Yealink Wireless Presentation Pod Security White Paper

Yealink Security Team

Yealink Network Technology Co., Ltd.

Table of Contents

Overview

This white paper aims to explain the security design, testing and improvements of the WPP(Wireless Presentation Pod) series of Yealink devices. The Yealink security team attaches great importance to the security performance of devices. It regularly conducts internal tests and hires third-party security agencies to perform comprehensive security vulnerability scanning and penetration testing, and releases security performance improvements through hot-fix patches or version updates.

This white paper introduces the security performance of the WPP series of devices from various aspects. The security test data cited include test reports of well-known security agencies in the industry and test results of authoritative security tools. Yealink will update the white paper from time to time, and the latest version of this white paper will be available on Yealink's official website.

1 Hardware Interface Security

1.1 Debugging Interface

- Disabling of the device’s debugging interface by default

All hardware debugging interfaces are disabled by default after release from the factory to prevent unnecessary physical debugging interface exposure and information transmission.

- Removal of the debugging interface's silk screen

The silk screen has been removed from the PCB of all devices to prevent reverse engineering.

2 System Security

2.1 Address Space Layout Randomization

Address space layout randomization (ASLR) protection is enabled to prevent buffer overflow.

2.2 Partition Encryption

The sensitive information of the device has been encrypted and stored in a memory chip (flash, nand, emmc, etc.) through bit-level data encryption of the operating system partition, and the relevant information cannot be obtained by cracking

2.3 Secure Boot

The device supports secure boot. When booting, the firmware or Flash key partitions are legally loaded and verified to ensure that the system can start normally after successful security verification. It also prevents others from gaining control of the device through access to the SHELL interface by modifying boot parameters

2.4 Firmware Security

The firmware data is encrypted with a high-strength encryption algorithm, and the firmware cannot be cracked or tampered with.

The firmware installation process has security checks that identifies incorrect firmware and tampered firmware and prevents their installation.

2.5 System Security Testing

Penetration tests are conducted to ensure that firmware analysis, malicious firmware updates, etc. are all safe.

Before the official release of the firmware, the Yealink security team also uses a variety of mainstream anti-virus software to scan the firmware to ensure that the firmware is not infected or embedded with viruses/Trojans. For details of the scan report, please refer to Appendix 2.

Yealink's security team and IT department regularly perform static and dynamic vulnerability scanning and penetration testing on the production environment and the company's internal network environment to ensure that the software development, firmware packaging and device production processes are completed in a secure network environment.

3 Coding Security

3.1 Universal Coding Security

- **Random number generation function**

Both device systems and applications use internal random algorithms to generate random numbers required for authentication or encryption to ensure security

- **String or memory manipulation functions**

The system and application use safe string or memory manipulation functions to prevent the use of strings or memory functions that do not perform bounds checking, which may be used by attackers for buffer overflow attacks

- **Format string function parameters**

The system and application use format string functions without external controllable variables as parameters to prevent serious harm caused by format string vulnerabilities.

3.2 App Coding Security

- **Stack cookie overflow protection**

Linux program code is used for compilation, and the CANARY stack overflow protection option of the Linux program is enabled by default

- **Stack non-executable protection**

Linux program code is used for compilation, and the NX stack non-executable protection option of the Linux program is enabled by default.

- **Base address random loading protection**

The PIE application base address random loading protection is enabled. Position-independent code / position-independent executable

- **Linux program code compilation**

When compiling Linux program code, the Strip function is used to delete the debug symbol table to increase the difficulty of reverse analysis and reduce the program size.

- **System call function parameters**

When using system call functions in coding, external controllable parameters should be filtered to prevent command injections

3.3 Codebase Management

Device-related code bases have strict permission management mechanisms and company red line requirements. In order to prevent source code leakage, uploading to public or semi-public services such as Github, Gitee and other public code repositories without permission is prohibited.

3.4 Third-party Components/Libraries

When third-party components/open source software are involved in a device, versions of prohibited categories are not used, and there are no residual, unused, or low-version third-party components. Externally disclosed vulnerabilities have been fixed in accordance with official patches, versions, or schemes

3.5 Port Security

- Port attack defense

A scan has been carried out with the Nmap port scanning tool, and no open ports were found.

- Access security

All physical interfaces, communication ports, and protocols that can manage terminals have access authentication mechanisms. The login service provided by the terminal should require the user to re-authenticate if the user has not carried out any operations for a period of time.

3.6 Coding Security Specifications

In addition to the above-mentioned security design, Yealink has strict internal coding security specifications for the R&D team. During every code update, the code is reviewed and a reliability verification is carried out.

4 Application Security

4.1 Session Security

Yealink's internal session security management mechanism is adopted, and session security is also ensured through various methods, such as high-quality pseudo-random number generator, high-strength encryption function, etc.

4.2 Data Management

The processing of data by the device is done on the device side. In other words, the final input processing (authentication), validity (data legitimacy), etc. are all performed on the terminal.

5 Data Security

5.1 Encryption Algorithm

The device uses secure encryption algorithms, and the key length meets the minimum requirements of the corresponding algorithm. For example, no less than 128 bits for AES, no less than 256 bits for SHA, etc. Insecure encryption algorithms, such as DES, TDES, and RC4 are not used.

5.2 Hash Algorithm

We use secure hash functions with no less than 256 bits, such as SHA256, and do not use hash functions with existing security risks, such as MD5 or SHA1.

5.3 Data Transmission Security

The wireless auxiliary stream supports TLS connection and can use TLS1.2 or TLS1.3.

5.4 Diagnostic Security

There are no plaintext passwords and user privacy data in the uploaded logs.

5.5 URL Security

It is currently prohibited for the product software of devices to include unpublished public network addresses (including public network IP addresses, public network URL addresses/domain names, and email addresses) which are invisible on the user interface or not described in the product documentation.

6 Security Testing and Release

6.1 Product Launch Process

Yealink follows the Secure Software Development Life Cycle (S-SDLC), emphasizing security in all stages of product requirements, design solutions, and coding specifications. Before any software version is released, it must go through an Alpha version test, a Beta version test, and a UAT test. At each stage, the security team participates in the penetration test of software and hardware, attack surface analysis, and security scanning.

6.2 Firmware Security Testing

Yealink's security team and IT department regularly perform static and dynamic vulnerability scanning and penetration testing on the production environment and the company's internal network environment to ensure that the software development, firmware packaging and device production processes are completed in a secure network environment.

Before the official release of the firmware, the Yealink security team also uses a variety of mainstream anti-virus software to scan the firmware to ensure that the firmware is not infected or embedded with viruses/Trojans.

Penetration tests are conducted to ensure that secure boot functionality, malicious firmware analysis, tampered firmware updates, etc. are all safe. Please refer to the security report in Appendix 1 for details.

6.3 Code Security Specifications

Yealink has strict internal coding security specifications for the R&D team. During every code update, the code is reviewed and a reliability verification is carried out.

Device-related code bases have strict permission management mechanisms and company red line requirements. In order to prevent source code leakage, uploading to public or semi-public services such as Github, Gitee and other public code repositories without permission is prohibited.

7 Privacy Security

Throughout the life cycle of the phone (starting from release from the factory), all user data, including account information, contacts, call records, audio and video call information, etc., are managed directly and locally by the customer in the customer device end. Yealink follows the local privacy protection policy and does not illegally acquire, store, or collect unauthorized customer data by any means.

The user data of the phone device is stored in an encrypted form to reduce the risk of privacy leakage caused by improper user operations.

The network connection and network access generated by the phone device are determined by the company's internal network and user behavior operations. The phone device does not initiate any communication with any IP address or server without the customer's permission.

If the user's device fails and needs to be diagnosed and repaired, Yealink has no way to obtain the current information of the device and therefore requires the cooperation and authorization of the customer to export the phone's diagnostic file locally for troubleshooting.

Disclaimer

This white paper is provided for informational purposes only and does not convey any legal rights to any intellectual property in any Yealink product.You may copy and use this paper for your internal reference purposes only.

YEALINK MAKES NO WARRANTIES, EXPRESS OR IMPLIED OR STATUTORY AS TO THE INFORMATION IN THIS WHITE PAPER.

To learn more about Yealink IP Phone Series devices, visit our website