LiveEngage Messaging Platform: Security Overview

Document Version: 2.0
July 2017

LIVEPERSON

Contents

Introduction 3
Supported Platforms 3
Protecting Data in Transit 3
Protecting Data at Rest 3
Encryption 3
Masking 3
International Security Compliance Program 4
Secure Development Life-Cycle 4
Security by Design 4
Static Code Analysis 4
Input Validation 4
Ethical Hacking, Vulnerability Assessments and Penetration Testing 4
System Components 5
Data Flow 5
Authenticated Interaction 5
Unauthenticated Interaction 6
Push Notifications 7
Agent Login 8
Standard Login 8
SSO 8

Introduction

LivePerson has developed an advanced, SDK-based solution for mobile messaging between Consumers and Brands.

As a leading provider with thousands of customers and many years of experience, we understand that the integration of any 3rd party SDKs into a brand's mobile application requires an appropriate risk assessment and due-diligence processes. We also realize that the content of Consumer and Brand interaction should be treated in accordance with the highest levels of Security and Privacy.

We've invested significant efforts in designing and implementing a robust security model to help protect our messaging solution. This document outlines the high-level security model and controls that has been implemented in LivePerson's messaging platform and SDK.

Supported Platforms

LivePerson's SDK applies to both iOS and Android devices.

Protecting Data in Transit

All communication between the LivePerson SDK and the LivePerson backend is encrypted using 256bit AES with 2048 RSA, and established over either HTTPS (for REST communication) or WSS (for data transmitted over WebSockets). In addition, requests are verified with a unique JSON Web Token (JWT).

Protecting Data at Rest

Encryption

On the Device → In both iOS and Android, data generated by the LivePerson SDK and stored on the consumer's mobile device is encrypted based on standard OS ciphers. The encryption is based on 256bit AES.

On the LivePerson Datacenter → An optional 192bit AES encryption is available using unique and dedicated set of keys for each Brand. LivePerson recommends enabling encryption for data at rest as part of the best practices for secure messaging.

Masking

LivePerson provides two primary, optional data masking functions:

RegEx based, real-time masking via the SDK → the masked data will not bestored on the device or on the LivePerson servers.

RegEx based, server-side masking → the masked data is displayed to the customer care professional during the active interaction, but will is not stored on the LivePerson servers.

International Security Compliance Program

Similar to LiveEngage, the Messaging platform is in-scope of the LivePerson International Security Compliance Program and adhere to the following standards: ISO27001, SSAE16 SOC2, PCI-DSS via Secure Form widget and EU Privacy directive, in progress to comply with GDPR prior to May 2018.

Secure Development Life-Cycle

Security by Design

Security is an integral part of the software development processes at LivePerson. The platform and all of its components have gone through constant security design reviews by the LivePerson Security team and R&D Architects. Additionally, LivePerson Mobile Application Developers has gone through a dedicated Secure Mobile Application Development training by a leading 3rd party instructor that specializes in this domain.

Static Code Analysis

The code behind the platform has undergone repeated static code analysis scans in order to help identify potential gaps/flaws. According to LivePerson's SDLC policy, any High Risk findings are fixed prior to deployment to Production.

Input Validation

All data exchanged between LivePerson backend and the Agent Workspace is validated on the server side to prevent browser side attacks (for example Cross Site Scripting - XSS).

Ethical Hacking, Vulnerability Assessments and Penetration Testing

The LivePerson Unified Messaging Platform, API and SDK have been tested multiple times by independent penetration testers and Ethical Hackers with specialization in Mobile Application Security. An additional weekly vulnerability assessment is executed against the infrastructure using Rapid7 Nexpose scanner.

System Components

Component	Description
Client Side	
Mobile App	Developed by the brand (not LivePerson).
LP SDK (iOS & Android)	To be embedded in the brand's mobile app.
LiveEngage Workspace	Browser-based web application from which agents interact with users.
Server Side	
UMS	LP Unified Messaging System which is responsible for asynchronous messaging.
REST API Backend	Channel for the SSO process.
WebSocket Backend	Channel for messaging between the consumer and the agent.
IDP (LP Proprietary Identity Provider)	Responsible for the SSO login process between the LivePerson environment and the customer environment.
AC Connector	Stores customer details such as URL, secret, public certificate, and more.

Data Flow

The LivePerson Messaging SDK has three primary types of interactions and dataflows.

1. Authenticated Interaction
2. Unauthenticated Interaction
3. Push Notifications

Below is a high-level description of all three dataflows and sequence diagrams.

Authenticated Interaction

There are 2 methods for establishing an Authenticated Interaction:

- Client Based Implicit Flow: a JWT is created by the brand upon Consumer authentication and communicated to LivePerson by the Consumer. A step by step process is outlined below:

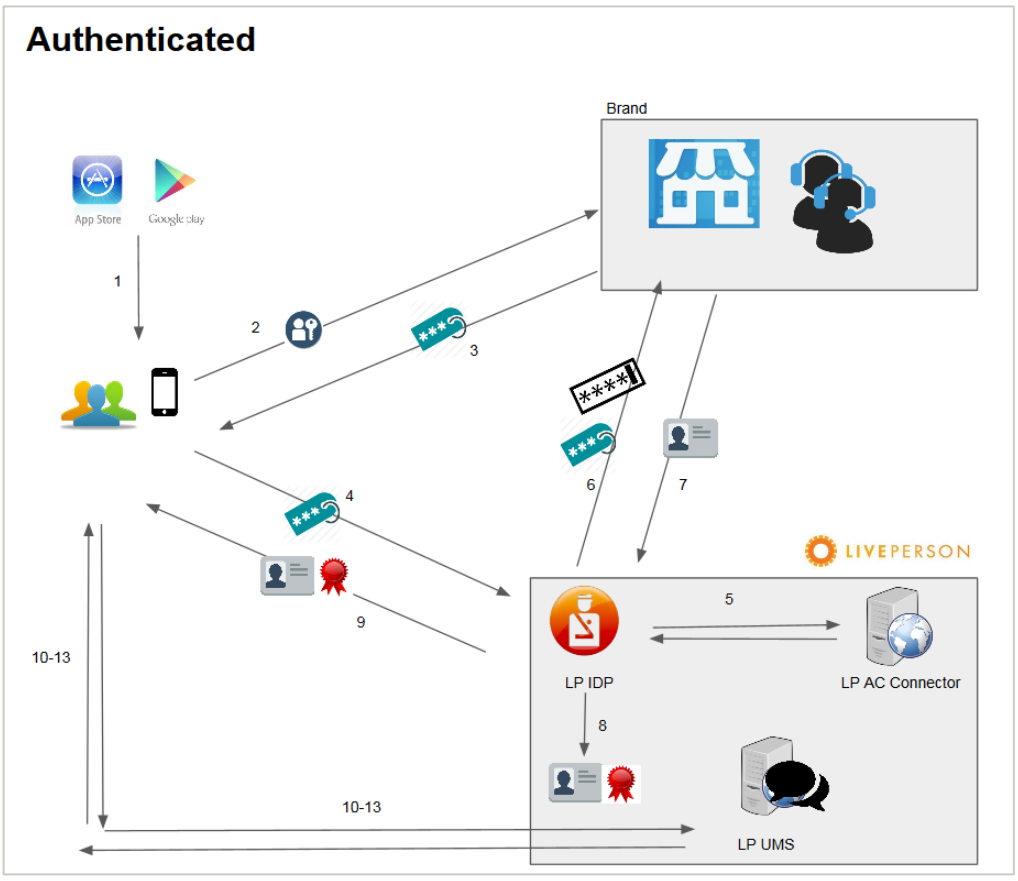
    1. Consumers authenticate directly to the Brand Mobile Application with their personal credentials (username/password/certificate).
    2. The Brand generates a unique consumer ID by a JSON Web Token (JWT).
    3. The JWT is communicated to LivePerson through the SDK (without exposing username/password).
    4. LivePerson generates an additional unique JSON Web Token (JWT) which is used in each message that is sent between the Consumer and LivePerson backend infrastructure.

Encryption of the JWT is optional by using JWE.

- Server Based Code Flow: An AuthCode (UID) is generated by the Consumer upon authentication. The AuthCode is communicated to LivePerson, which then directly communicates with the Brand Servers to obtain the JWT based on that unique ID. The JWT is signed with the Brand Public Key. A step by step process is outlined below:

    1. Consumers authenticate directly to the Brand Mobile Application with their personal credentials (username/password/certificate).
    2. The Brand App / LivePerson SDK generates a Unique AuthCode.
    3. The AuthCode is securely communicated to LivePerson through the SDK.
    4. LivePerson sends the AuthCode to the Brand Server over encrypted channel.
    5. The brand generates a JWT which is communicated to LivePerson over encrypted channel.
    6. The JWT is used in each message that is sent between the Consumer and LivePerson backend infrastructure.

Encryption of the JWT is optional by using JWE.

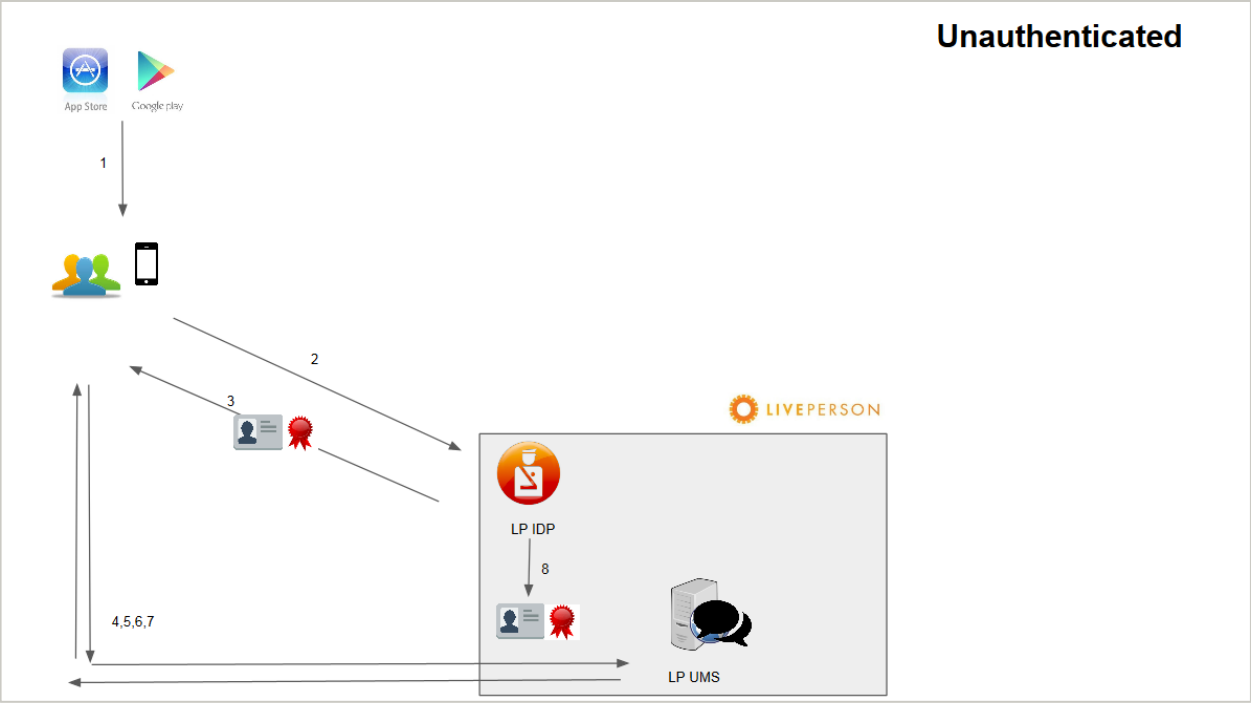


Unauthenticated Interaction

The consumer does not authenticate to the brand application or systems. However, the consumer is still able to initiate a messaging request and communicate with a brand agent.

The process for unauthenticated interactions is described below.

1. The consumer downloads the brand mobile application from the App Store or Marketplace.
2. The consumer initiates a request for an instant messaging conversation.
3. LivePerson IDP receives the request, generates a random UUID for the consumer, creates a LivePerson JWT, and sends it back to the consumer via the LivePerson SDK.
4. The consumer establishes a secure websocket to LivePerson UMS.
5. Each message or communication to UMS is sent with LivePerson JWT.
6. UMS verifies LivePerson JWT validity and expiration.
7. If LivePerson JWT is expired or not verified, the socket is terminated.

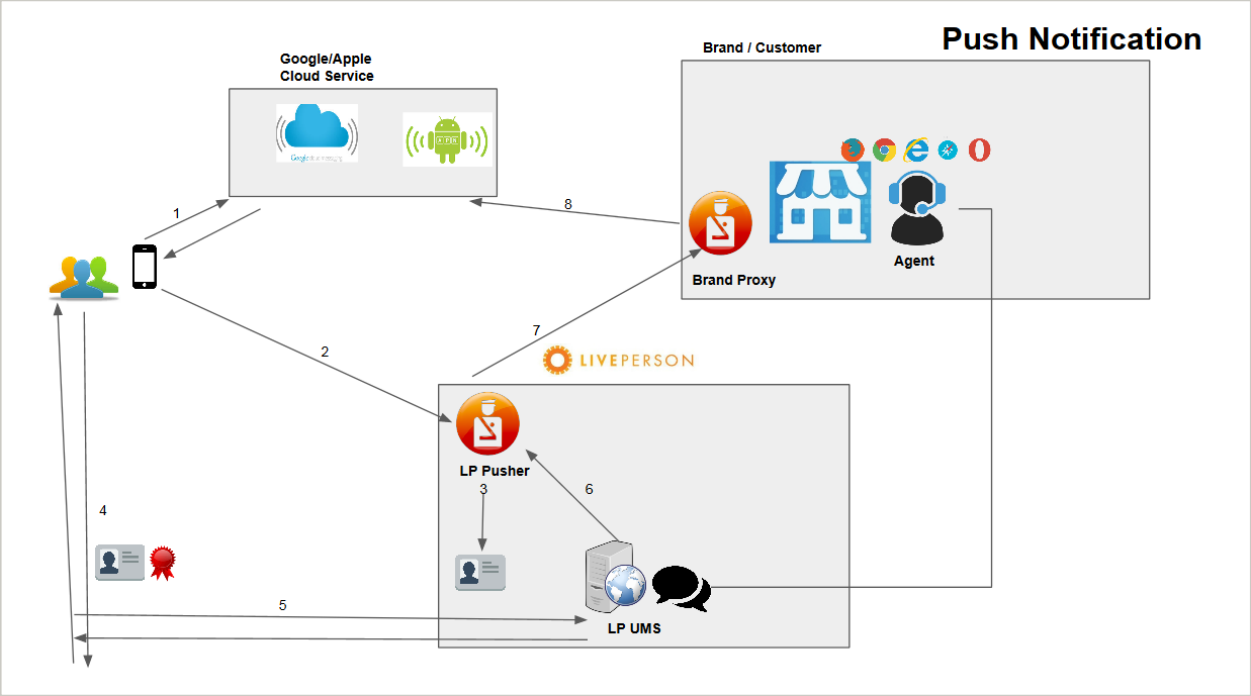


Push Notifications

Push notifications are designed to allow an agent to communicate with or respond to consumer inquiries when the consumer is not using the brand application.

The process for push notifications is described below.

1. The LivePerson SDK generates a request for a new token (UID) from the Apple Push Notification Service (APN), or the Google Cloud Messaging (GCM) service.
2. The LivePerson SDK registers (userID + APN/GCM token) to LivePerson Pusher.
3. LivePerson Pusher adds a new record for the new consumer token.
4. Any message notification is sent to LivePerson's UMS with an LivePerson JWT.
5. All messages are verified by the LivePerson Gatekeeper.
6. When UMS generates a new message to the consumer, a new notification is generated to LivePerson Pusher.
7. LivePerson Pusher sends the message with APN/GCM token to Brand Proxy (Customer Pusher).
8. Brand Pusher sends a push notification via GCM/APN to the consumer device.
9. LivePerson UMS sends the message to the agent queue, to be reviewed by the brand agent.



Agent Login

Access to the LiveEngage interface requires authentication.
LivePerson provides two options for agent authentication:

- Standard login
- SSO

Standard Login

Agents authenticate using a unique siteID, Username and Password. Brands are responsible for the User Management and Login Policy settings of the account. The default Login Policy requires a minimum password length of 8 characters. Brands may opt to change the password policy, add IP based access lists, and implement additional security settings.

SSO

Brands may choose to implement and enforce a SAML 2.0 based Single Sign-On.
If the SSO feature is enabled, the agents authenticate to the brand authentication platform. Upon successful authentication by the brand, a token (bearer) is securely provided to LiveEngage, and the agent is logged in.