347%	+0.16592
hostile downloaders	3.9%	-94.7%	0.12925%	-3.34500
privilege escalations	3.0%	66.4%	0.09873%	+0.01365
sms fraud	2.9%	108.6%	0.09730%	+0.03043
spyware	1.8%	272.7%	0.06078%	+0.03740
call fraud	0.5%	-61.8%	0.01536%	-0.04227
rooting malware	0.4%	-43.5%	0.01282%	-0.01969
phishing	0.4%	-46.2%	0.01262%	-0.02098
toll fraud	0.3%	-79.7%	0.00893%	-0.05418

The discussions below address trends inside and outside of Google Play and by geographic regions, as we have seen variances across these vectors.

Google Play: PHA trends

The overall health of Google Play has increased year over year. The number of installed trojans dropped by 51.5%, hostile downloaders dropped by 54.6%, backdoors dropped by 30.5%, and phishing apps dropped by 73.0%.

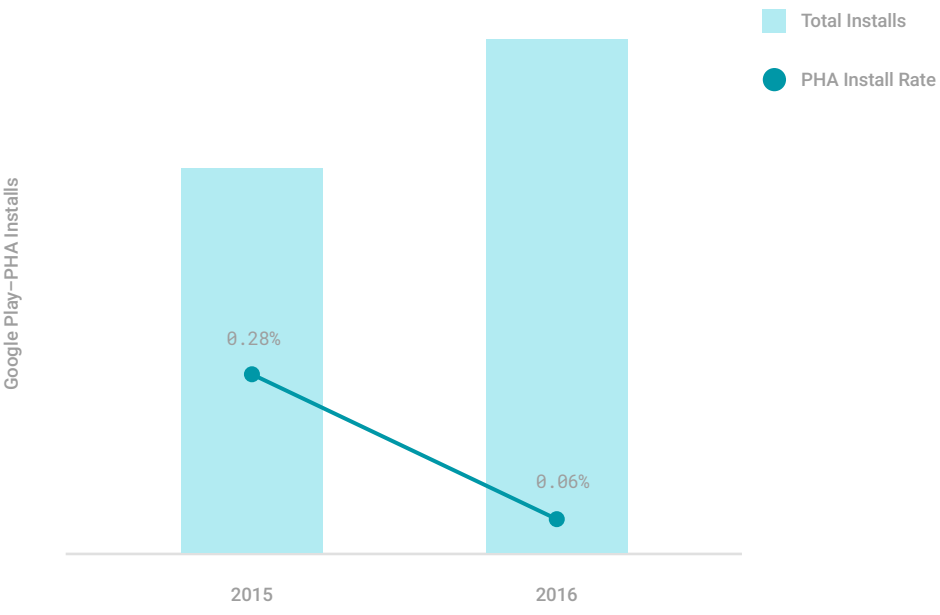
The drops in trojans, hostile downloads, and backdoors can be attributed to Google's work to reduce the number of installs of the Ghost Push family, while the drop in phishing apps came from research into phishing apps that target popular social networks. Two categories—SMS fraud and Toll fraud—did see increased install rates. We discuss these categories in more detail later in this report. All other PHA categories combined represent less than 0.01% of installs from Google Play and saw minor decreases in 2016.

While most PHA categories dropped, there has been a rise in legitimate developers collecting more information about users and devices for analytics and advertising purposes. In particular, we have seen increased installation of apps that include of third-party SDKs that collect user identifying information, such as social network account names or phone numbers. This trend is one of the reasons that we introduced the MUwS classification and are working to enforce new policies for MUwS in 2017.

As the installation base of legitimate apps that violate data-collection guidelines is much larger than the installation base of PHAs, our work with developers has led to the most significant month-over-month improvements to the health of Google Play.

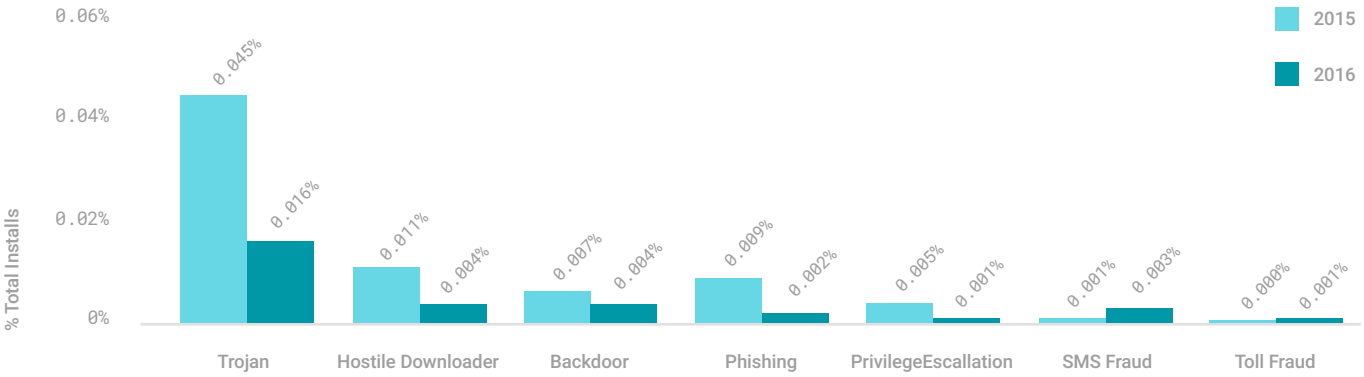
This chart illustrates the year-over-year trends for PHA install rates relative to total installs for 2015 to 2016. The decrease in PHA install rate is primarily due to campaigns to clean up apps in the Ghost Push family.

Google Play—PHA installs



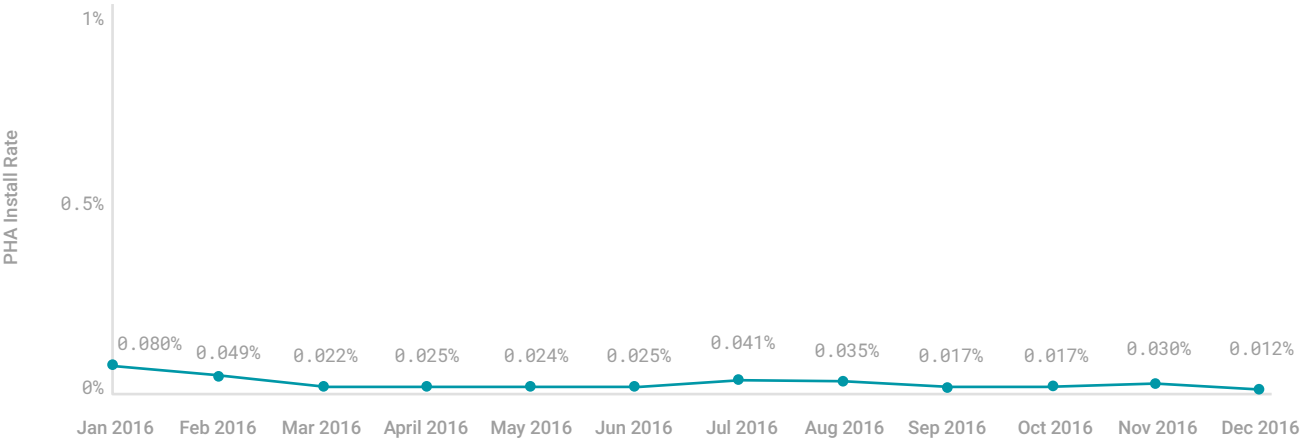
This chart illustrates the year-over-year trends for PHA install rates by PHA category for 2015 to 2016.

Google Play—PHA installs, by category



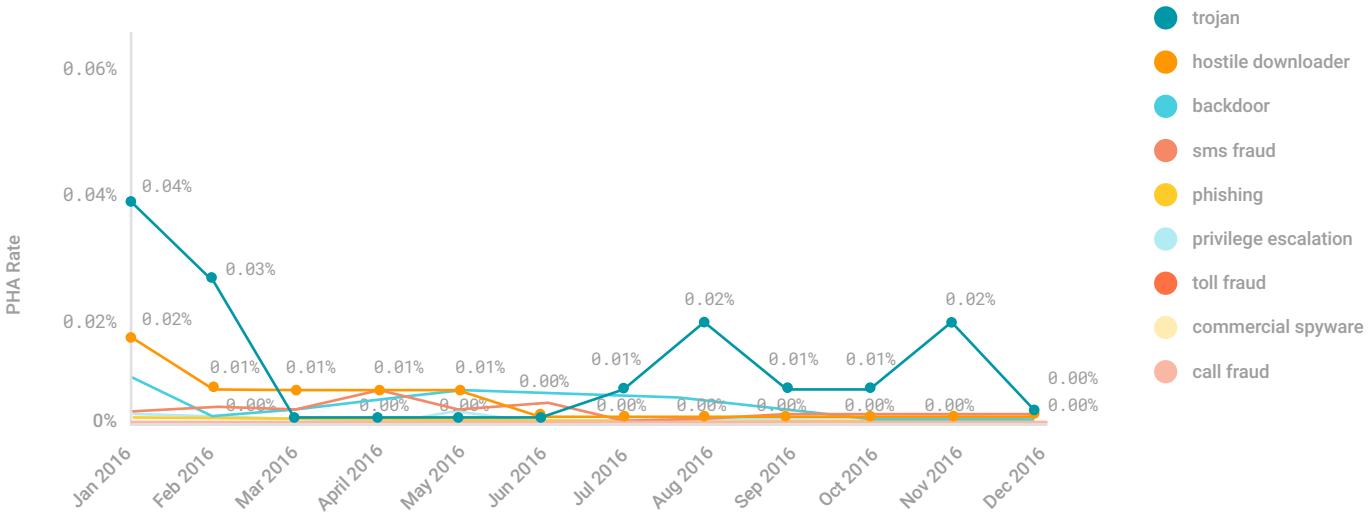
In 2016, there was a slight decrease in PHA installs from Google Play in Q1, with the install rates remaining largely constant after that.

Google Play—PHA installs



This chart further breaks down PHA installs in 2016 by specific PHA categories.

Google Play—Top PHA categories



Google Play: Trojans

In 2016, trojan installations decreased from 0.05% of all installations from Google Play in 2015 to 0.02% of all installations from Google Play. We attribute this decrease to our focus on trojans after the Ghost Push family became one of our focus areas throughout 2016.

Two countries stood out to us as being most improved in terms of decreased trojan installation rates. Thailand saw a year-over-year decline in Trojan installations from Google Play of -76.9%. Russia came right after in second place with a year-over-year decline of -71.6%.

Despite these successes, trojans remained a focus for Google Play in 2016. There are three distinct peaks in the data for trojans. The peak from January to February corresponds to install trends for Ghost Push.

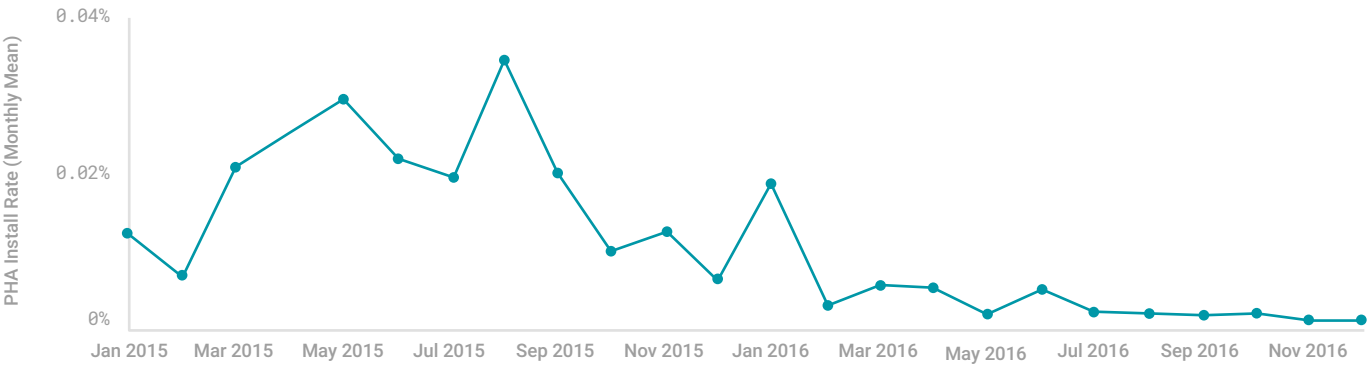
The peak from July to August is attributed to a click fraud PHA family called Chamois. Chamois committed ad fraud by tricking users into clicking pop-up ads using deceptive UI overlays. It installed unwanted apps in the background to hide its activities from the user and performed mobile billing fraud by sending premium text messages. After discovering and analyzing this PHA, we implemented rules to detect the ad fraud, as well as rules to find and remove it using SafetyNet and Verify Apps.

The November peak was caused by a variant on the Ghost Push family.

Google Play: Phishing

Another PHA category with a strong decline in prevalence on Google Play is phishing. In 2015, phishing apps accounted for 0.009% of Google Play installs in 2015. In 2016, this number dropped to 0.002%.

Google Play—Phishing trends



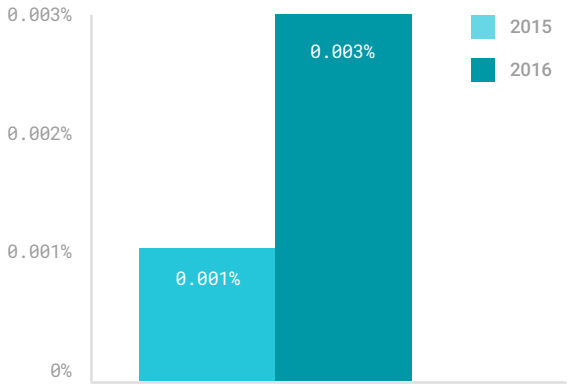
The reason for the decline can be attributed to apps that were unofficial extensions to social media apps such as VKontakte and Instagram. In 2015, VKontakte—Russia's most popular social network—removed the ability to play music from their mobile app. Multiple third-party apps that allowed users to play music from VKontakte on mobile devices were developed to fill this gap. Credentials that users entered were sent to third-party servers not associated with VKontakte, putting the VKontakte accounts of Android users at risk. We classified these applications as phishing and removed them from Google Play to prevent potential abuse. We are not aware of the collected credentials being used maliciously.

Google Play: SMS fraud

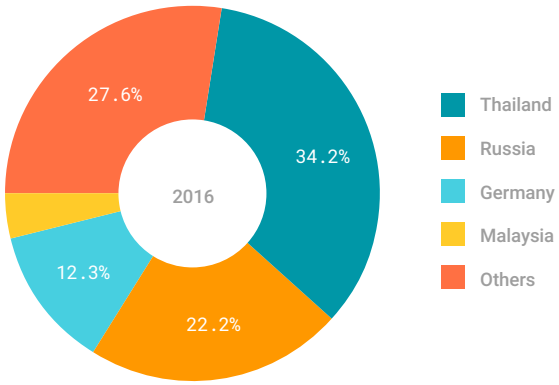
In contrast to the decreases in most PHA categories, SMS fraud increased from 0.001% of installs in 2015 to 0.003% of installs in 2016. This section examines trends around SMS fraud apps from Google Play.

Thailand was most impacted with 34.2% of total SMS fraud installs. Russia (22.2%), Germany (12.3%), and Malaysia (3.8%) were the countries with the next highest numbers of SMS fraud app installations.

Google Play—SMS fraud



Google Play—SMS fraud, by country



SMS fraud downloads in Thailand came from applications where users could purchase wallpapers. Users who agreed were then charged a daily fee. While SMS payments are a legitimate way to monetize Android applications, the disclosure code of the premium rate subscriptions in these cases did not meet Google Play policies and we had reason to believe that users did not understand they would be signed up for a premium payment subscription.

Russia and Germany were both targeted by an SMS fraud family called WallySMS which we suspect is of Russian origin. These applications are advertised as games or system tools, like launchers or home screen improvements but actually defraud users of money through premium SMS.

Malaysia was not specifically targeted by any particular PHA family. Rather, a couple of disconnected applications contributed to the SMS fraud downloads number. The most-downloaded applications were system tools—like disk or memory optimizers—that subscribed users to premium SMS subscriptions without their consent.

Google Play: Toll fraud

Toll fraud applications charge users by other means than premium SMS or premium calls, such as using the WAP protocol to make online payments that are billed through the phone bill. We saw a year-over-year increase in toll fraud PHAs, from 0.0001% of all installs in 2015 to 0.0006% of all installs in 2016. This was very similar to the increases we saw for SMS Fraud. In 2016, the most common countries for toll fraud were Thailand, Germany, and Russia.

Google Play: Country trends

In 2016, the top 50 countries accounted for nearly 50% of app installs from Google Play. 48 of these 50 countries saw a reduction in PHA installation rates

compared to 2015. Only Austria and Venezuela stayed flat compared to the previous year. None of the top 50 countries saw an increase in PHA installation rates in 2016.

This table presents data for the top 20 countries as determined by overall app installs. The total install magnitude is determined by all application installs, whether they are PHAs or not. The table lists the percentage of devices for that locale that have a PHA installed, and compares the change in install rate to the rate for the previous year.

Country	PHA Install Rate		Change
	2015	2016	
Argentina	0.08%	0.03%	-0.05%
Brazil	0.10%	0.03%	-0.07%
Canada	0.06%	0.01%	-0.05%
France	0.05%	0.02%	-0.03%
Germany	0.05%	0.03%	-0.02%
Great Britain	0.04%	0.02%	-0.02%
India	0.10%	0.05%	-0.05%
Indonesia	0.09%	0.05%	-0.04%
Italy	0.06%	0.02%	-0.04%
Japan	0.02%	0.00%	-0.02%
Korea	0.04%	0.01%	-0.03%
Mexico	0.08%	0.03%	-0.05%
Russia	0.20%	0.07%	-0.13%

Country	PHA Install Rate		Change
	2015	2016	
Saudi Arabia	0.10%	0.03%	-0.07%
Spain	0.06%	0.02%	-0.04%
Taiwan	0.14%	0.02%	-0.12%
Thailand	0.17%	0.09%	-0.08%
Turkey	0.08%	0.05%	-0.03%
United States	0.03%	0.01%	-0.02%
Vietnam	0.24%	0.05%	-0.19%

Google Play: Top PHA decreases by country

In this section we examine Russia, Vietnam, and Taiwan: three countries that saw significant year-over-year declines in the observed PHA rate.

Russia

PHA installation rate in Russia dropped by 0.14% to 0.07% of all Play installs in Russia. The strong decline in PHA installations in Russia can be explained by an 81.5% drop in phishing applications, as mentioned in the discussion on phishing above.

Vietnam

In 2016, 0.05% of all Play installs in Vietnam were from PHAs. This is a 0.19% improvement over the 0.24% install rate in 2015.

The decline in Vietnam's PHA rate is due to cleaning up a number of independent applications and PHA families that were getting some traction toward the end of 2015. For example, one PHA that masqueraded as a game accounted for 20% of PHA downloads from Play for Vietnam. Additionally, a smaller network of gambling applications that spammed contacts of gamblers on a popular social media platform was removed from Google Play towards the end of 2015. Throughout 2016, all PHA families targeting Vietnam were detected quickly, which contributed to the increased safety of Google Play in Vietnam.

Taiwan

PHA install rate in Taiwan declined over the past year from 0.14% in 2015 to 0.02% in 2016. Major reductions in Backdoor (-98.4% YoY) and Phishing (-97.4% YoY) categories were the main contributing factors to the overall improvements in Taiwan. Within Taiwan, 74% of the top 100 PHA apps in 2015 belonged to two large backdoor families. These backdoor apps were strongly enforced throughout the year and the backdoor category was reduced down to 10% of the top 100 PHA apps in 2016.

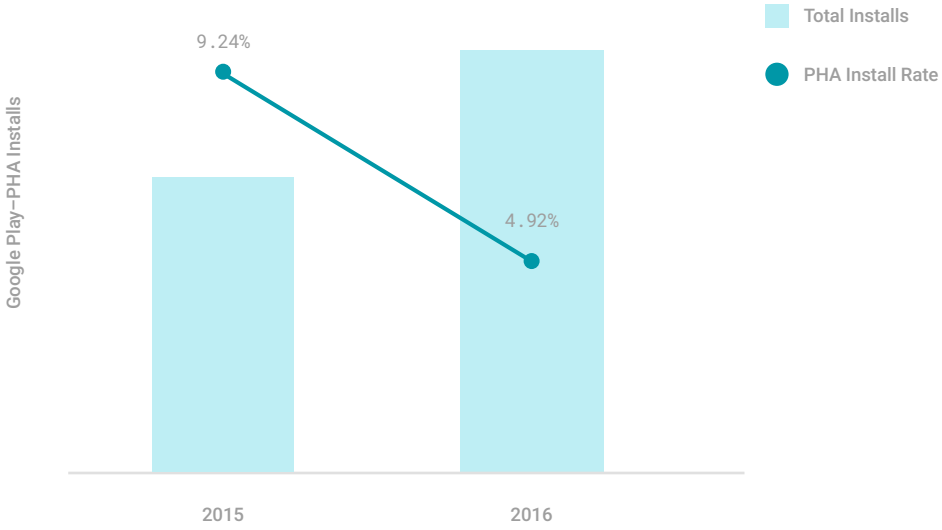
Outside of Google Play: PHA trends

This section covers trends around applications that are installed from a source other than Google Play.

Similar to the 2015 Year in Review report, the PHA installation rates and PHA categories outside of Google Play are significantly higher than those inside of Google Play. Overall, the health of the ecosystem has improved in 2016 compared to the previous year. In 2016, the ratio of PHA installs to total installs decreased by roughly 47.2% from the previous year.

The substantial drop in PHA installs from outside of Google Play can be attributed to reducing installations of the Ghost Push PHA family, described in the 2015 Year in Review. Ghost Push used a network of hostile downloader applications to push trojans onto affected devices. This network of hostile downloaders became less significant in 2016 with an observed reduction in installation attempts of more than 90%. While some new PHA families and variants of the Ghost Push family appeared in 2016, none achieved the level of distribution that Ghost Push had in 2015.

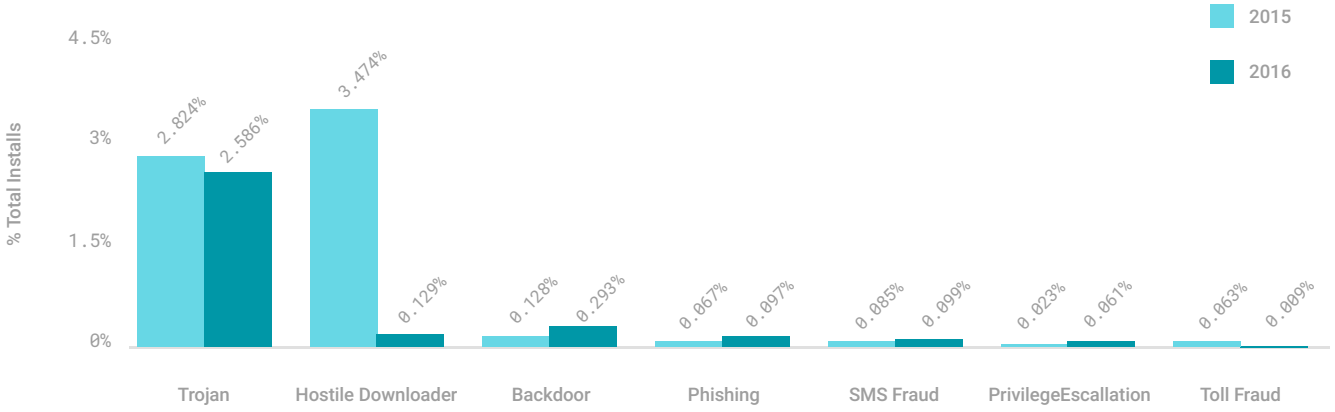
Outside of Google Play—PHA installs



This chart illustrates the year-over-year trends for PHA install rates by PHA category for 2015 to 2016.

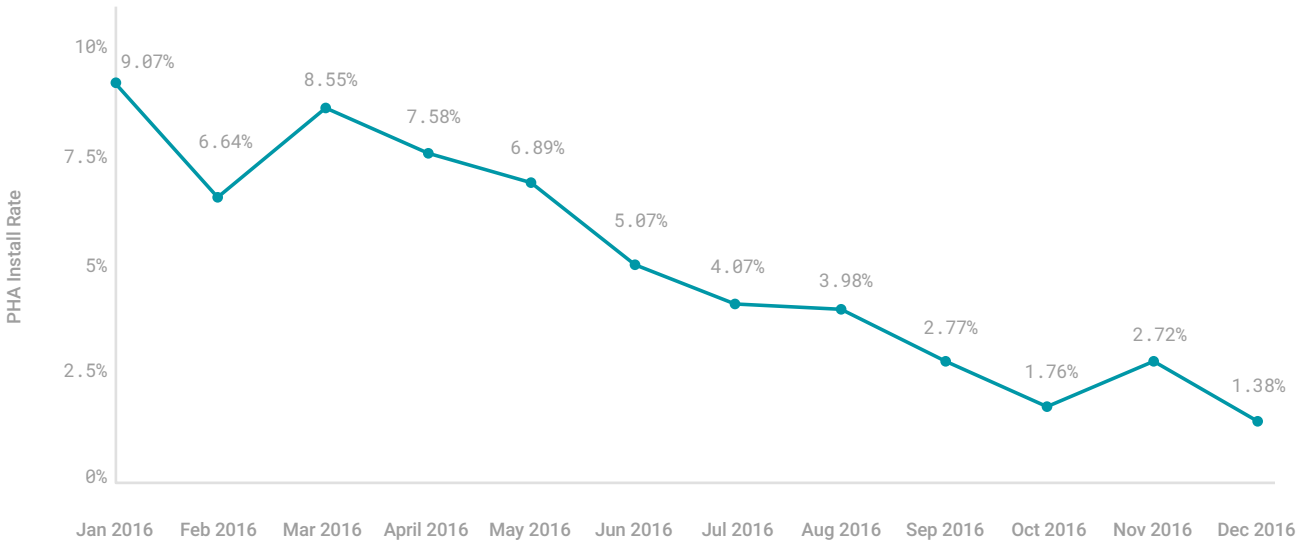
Outside of Google Play—PHA installs, by category [6]

6 Data collection for install trends by category outside of Google Play began in October 2014, so we are only comparing 2015 and 2016.



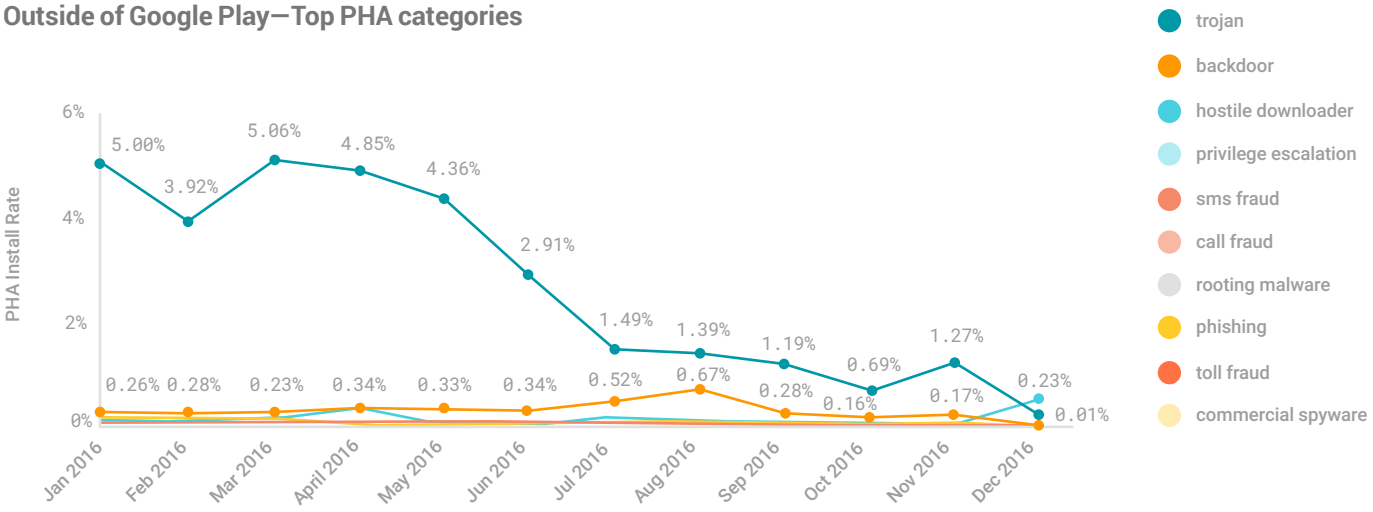
This chart shows the rate of PHA installs through 2016. Note that this is a percentage of total installs, so a given device may install more than one PHA.

Outside of Google Play—PHA installs



In 2016, trojans dominated the overall PHA install rate. Trojans are frequently related to backdoors and hostile downloaders because they trick users into clicking on links that install other apps.

Outside of Google Play—Top PHA categories



Outside of Google Play: Trojans

In 2016, trojans were the most common PHA category from outside of Google Play, representing 79.2% of all PHA installs. This steep rise in occurrence from 2015 (where trojans made up 25.7% of all PHA installs) is due to a change in behavior among the Ghost Push trojan family. While 2015 featured a multi-stage approach that used a hostile downloader app to downloading more apps, the 2016 versions used a single stage approach using trojans to trick users into installing harmful apps. As a result, hostile downloaders (formerly the most prevalent PHA category) declined from 66.9% to 4.0%. In 2016, the most widely distributed variant of Ghost Push would attempt to root a device, and then inject code into other apps in order to execute malicious functionality, such as making purchases on Google Play without user consent.

Outside of Google Play: Backdoors

While backdoor apps are far less common than trojan apps, this PHA category nevertheless saw an increase in 2016. In 2015, backdoor apps made up 0.3% of all PHA installs from outside of Play. In 2016, that number grew to 8.8% making the category the second most common after trojans.

The primary increase in backdoor apps in mid-2016 was caused by a PHA family that Google has internally named Chamois. A Chamois app often disguises itself as a game and monetizes by sending premium SMS. Chamois first appeared in late 2015 and was among the most downloaded backdoor

apps in that year. Chamois's functionality is found in dynamically downloaded plugins, and the Chamois family has continued to evolve throughout 2016. Verify Apps has also continuously improved its detection and removal capabilities for this family.

Outside of Google Play: Country trends

In 2016, each of the top 50 countries had more than 20 million app install attempts from outside of Google Play. The PHA install rates dropped in all 50 countries with Vietnam, Spain, and Egypt improving the most. In this section, we explore the reasons for the improvements in these three countries.

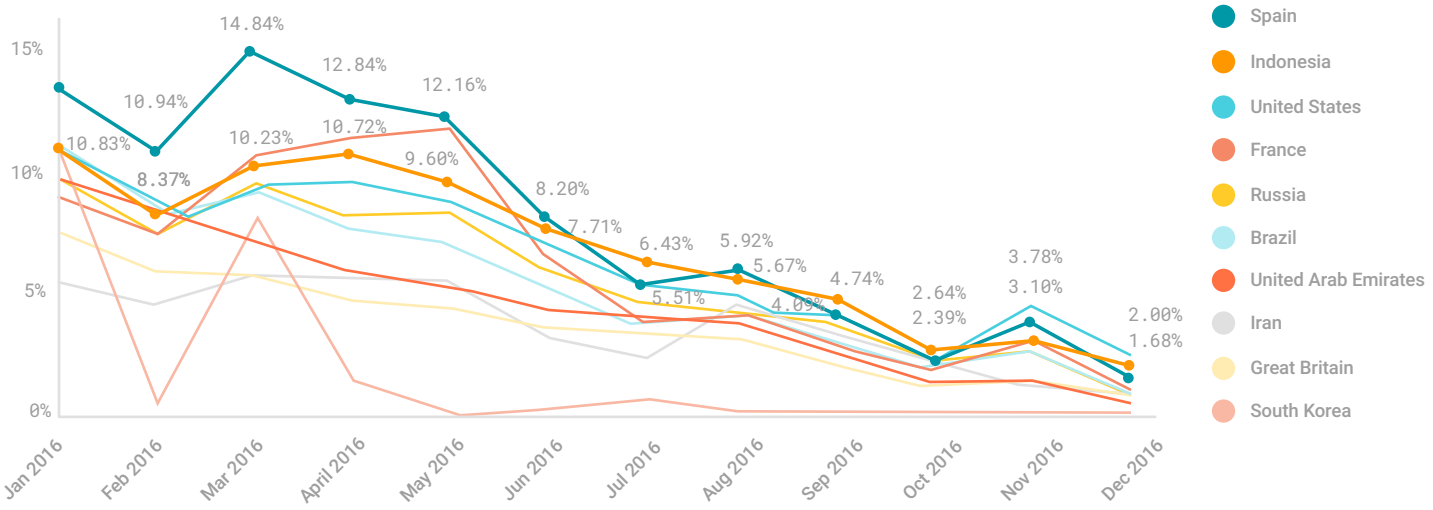
This table presents data for the top 20 countries as determined by overall app installs. The total install magnitude is determined by all applications installs, whether they are PHAs or not. The table lists the percentage of devices for that locale that have a PHA installed, and compares the change in install rate to the rate for the previous year.

Country	PHA Install Rate		Change
	2015	2016	
Brazil	11.29%	4.80%	-6.49%
Canada	3.16%	0.88%	-2.28%
Egypt	22.14%	14.95%	-7.19%
France	9.57%	5.90%	-3.67%
Germany	5.35%	3.42%	-1.93%
Great Britain	7.24%	3.42%	-3.82%
India	12.31%	5.77%	-6.54%
Indonesia	13.72%	6.70%	-7.02%
Iran	9.25%	3.64%	-5.61%
Italy	5.91%	1.31%	-4.60%

Country	PHA Install Rate		Change
	2015	2016	
Japan	0.27%	0.23%	-0.04%
Korea	3.74%	1.79%	-1.95%
Russia	10.56%	5.42%	-5.14%
Spain	15.04%	7.73%	-7.31%
Taiwan	3.35%	1.30%	-2.05%
Thailand	14.38%	7.61%	-6.77%
Turkey	10.03%	5.60%	-4.43%
United Arab Emirates	9.97%	4.60%	-5.17%
United States	9.97%	5.97%	-4.00%
Vietnam	20.39%	7.39%	-13.00%

The 10 countries with the highest number of installs all showed a decline in their PHA install rates in the second half of 2016. In the first half of the year Spain, Indonesia, and France were the three countries with the highest PHA rate from apps outside of Google Play.

Outside of Google Play—PHA installs, by country



Vietnam, Spain, and Egypt

Vietnam is the most improved country for devices that install apps from outside of Google Play. The percentage of installed PHAs from outside of Play declined from 20.4% PHA in 2015 to 7.4% in 2016. Vietnam had one of the highest PHA install rates among the top 50 countries for PHA installs in 2015 while in 2016 it moved to the middle of the pack. The reasons for the improvements in ecosystem health in Vietnam was the decline of the Ghost Push family which in 2015 made up 19 of the 20 most installed PHA in Vietnam in 2015.

The second and the third most improved countries in 2016 were Spain and Egypt with a decline of PHA installs rates by 7.31% and 7.19%, respectively. The PHA install rate in Spain roughly halved from 15.04% in 2015 to 7.73% in 2016. Similarly, there was a significant reduction of PHA install rate in Egypt from 22.14% in 2015 to 14.95% in 2016. The improvements in ecosystem health in Spain and Egypt were also due to the decline of the Ghost Push family, which made up majority of PHA installs in 2015.

PHA family highlights

This section covers six notable PHA campaigns from 2016: Turkish Clicker, Ghost Push, HummingBad, DressCode, Godless, and Gooligan.

Turkish Clicker

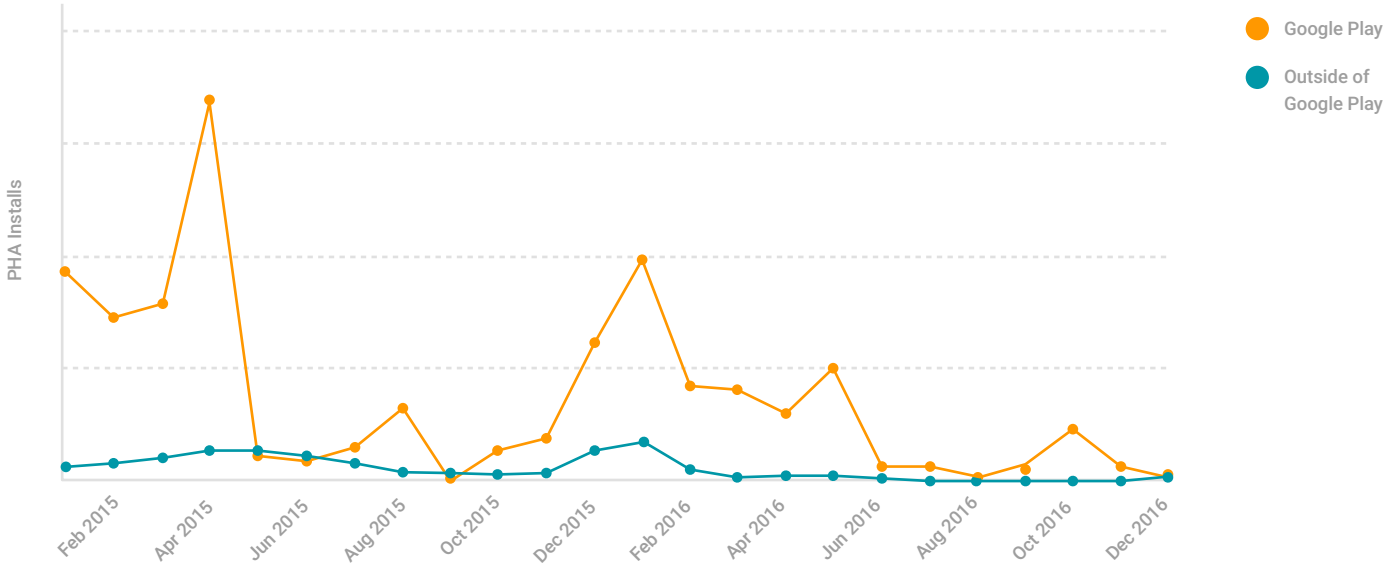
Turkish Clicker is a family of click fraud apps that have been classified as hostile downloaders. The first Turkish Clicker apps were uploaded to Google

Play in mid-2014. This family primarily targeted Google Play users. We began monitoring them in September 2014 after potentially harmful behavior was first detected by Google. Over the next two years, Turkish Clicker's creators continuously tried to get back into Google Play by rapidly changing their apps' code to evade detection and creating more than 1,100 Android Developer accounts with the help of stolen credit cards. The family was first publicly described in a January 2016 blog post by Checkpoint. Between September 2014 and Checkpoint's public disclosure, Google detected, blocked, and removed 1,077 apps in this family. The detection and protections resulted in near-complete elimination of this family by the second half of 2016.

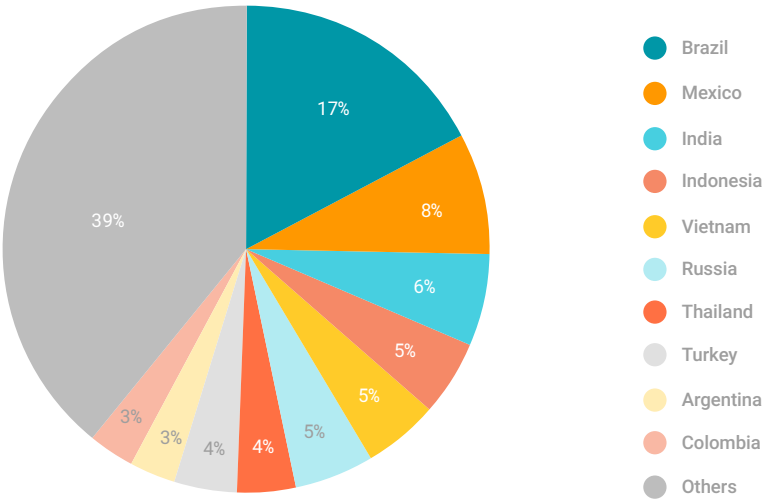
These apps first masqueraded as Turkish free movie apps and later expanded into popular game franchises. However, their real purpose is to make money with click fraud. When a Turkish Clicker app is launched, it connects to a C&C server to download a list of URLs and some Javascript code. That Javascript code clicks every ad on the provided list of websites.

By itself, this click fraud behavior is not a PHA: clicking on HTTP links does not violate any of Android's security boundaries or put user data at risk. This family has been classified as hostile downloaders because, in some cases, the ads unintentionally downloaded other PHAs to the user's device. When Turkish Clicker clicked these apps in the background, the trojans were downloaded to Android devices. Users would still have to find and install these apps to their device to be harmed. Without this additional step, users or their data were not put at risk by Turkish Clicker. Also, Verify Apps would be invoked at the time of installation—so known PHAs would be blocked.

Google Play and outside of Google Play—Turkish Clicker installs



Turkish Clicker installs, by country



Ghost Push

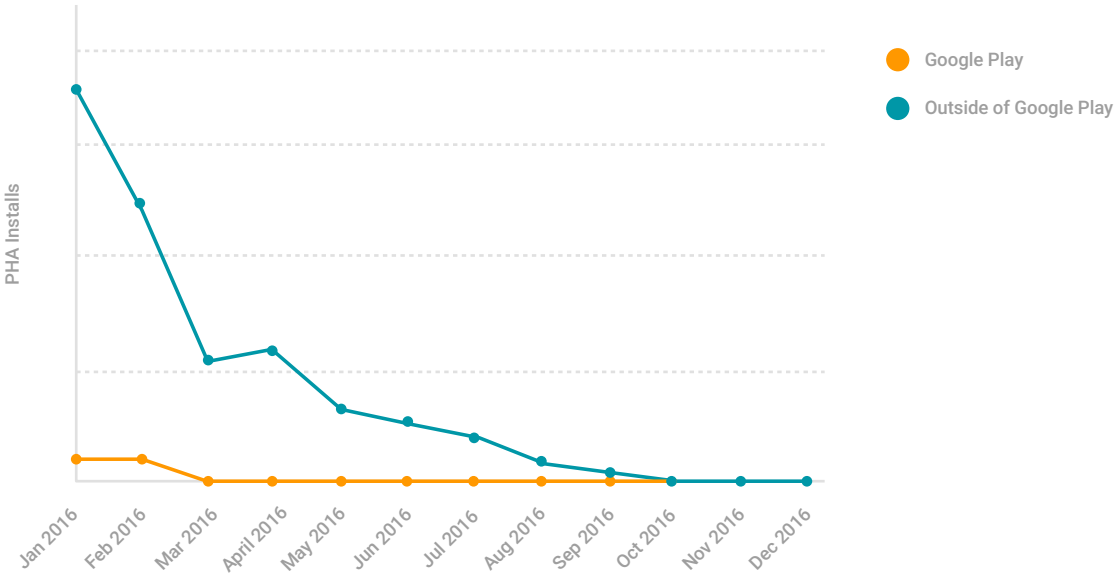
We have monitored the Ghost Push PHA family since October 2014. At first, Ghost Push was a hostile downloader that attempted to download other PHAs. More recently, Ghost Push evolved to root and backdoor devices to defraud per-install advertising networks through forced, background installs of advertised apps. Ghost Push is predominantly found in apps from outside of Google Play.

In the summer of 2015, there was a sudden spike in Ghost Push variants, which contributed to a significant overall rise in install attempts. At this point, Ghost Push also received public attention by affected users.

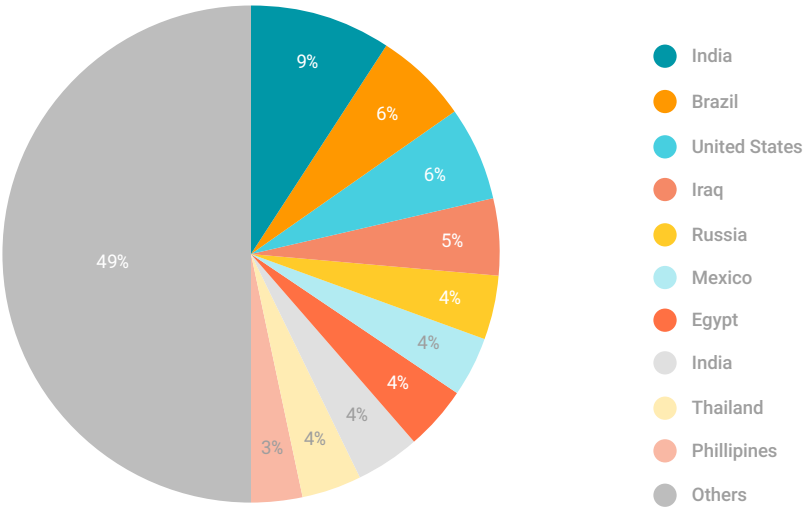
Last year's Android Security Year in Review report also featured Ghost Push as a noteworthy PHAs. By late 2015 it was in decline and nearly disappeared in 2016. By the end of 2016, Ghost Push installation attempts dropped to near zero. We believe that Ghost Push is continuing to evolve, and that it has split into several distinct families in an effort to avoid detection. We are currently tracking several families that appear to use similar monetization schemes and technical capabilities to those we've seen in Ghost Push. These families are currently being investigated and blocked as they evolve, so we are not able to provide details at this time.

We believe that Ghost Push originated in China. Ghost Push primarily affected India, Brazil, United States, Iraq, and Russia in Google Play.

Ghost Push installs



Ghost Push installs, by country

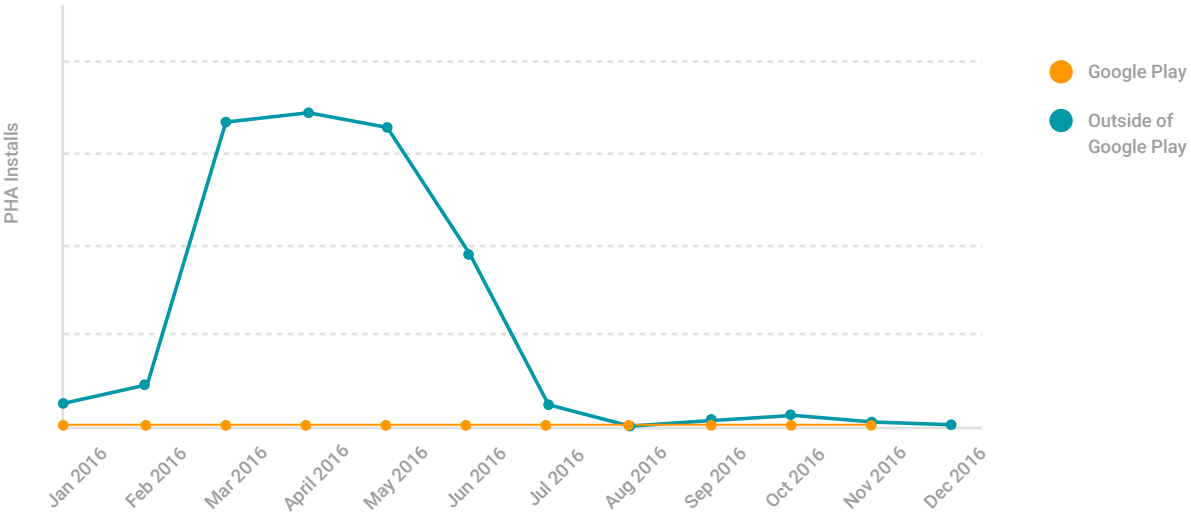


HummingBad

The HummingBad PHA family operates very similarly to Ghost Push: devices are rooted without user consent and then used to secretly abuse third-party properties on the web. However, HummingBad's monetization method relies on click fraud. Like Ghost Push, we believe this family originated in China.

HummingBad spread almost exclusively outside of Google Play. Of the 24,000 HummingBad apps with about 379 million installation attempts (25.1% of which were blocked or warned by VerifyApps), only one app was uploaded to Google Play and was suspended before it reached 50 downloads. Despite the high number of installation attempts, the number of affected devices is far lower. Due to the nature of its distribution, affected devices would generally receive multiple HummingBad installation attempts, many of which were blocked by our SafetyNet on-device protection. Additionally, because HummingBad needs to root a device in order to perform harmful actions, devices weren't immediately affected just because the app was installed on the device. Rather, the app needed to be installed on the device, *and* a security vulnerability needed to be present for the app to exploit.

HummingBad installs



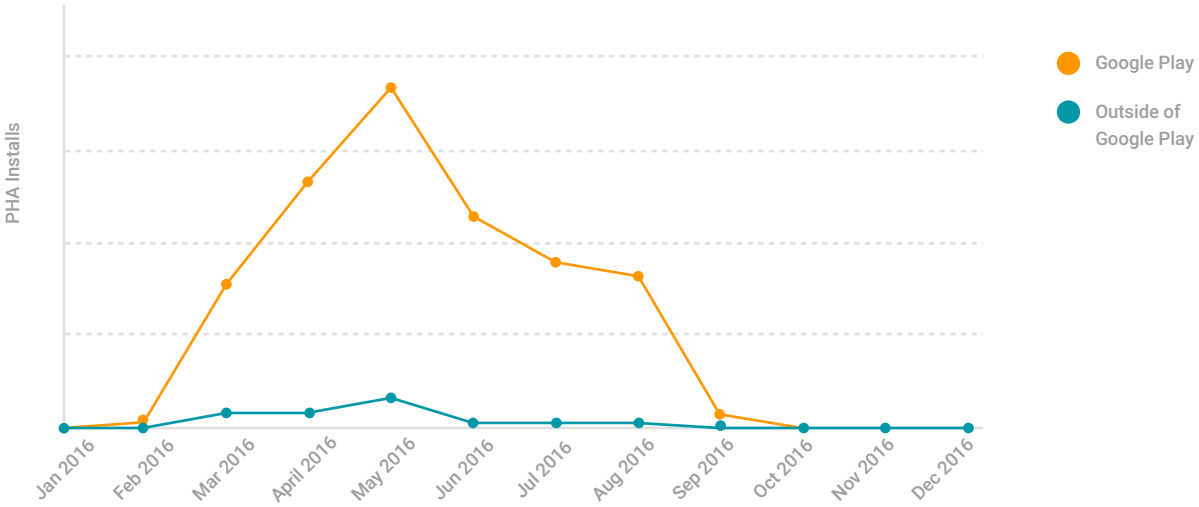
HummingBad peaked between February and July of 2016. As with Ghost Push, HummingBad primarily spread in Southeast Asia. Because this PHA was mostly distributed outside of Google Play, we do not have detailed country data for the year.

DressCode

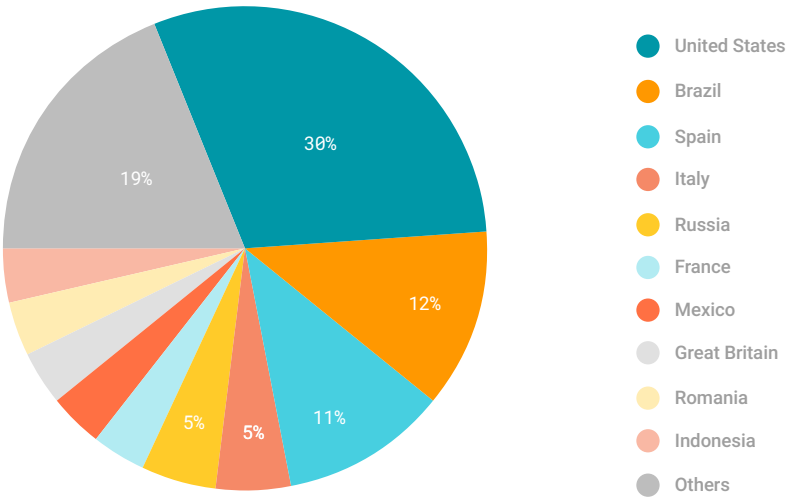
DressCode is an unusual PHA because it does not exploit vulnerabilities in traditional ways. It uses legitimate functionality to open a SOCKS proxy network connection to access the local network a device is connected to. Due to this functionality, members of the DressCode family are considered backdoors. This PHA family can also receive commands from an external Command & Control server, so the exact behavior can be modified depending on what commands are received.

For a device to be affected, the user must download and install an app from this PHA family. To combat this, we monitor new applications for this behavior and added rules to Verify Apps to remove or warn users for 1,115 DressCode apps. Due to perceived potential for harm we began warning users about apps in this family. Despite this caution, we have not observed any malicious activity from DressCode apps.

DressCode installs



DressCode installs, by country

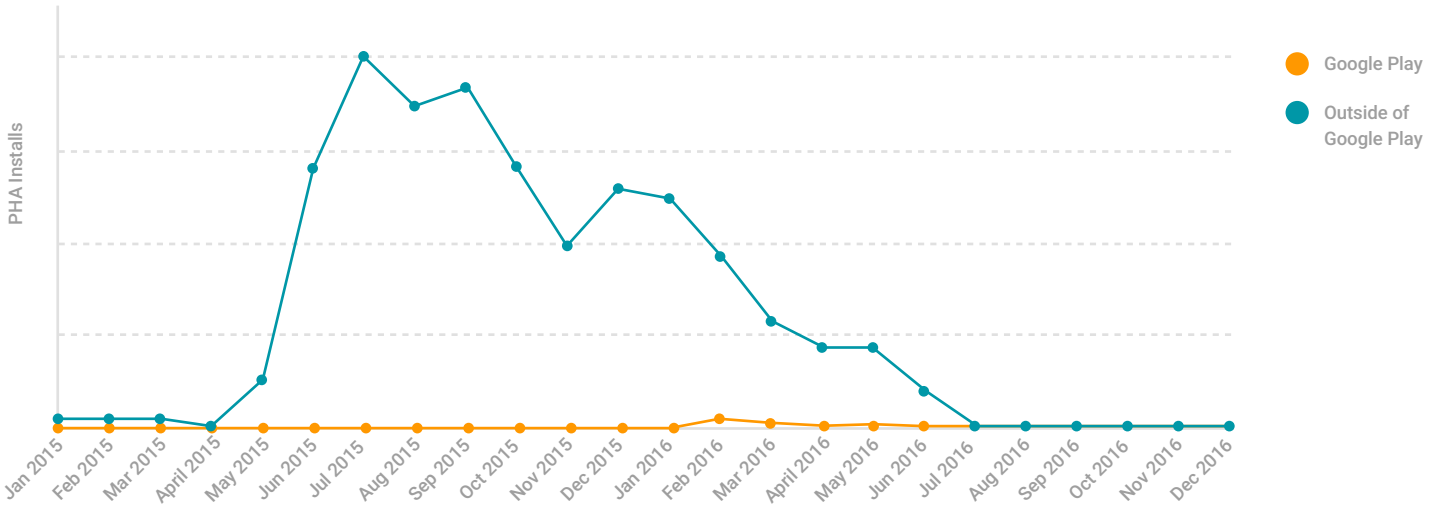


Godless

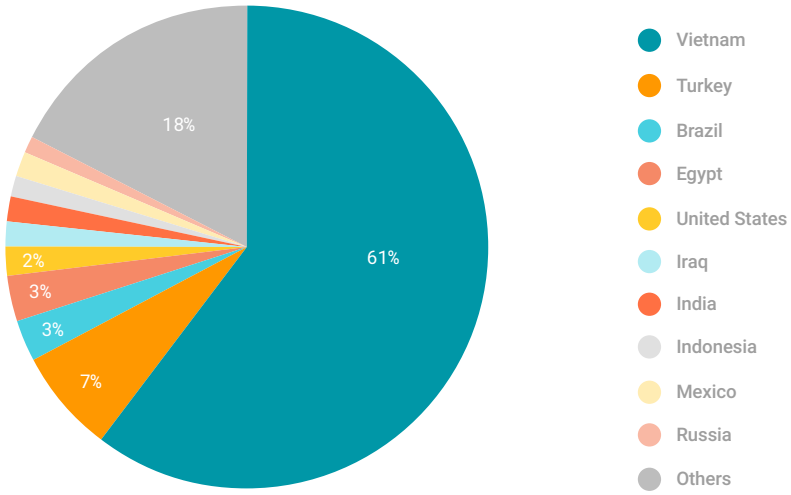
The Godless PHA family attempts to root a device by trying several different known rooting vulnerabilities. If it successfully roots the device, it attempts to install other apps without user permission.

This PHA relies primarily on rooting vulnerabilities, CVE-2014-3153 and CVE-2015-3636. Both of these vulnerabilities have been publicly disclosed for some time, and are patched as of September 2015. Verify Apps alerts users if they attempt to download apps using these exploits. Devices running 6.0 and above are not affected by this PHA family.

Godless installs



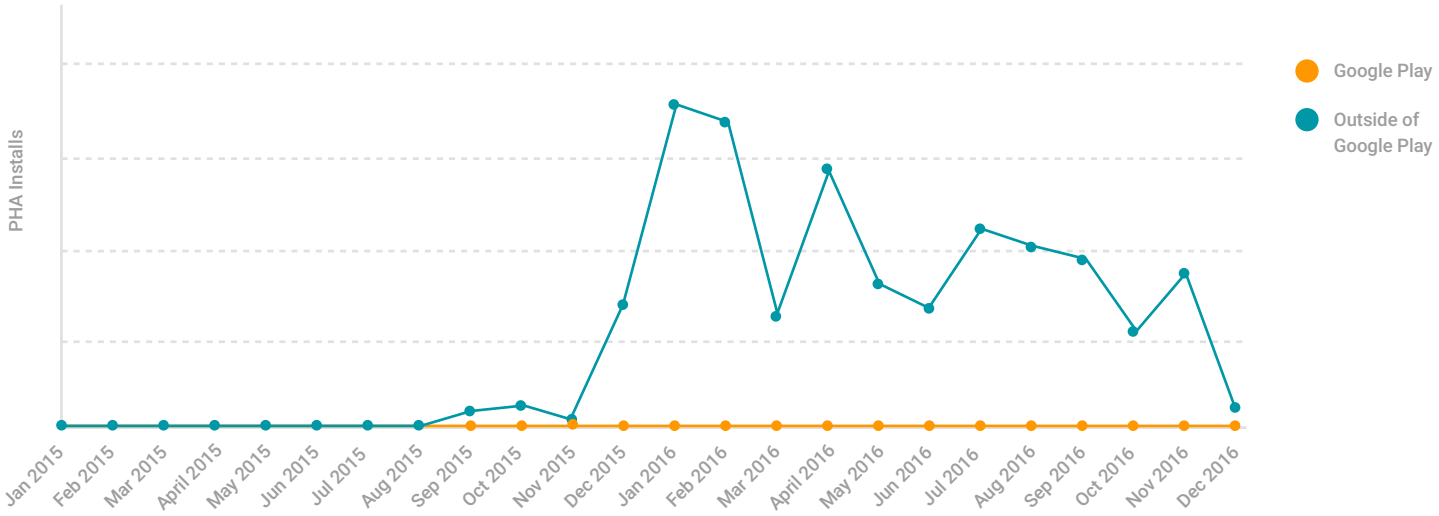
Godless installs, by country



Gooligan

Gooligan is a variant of the Ghost Push family that steals OAuth tokens to impersonate the user on Google Play to post false reviews and download apps. Its primary function is to falsely promote apps. Google and security research firm Checkpoint collaborated to identify affected users, revoke the compromised tokens, and dismantle Gooligan's command and control infrastructure. For more details, see this blog post.

Gooligan installs



Because this PHA was mostly distributed outside of Google Play, we do not have detailed country data for the year.

Noteworthy Vulnerabilities

This section covers some platform vulnerabilities that received major media attention. Despite the attention, many were not exploited or were very contained due to protections that are built into the Android Platform and the security services provided by Google.

Adups data transmission

Some Android device manufacturers use third-party over-the-air (OTA) update services to deploy software updates to their devices. This OTA software is not part of the core Android platform. In November, 2016 it was discovered that some manufacturers who used the Adups OTA service incorrectly implemented a configuration that sent sensitive information from devices to Adups servers. Once notified, the device manufacturer updated their devices to no longer send this type of data to Adups.

CVE-2015-1805

In February 2016, security researchers discovered a way to exploit CVE-2015-1805 and notified Google that it was being used externally by a popular device-rooting application, Kingroot. Google released an emergency patch in March to address this issue and added rules to our PHA detection systems to alert users to apps that use this exploit

Dirty Cow

Dirty Cow (CVE-2016-5195) was a privilege-escalation vulnerability in the Linux kernel that made Android devices susceptible to rooting. A local attacker could have used this vulnerability to gain write access to read-only memory, such as the memory cached versions of executable files, and increase their privileges on the system. This vulnerability required a user to download an app that took advantage of the loophole to gain root access.

Quadrooter

Quadrooter is a set of four vulnerabilities (CVE-2016-2503, CVE-2016-2504, CVE-2016-2059 and CVE-2016-5340) that affect a Qualcomm chipset widely used among Android devices. These vulnerabilities were publicly disclosed at DEF CON 24 in August 2016.

Google investigated all four of these vulnerabilities and determined that:

— Android devices running 6.0 or higher with a security patch level of August 05, 2016 or higher are not vulnerable to CVE-2016-2059, CVE-2016-2503, and CVE-2016-2504.

— Android devices running 4.4.4, 5.0.2 or 5.1.1 with a security patch level of August 05, 2016 or higher are not vulnerable to CVE-2016-2503 and CVE-2016-2504, and are only vulnerable to a mitigated, low-severity version of CVE-2016-2059.

Patches for CVE-2016-2059, CVE-2016-2503, and CVE-2016-2504 were released in July. Patches for CVE-2016-5340 were released in October.

For a device to be affected, a user must download and install a PHA that takes advantage of one of the vulnerabilities.

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—from SoC manufacturers, to Android device manufacturers, to mobile network operators—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 more secure and protects Android users across the world.

android