Security on Yealink MeetingBoard

Introduction

Yealink MeetingBoard is based on the Android operating system. As many users may worry about the security of the device and the Android system, Yealink devotes ourselves to providing users with a secure system environment based on Android. No matter the Android system or the related service on Android provided by Yealink, they are both equipped with a multi-layer security strategy with unique ways to keep data and devices safe.

The following security solutions provided by Yealink are introduced in details in this guide:

- Device integrity
- App security
- Android security updates
- Hardware-backed KeyStore and KeyChain
- Network security
- Data protection

For detailed security assessment report, please refer to Yealink MeetingBoard Attestation Letter by Independent Pentration Testing Lab NetSPI.

Device Integrity

Device integrity can protect the MeetingBoard device from any changes to the operating system. With companies using the devices for essential communication and core productivity tasks, keeping the OS secure is important. Device integrity is reflected in the following two aspects:

- Verified Boot

- Sandboxing

Verified Boot

Yealink MeetingBoard supports the Verified Boot, which is Android's secure boot process that verifies system software before running it. It mitigates attacks against devices by providing a boot process that verifies system software using a hardware root of trust. This makes it more difficult for software attacks to result in a persistent OS compromise and provides users with a safe state at boot time.

Each Verified Boot stage is cryptographically signed. Each phase of the boot process verifies the integrity of the subsequent phase before executing that code. As of Android 7.0, the locked boot loader is used to fully boot compatible devices only when the operating system satisfies the integrity check.

Yealink MeetingBoard also enables the rollback protection, which is required for those devices running on Android 9.0 and above. The rollback protection state is stored in the tamper-evident storage. Any kernel compromise or physical attack cannot downgrade the OS to an older and vulnerable version.

Sandboxing

All Android apps is running inside many of sandboxes, which means our OS can prevent malicious or buggy App code from compromising other Apps or the rest of the system. Besides, system components run in the least-privileged sandboxes to prevent compromises in one component from affecting others. The remotely reachable components, for example, the WebView is isolated in

their own restricted sandbox.

Android employs the following three sandboxing techniques:

- Security-Enhanced Linux (SELinux)
- Seccomp
- File-system permissions

SELinux

Android uses SELinux to enforce mandatory access control (MAC) over all processes and Apps, even processes running with root and superuser privileges. SELinux provides a centralized auditable security policy that can be used to strongly separate processes from one another.

Android devices implement SELinux policy on a per-domain basis in enforcing mode—no permissive mode domains are allowed. Illegitimate actions that violate policy are blocked and all violations (denials) are logged by the kernel. They are then readable using the dmesg and logcat command-line tools.

Seccomp Filter

In conjunction with SELinux, Android uses Seccomp to further restrict access to the kernel by blocking access to certain system calls. As of Android 7.0, Seccomp was applied to processes in the media frameworks. It blocks access to certain system calls, such as swapon/swapoff, which have been implicated in some security attacks. Besides, it blocks the key control system calls, which are not useful to Apps. As of Android 8.0, a Seccomp filter is applied to all apps, enforcing a whitelist of system calls that are allowed.

Filesystem Sandboxing

Android uses Linux filesystem-based protection to further isolate application resources. Android assigns a unique user ID (UID) to each application and runs it as that user in a separate process. By default, Apps cannot access each other's files or resources just as different users on Linux are isolated from each other.

App Security

Apps are an integral part of Yealink MeetingBoard, and users increasingly rely on MeetingBoard for core productivity and communication tasks. Yealink MeetingBoard provide multiple layers of application protection.

App Signing

Android requires that all apps be digitally signed with a developer key before installation. Android 9.0 supports APK key rotation, which gives apps the ability to change the signing key as part of an APK update.

App Updating

When the system installs an application update, it will compare the certificate that Android uses it to identify the application's author in the new version with the one in the existing version. If it matches, the update will be complete.

App permissions

Permissions provide transparency about what resources or information apps wish to access. For protecting users' privacy, for apps to access system

features, such as camera and the web, or user data, an Android app must explicitly request permission.

Hardware-backed KeyStore and KeyChain

KeyStore

The Android KeyStore class lets you manage private keys in secure hardware to make them more difficult to extract from the device.

Keystore supports symmetric cryptographic primitives such as AES (Advanced Encryption Standard) and HMAC (Keyed-Hash Message Authentication Code) and asymmetric cryptographic algorithms such as RSA and EC. Access controls are specified during key generation and enforced for the lifetime of the key. Keys can be restricted to be usable only after the user has authenticated, and only for specified purposes or with specified cryptographic parameters.

For devices that support a secure lock screen and ship with Android 7.0 or higher, KeyStore must be implemented in secure hardware. This guarantees that even in the event of a kernel compromise, KeyStore keys are not extractable from the secure hardware. Devices with Android 9.0 and above use the hardware-backed Keymaster 4, which offers additional protections against tampering.

KeyChain

Android 4.0 introduced the KeyChain class to allow Apps to use the system credential storage for private keys and certificate chains. KeyChain is often

used by Chrome, Virtual Private Network (VPN) Apps, and many enterprise Apps to access keys imported by the user or by the mobile device management App. Android 10 introduces several improvements to the KeyChain API. When an app calls KeyChain.choosePrivateKeyAlias, devices now fillter the list of certificates a user can choose from based on the issuers and key algorithms specified in the call Whereas the KeyStore is for non-shareable app-specific keys, KeyChain is for keys that are meant to be shared across profiles. For example, your MeetingBoard management agent can import a key that Chrome will use for an enterprise website.

Network Security

Yealink MeetingBoard provides network security for data-in-transit to protect data sent to and from devices. The secure communications over the Internet for the Teams or other Apps (Company Portal) by supporting the Transport Layer Security, including TLS v1.2 and TLS v1.3 is supported for MeetingBoard.

Wi-Fi

Android 10 supports the Wi-Fi Alliance's Wi-Fi Protected Access version 3 (WPA3) and Wi-Fi Enhanced Open standards. WPA3 and Wi-Fi Enhanced Open improve overall Wi-Fi security, providing better privacy and robustness against known attacks.

VLAN

VLANs is supported for MeetingBoard, which can be used to create secure user groups and prevent others outside of the broadcast domain from receiving sensitive data of the MeetingBoard. They can also be used to enhance firewall

functions and restrict network access for one or more users. By segregating devices into VLANs, security filters can be implemented in the network to prevent the devices from receiving unnecessary traffic from other devices. This helps prevent disruption due to DoS attacks or attempts to compromise the devices. It also allows locking down access to configuration and signaling servers to only allow access from the devices.

Third-party Apps

MeetingBoard running Android 10 supports Network security configuration, which lets Apps easily customize their network security settings in a safe, declarative configuration file without modifying app code. You can configure these settings for specific domains, for example, opting out of cleartext traffic. This helps prevent an app from accidentally regressing due to changes in URLs made by external sources, such as backend servers. The MeetingBoard is on the Android 10, with all apps installed are encrypting traffic by default.

This safe-by-default setting reduces the application attack surface while bringing consistency to the handling of network and file-based application data.

Certificate Handling

As of Android 7.0, all new devices must ship with the same certificate authority store.

Certificate authorities (CA) are a vital component of the public key infrastructure used in establishing secure communication sessions via Transport Layer Security (TLS). Establishing which CAs are trustworthy—and by extension, which digital certificates signed by a given CA are trustworthy—is critical for secure communications over a network.

With Android 7.0, compatible devices trust only the standardized system CAs

maintained in AOSP. Apps can also choose to trust user- or admin- added CAs. Trust can be specified across the whole app or only for connections to certain domains. These protections are further improved through preventing apps that target Android 9.0 from allowing unencrypted connections by default.

When device-specific CAs are required, such as a carrier app needing to securely access components of the carrier's infrastructure (e.g., SMS/MMS gateways), these apps can include the private CAs in the components/apps themselves.

802.1x

We also support 802.1x authentication on our MeetingBoard, please refer to Yealink 802.1X Authentication.

Data Protection

Android uses industry-leading security features to protect user data. The platform creates an application environment that protects the confidentiality, integrity, and availability of user data.

Encryption

Configuration files may contain sensitive information such as user accounts, login passwords, or registration information. To protect sensitive information from tampering, user can choose to encrypt configuration files. Yealink provides tools for encrypting configuration files on the Windows platform and Linux platform respectively. After that, the administrator can deploy MeetingBoard

using encrypted configuration files and AES keys.

For more information, please refer to Yealink Configuration Encryption Tool User Guide.

Android Security Updates

Yealink MeetingBoard has updated the patch of Android 10 system at 2021.Nov, and it will be keep updating the security patches of Google update simultaneously with the firmware update of MeetingBoard.

*Part of the content of this article refers to "Android Security White Paper 2019", for more information, please visit: https://static.googleusercontent.com/media/www.android.com/zh-CN//static/2016/pdfs/enterprise/Android_Enterprise_Security_White_Paper_2019. Pdf

Customer Feedback

Yealink is striving to improve our documentation quality and appreciate your feedback. Email your opinions and comments to DocsFeedback@yealink.com.

Technical Support

Visit Yealink WIKI (http://support.yealink.com/) for the latest firmware, guides, FAQ, Product documents, and more. For better service, we sincerely recommend you to use Yealink Ticketing system (https://ticket.yealink.com) to submit all your technical issues.