Security Design of Reolink Open Platform v.1.0

Glossary

Term	Description	Note
Authentication	In computer security, authentication usually refers to the verification of the identity of a user or system to determine whether they are authorized to access a particular resource or perform a certain operation.	-
Encryption	Encryption transforms a piece of plaintext into ciphertext using a certain algorithm, making the text incomprehensible to the unauthorized person.	-
Mutual Authentication	A network security technology that ensures mutual authentication between a device and a server. Not only does the server verify the identity of the device, but the device also verifies the identity of the server.	-

Overview

Objective

This document is for our partners to understand the security design of the Reolink Open Platform.

Applicability

The manual is suitable for people who have a certain level of knowledge of computer security.

Security Principles

- The server is developed using Node.js, and is completely free and open-source for all Reolink partners.
- Reolink will never set any backdoors or steal the private information of users.
- The Core Server should be deployed, operated, and maintained by partners,

and Reolink is not responsible for the deployment, operation, and maintenance of the Core Server.

- After the device is successfully configured with ROP, it will only communicate with the Core Server and will not communicate with any other servers.

About Permission Assignment

This document only involves authentication and does not involve specific permission assignment issues. Partners should design permissions according to their own needs and application scenarios.

Client Security Design

Scope

Client security design refers to the security design between the client and Core Server.

Principle

Reolink is not responsible for the security between the client and Core Server. The solution provided by Reolink is for reference only, and the specific deployment solution should be made by partners according to their different situations.

Ports That Should Not Be Open to the Public Network

The HTTP signal port and the RTSP stream port provided by Core Server to client should not be opened directly for public network access. Otherwise, they are more likely to be attacked.

On the server where Core Server is deployed, these TCP ports should not be open to the public network:

- HTTP port for signals (default value: 61009, can be modified in the configuration file)
- RTSP port for streams (default value: 61008, can be modified in the configuration file)

About Encryption and Authentication

The Core Server is not responsible for encryption and authentication. If encryption and authentication are needed, partners need to deploy components or servers for encryption and authentication by themselves.

The following table includes typical solutions to encryption and authentication for two scenarios:

Scenario	Suggested solution	Note
The client using Core Server is a server built by the partner.	All authentication and encryption are completed on the partner's own server.	The partner's own server forwards requests to the Core Server.
Connecting to the Core Server directly through a website or app on the public network	Partners are strongly recommended to deploy a gateway for encryption and authentication, instead of requesting the Core Server directly.	Nginx is mentioned by Reolink as a reference for the gateway for encryption and authentication. Partners can use Nginx or other components as the gateway.

About HTTP Signal Authentication

Authentication of HTTP signals is completed on the partner's own server, and the Core Server is not responsible for authentication of HTTP signals.

About RTSP Stream Authentication

Authentication of the streaming address is required before playing RTSP streams, and the Core Server will perform the authentication of the streaming address. This is the only authentication the Core Server is responsible for. The streaming address must not be leaked. Otherwise, the attackers who obtain the streaming address will be able to see the live view or recordings.

Device Security Design

Scope

Device security design refers to the security design between the device and Core Server.

Prerequisites

- The server is correctly deployed. After that, it will generate the root CA certificate, the second-level CA certificate issued by the root CA certificate, and the server certificate issued by the second-level CA certificate.
- The device is properly initialized by the partner.

Mutual Authentication Between the Device and Core Server

It can ensure the security of communication between the device and Core Server.

Steps:

- After receiving the command to initialize the device, the device will download the root CA certificate of the server from Core Server and verifies the certificate.
- The device will send an HTTPS request to Core Server with the authorization code sent from the app, in order to obtain the private key certificate of the device.
- The device will use the private key certificate to establish a standard mutually authenticated TLS channel with the server.

Device Stream Security Design

Scope

The device will send streams to Core Server, and the security design protects the connection for streams between the device and Core Server.

Stream Encryption Options

Core Server provides three options for stream encryption: full encryption (TLS), selective encryption (encryptTCP), and non-encryption (TCP).

Encryption Type	Application Scenarios	Note
Full encryption	Applicable in all scenarios	The default option, standard TLS encryption
Selective encryption	Applicable in most scenarios, difficult for eavesdroppers to recover the streams	Recommended, better device performance than using full encryption
Non-encryption	Not recommended unless the device is on the same LAN as Core Server	-

Alarm Notification Security Design

Scope

Security design for pushing alarm notifications from Core Server to the partner's server.

Prerequisites

- The partners must have their own servers that can receive HTTPS POST requests from Core Server.
- The partners have set the ROP_ALARM_WEBHOOK parameter when deploying Core Server,
- The partners have enabled the alarm push notification function for some or all devices. The function is disabled by default.

Encryption and Authentication

- About encryption: It is strongly recommended to use HTTPS callback addresses instead of the HTTP ones.
- About authentication: The address provided by the partner should be unique and unknown to others. It is best to include authentication parameters in the QueryString of the address, making it difficult to attack.