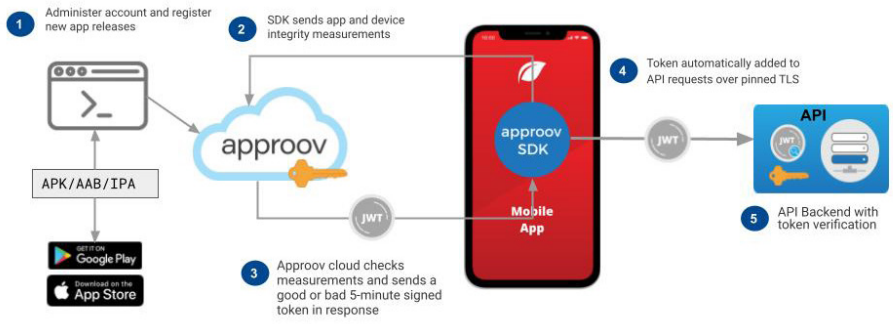
approov
MOBILE APP PROTECTION

Approov Mobile App Protection & Google SafetyNet: Comparisons and Benefits of Integration

This overview describes the Approov Mobile App Protection and Google SafetyNet technologies and how they work together to provide even better protection for your applications and the APIs they use.

Approov Mobile App Protection

Approov Mobile App Protection ensures that all mobile API traffic does indeed come from a genuine and untampered mobile app, running in a safe environment. Doing this blocks all scripts, bots and modified or repackaged mobile apps from abusing an API.



Approov consists of an SDK which is integrated into the customers' mobile app, and a cloud service which manages a dynamic attestation process on the app. Only positively attested app instances are provided with a valid, short lived, and cryptographically signed Approov token. This token can then be used to gain access to backend API services, ensuring only legitimate API requests are able to do so successfully. Since an industry standard JSON Web Token (JWT) is used, the check can be easily integrated into existing backend systems, either in a cloud service or at the network perimeter in an API gateway, CDN, WAF or similar.

The attestation process proves the integrity of the app, ensuring tampered, cloned or re-packaged versions fail. Further protection is provided against man-in-the-middle attacks, against apps that are running on rooted/jailbroken phones or an emulator, under the control of a debugger or in the presence of instrumentation or code hooking frameworks. The attestation process used by Approov is invisible to the end user and does not significantly impact app performance.

Approov customers are able to make fine grain choices about which attributes are allowable for their running apps, adaptable to particular end user markets and conditions. Metrics are gathered live from devices and made available to Approov customers to help them understand their user base and create good security access policies.

Approov protects against a very wide range of threats by its attestation approach, establishing the integrity of the requests being made. This blocks automated bots and scripts which might attempt to gain access to backend resources by spoof-

ing API traffic as if it was coming from a real app. Such traffic will usually employ valid user credentials so user authentication mechanisms offer no protection. Indeed such approaches are typically used by attackers for illicit automated account takeover or data scraping activities. Approov also protects against unauthorized access via modified or fake repackaged apps that may be subverting business models, redirecting ad revenue and putting end users at risk.

Google SafetyNet Attestation API

Google SafetyNet Attestation API is designed to check device integrity to ensure it is secure and compatible. In effect this is an advanced root check. It verifies the file system, finding changes indicative of rooted devices and performs various other checks for signs which indicate that the device has been compromised.

Since a significant percentage of non-nefarious users have rooted their phones in order to add features and capabilities, the dependence on a root check to determine the 'goodness' of the device is problematic. Aggregated data from Approov customers indicates a rooted phone occurence of 10-15% of the total Android user base in various markets. In some vertical markets, such as financial services and healthcare, it is understandable and acceptable to block API access for rooted devices. For more general retail sectors, customer stickiness is a key metric and therefore security policies tend to consider multiple factors before deciding to rate limit or block a particular customer. The Android installed base has many shades of gray when it comes to the "goodness" of the devices and its apps and all businesses must consider this as they roll out security solutions.

SafetyNet's file system checks are advanced, but not always definitive. Some popular rooting apps such as Magisk use systemless rooting approaches and have been able to cloak themselves sufficiently from SafetyNet to evade detection.

Google SafetyNet Attestation API does provide an attestation of the running app via the `apkDigestSha256` feature. However, this is only reliable if full integrity is reported which is not possible if it is running on any kind of unusual or rooted devices. Many of our customers do allow execution on rooted devices, and rely on Approov to definitively tell them about the integrity of their running app, and whether it appears to be under specific attack, hooking or modification. They have no interest in the integrity of the device as a whole. SafetyNet is not able to do this since, unlike Approov, it does not specifically analyze the memory map of running apps to detect instrumentation frameworks. It relies on the fact that they can only run on rooted devices, and it attempts to detect root via an overall file system check

Since it is part of Google Mobile Services (GMS), SafetyNet will only run on devices that support these services. In some markets, such as the far east, there are a significant number of devices that do not have this available but need to be supported. Thus a client side check is required for SafetyNet availability with a bypass if it is not. Unfortunately, such a bypass creates a fundamental security weakness since this can be exploited by attackers on any device by faking this check and thus undermining the entire security model.

In common with Approov, the result of a SafetyNet is a cryptographically signed attestation assessing the device's integrity. The server side is responsible for providing a one time nonce value to initiate this process and to verify the result using custom server side logic. This is described in the Google documentation, which notes that a number of different special cases have to be dealt with in the implementation. Thus it is not possible to implement the verification at the network perimeter in an API gateway, WAF, CDN or similar due to this complexity. This also makes integration into managed backend systems more complex.

There are a limited number of attestation calls that can be made with a standard free API key — currently 10K per day — so to use at scale it is likely that a paid level would be required. Also, since Google SafetyNet performs a full analysis of the hashes of the OS image it is quite slow (usually several seconds). This means that it cannot be run continuously for every API call and care is needed in its use to mask this latency from the user. Furthermore, SafetyNet tokens are quite large (several KB in size) and therefore cannot practically be included in all API requests. Thus in practice it is necessary to only perform the SafetyNet check once, or very infrequently, and then persist the result and associate it with some other unique ID associated with the device. This of course then weakens the security, since if it is possible to steal an ID from a SafetyNet verified device then it would be possible to use that in any subsequent API request and evade all checking.

It should also be noted that SafetyNet does not provide any additional support for proving that the SafetyNet token itself is not being intercepted by a Man-in-the-Middle. Any such interception would then allow an attacker to spoof requests as if they were being made from a SafetyNet

protected device.

SafetyNet and Approov Together

Approov provides an option for integration with SafetyNet. This leverages all of the device integrity checking benefits of SafetyNet, whilst retaining the ease of integration of Approov. To add SafetyNet protection, all that is required is to add a dependency to the SafetyNet GMS library in your app, and use the Approov CLI to provide the SafetyNet API key from the Google developer console. Approov then automatically transmits this API key to your running apps (so that it may be dynamically updated without app modifications) and ensures each device performs a SafetyNet check, in addition to all of the Approov protections. As discussed, this checks the integrity of the device and the identity of the calling app. If desired you can force Approov to always require SafetyNet or to allow it to work gracefully on Android devices that do not support it. A device bound server encrypted token is stored in the app to allow it to subsequently prove that it has passed the SafetyNet checks, without incurring the latency or payload costs of SafetyNet for subsequent requests. Approov implements all of the server side logic for implementing SafetyNet, and provides a simple JWT with frontend integrations for adding these automatically to requests, whilst dynamically pinning connections, and numerous backend integrations for easy verification token verification.

Summary

SafetyNet is a useful addition in the armory of weapons which are available to enterprises to thwart attacks against Android apps. It is particularly useful for providing a view on device integrity. However, any business that is not prepared to implement a blanket ban on rooted devices, should note that SafetyNet should not be relied upon to solely judge the validity of a platform, as acknowledged in Google's own documentation:

"You should use the SafetyNet Attestation API as an additional in-depth defense signal as part of an anti-abuse system, not as the sole anti-abuse signal for your app."

When using SafetyNet alone, businesses that decide to reject transactions from rooted devices must be prepared to block a significant percentage of their installed base—including many genuine customers. A better approach is to create a more comprehensive security protocol that considers a number of detailed factors before deciding whether or not to allow a transaction. Moreover, SafetyNet presents a number of practical challenges during implementation, such as the latency of the attestation process which makes it impractical to protect all requests without considerable additional implementation complexity.

Approov, together with SafetyNet, provides a better solution. By delivering the server side operations as a service, providing finer grain detail and checking on the health of the actual app that is running, the full Android ecosystem can be supported and secured with minimum integration effort. Furthermore, since Approov also supports iOS a unified approach can be taken across both of these major platforms.