

# ESET SECURE AUTHENTICATION

PHP SDK Getting Started Guide

## ESET SECURE AUTHENTICATION

**Copyright © 2016 by ESET, spol. s r.o.**

ESET Secure Authentication was developed by ESET, spol. s r.o.

For more information visit [www.eset.com](http://www.eset.com).

All rights reserved. No part of this documentation may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning, or otherwise without permission in writing from the author.

ESET, spol. s r.o. reserves the right to change any of the described application software without prior notice.

Customer Care: [www.eset.com/support](http://www.eset.com/support)

REV. 6/14/2016

# Contents

- 1. Introduction .....4
- 2. SDK Overview .....4
- 3. SDK License Activation .....5
- 4. SDK in practice .....5
  - 4.1 Using the SDK .....5
  - 4.2 SDK System Integration .....6
    - 4.2.1 Database requirements .....6
    - 4.2.2 Reading and Writing 2FA Data .....6
    - 4.2.3 Update Login UI With 2FA Methods .....7
    - 4.2.4 Update the Management UI to Enable/Disable 2FA For Users .....9
  - 4.3 Additional Components .....9

# 1. Introduction

This document details what is required to add a second authentication factor (2FA) to an existing application by using the ESET Secure Authentication SDK.

# 2. SDK Overview

The ESET Secure Authentication SDK is used to add an additional authentication factor to any application. The SDK integrates with existing systems via a client-side library. *Figure 1* depicts an authentication platform before SDK integration.

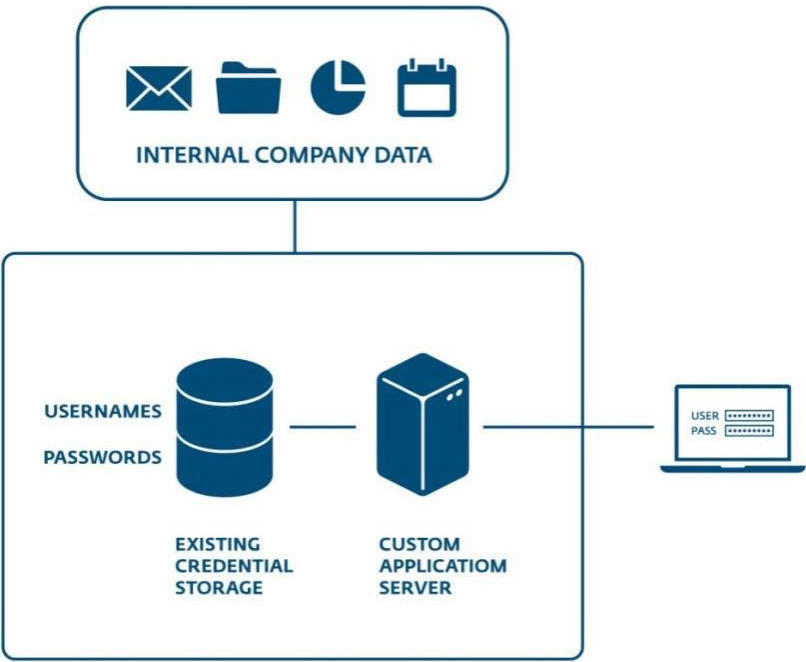


Figure 1: Before ESET Secure Authentication SDK integration

Figure 2 below depicts an authentication platform after SDK integration.

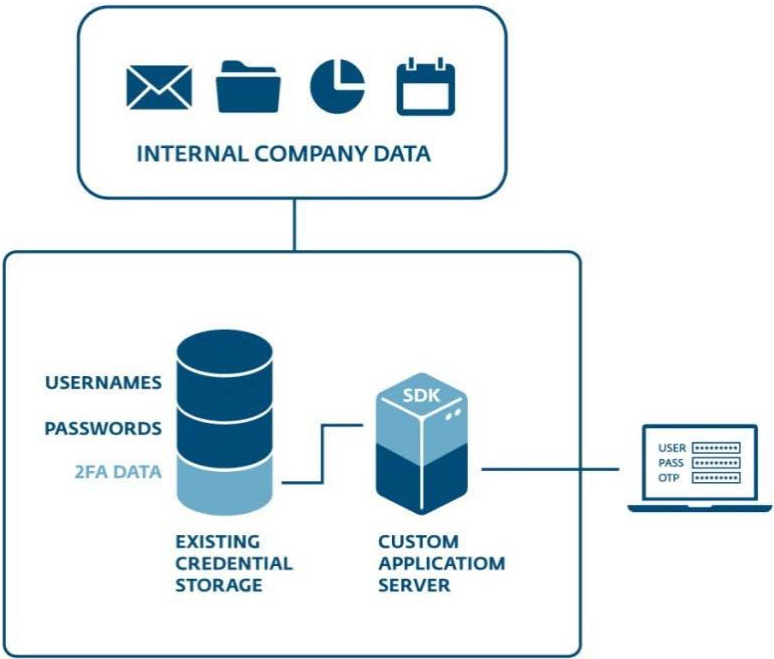


Figure 2: After ESET Secure Authentication SDK integration

The remainder of this document provides instructions for obtaining the SDK, using the example code and finally integrating the SDK into your existing platform.

### 3. SDK License Activation

To use the ESET Secure Authentication SDK, you require an API Key and a Shared Secret. These credentials are derived from your ESET Secure Authentication license.

To derive your SDK credentials, navigate to <https://esa.eset.com/sdk>. Log in using the ESET Secure Authentication license information you used to access the SDK download.

To activate your license for the SDK, tick the checkbox next to **I accept terms of End User License Agreement**, then click **Get API credentials**. Afterward your **API key** and **API secret** will be shown.

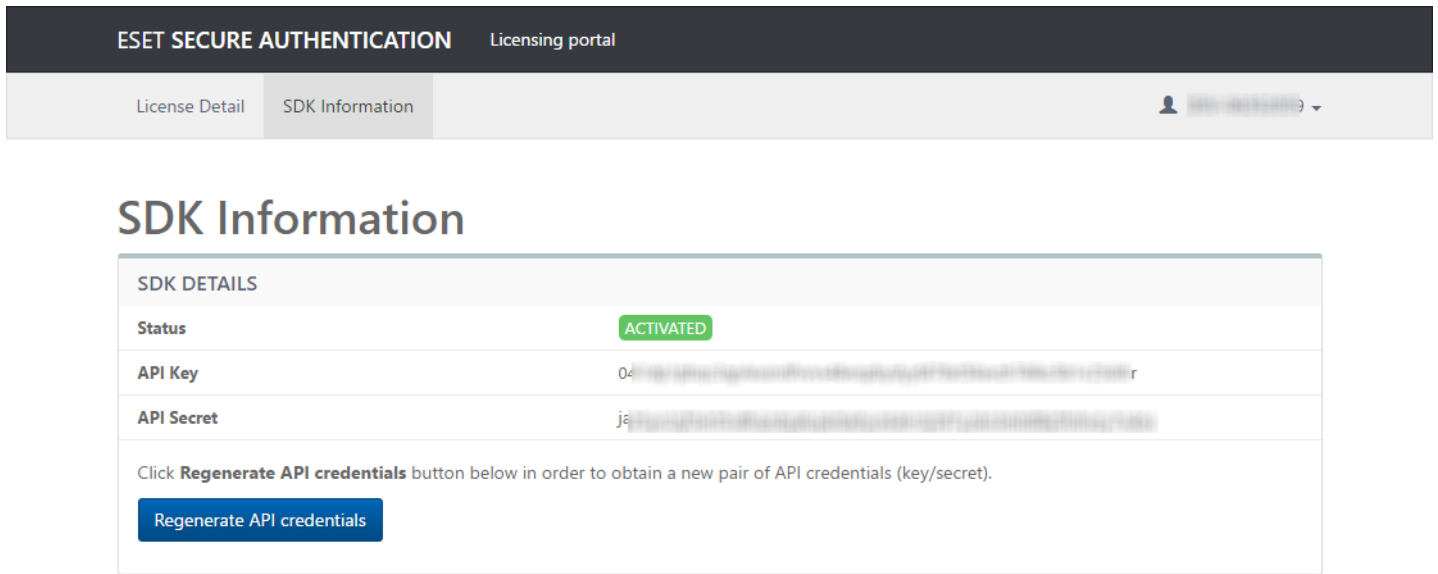


Figure 3: License activated

### 4. SDK in practice

The quickest way to get up and running with the SDK is to run the sample code.

Download the sample application .zip file from <https://esa.eset.com/sdk/docs/samples/>

The zip file contains a **README.txt** – follow the instructions therein.

#### 4.1 Using the SDK

Users of the SDK can consult the ESET Secure Authentication SDK Developer Guide for instructions on integration with their platform.

The SDK Developer Guide is available here: <https://esa.eset.com/sdk/docs/>

This guide contains:

- How to reference the SDK library in an application
- An overview of how to use the SDK
- Detailed reference documentation for namespaces and classes

## 4.2 SDK System Integration

Once the SDK has been evaluated, it must be integrated with your existing authentication system.

The following steps are required to integrate the ESET Secure Authentication SDK into a system:

- Extend the user storage database with extra 2FA fields
- Implement classes for reading and writing 2FA data for users
- Update existing login UI to accept OTPs
- Update existing user management UI to manage a user's 2FA settings
- Implement optional components

### 4.2.1 Database requirements

The ESET Secure Authentication SDK stores a user's 2FA data in your existing database as a string. You will therefore need to add a column capable of storing unicode strings of varying length.

MySQL users	We recommend that you use the <code>TEXT</code> datatype
Postgres users	We recommend that you use the <code>character_data</code> datatype
Oracle users	We recommend that you use the <code>NCLOB</code> datatype
Microsoft SQL Server users	We recommend that you use the <code>nvarchar(max)</code> datatype

If you do not have a mobile telephone number field for each user, it is highly recommended that you create a field capable of storing mobile numbers (numeric strings of varying length). This will help to ensure compatibility with future releases.

### 4.2.2 Reading and Writing 2FA Data

The ESET Secure Authentication SDK exposes the `IUserStorage` interface that transports data between the SDK and your database; this interface must be implemented to read and write 2FA data (see the SDK Developer Guide for details).

The interface uses two methods that must be implemented, `loadUser` and `saveUser`.

#### **loadUser**

This method uses the following input parameter:

- string username - the user whose 2FA data will be retrieved

This method has the following return type:

- string data - the 2FA data for the user

In other words, the supplied username must be used to return the 2FA data for that user.

#### **saveUser**

This method has the following input parameters:

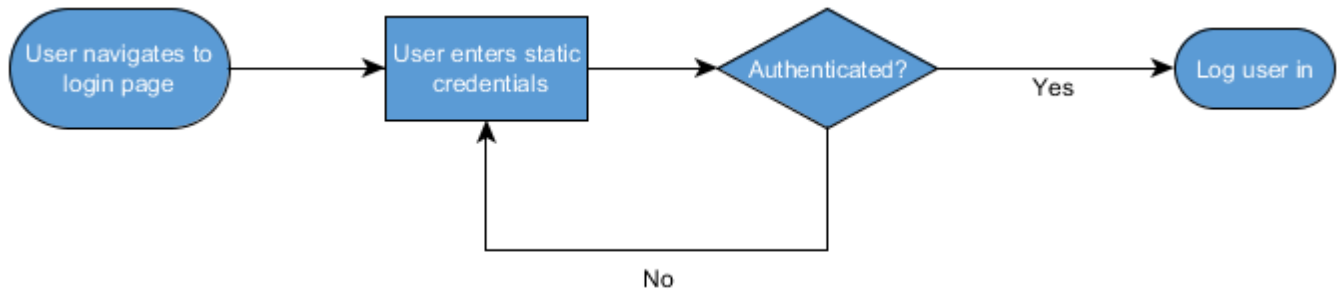
- string username - the user whose 2FA data you want to store
- string data - the 2FA data to store

This method has no return value. In other words, the SDK will provide the username and 2FA data – make sure that this data is written to the 2FA field in your user database.

### 4.2.3 Update Login UI With 2FA Methods

This section describes the authentication logic that must be implemented for 2FA users.

Once a user has authenticated their static password against the existing system, the pre-authenticate method must be called for that user. This method checks the 2FA type for the user (SMS, Mobile App, etc) and sends an SMS OTP if required. It returns a result that contains the expected credential type, which must be used to guide the user during the logon process. Details of this type may be found in the [Developer Guide](#).



**Figure 4: Single factor authentication logic (before integration)**

An example of authentication logic after the ESET Secure Authentication SDK has been integrated is depicted in *Figure 5*.

Note that the logic for this will vary from system to system depending on the requirements.

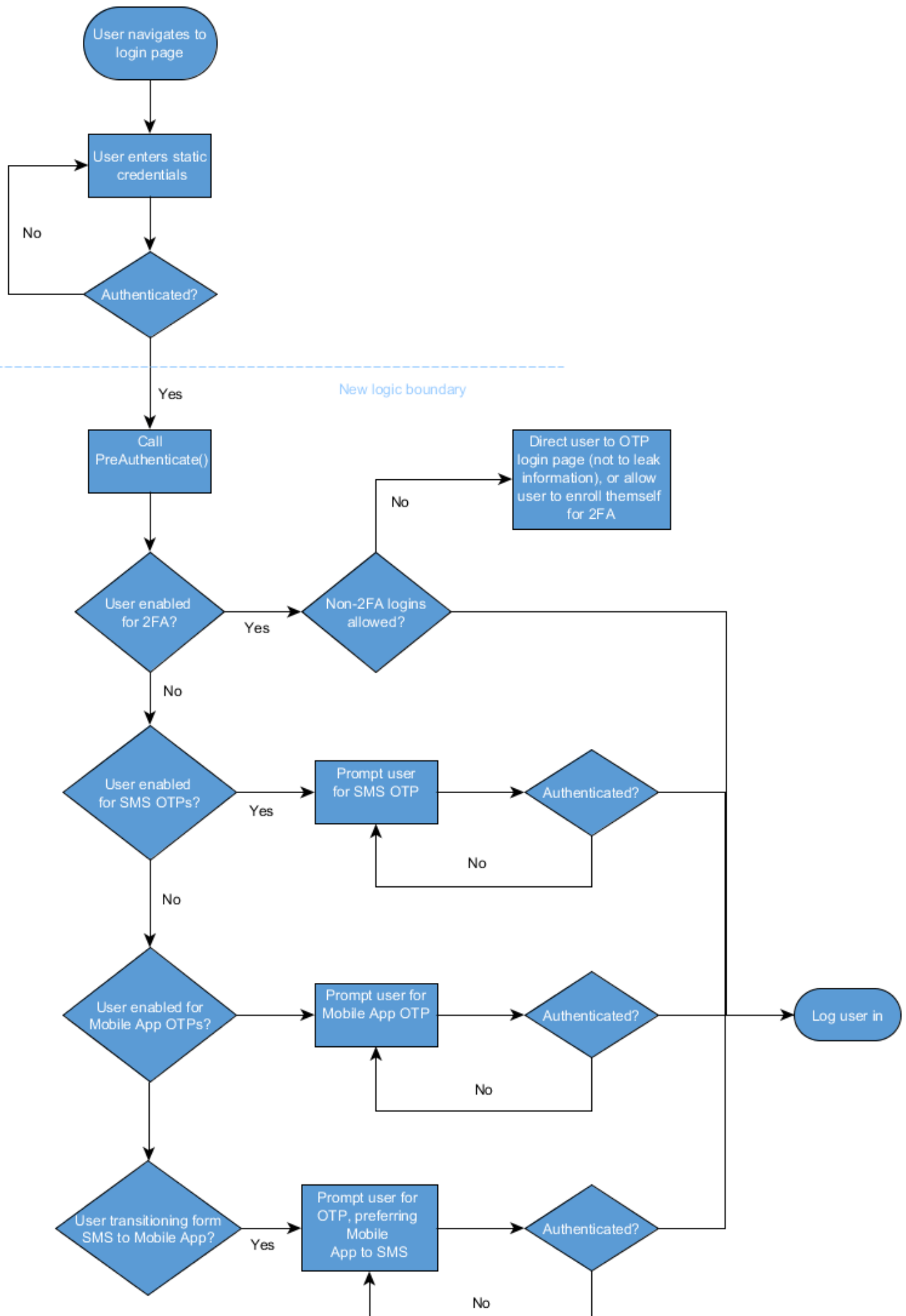


Figure 5: Two factor authentication logic with ESA SDK



#### 4.2.4 Update the Management UI to Enable/Disable 2FA For Users

User management is performed via the `TwoFactorUser` class. Users can be enabled for 2FA using mobile app OTPs or SMS OTPs. The transitioning state allows a user to upgrade from SMS to Mobile App OTPs. An administrator may perform user management, or users may enroll themselves.

The `TwoFactorUser` class exposes methods (actions) that may be performed on users.

### 4.3 Additional Components

The following components are optional.

#### Logging Integration (Recommended)

The ESET Secure Authentication SDK logs non-critical events via a logging wrapper, so as not to limit you to any logging framework. This means it is easy to use your existing logging framework. See the [Developer Guide](#) for further details.

#### Auditing Integration (Recommended)

The SDK audits various critical events via the `IAuditor` interface. If this interface is not implemented, auditing events are sent to the configured logger. The implementation is set with the `setAuditor` method of the `TwoFactorConfiguration` class.

#### Using an Alternative SMS Gateway (Optional)

The ESET Secure Authentication SDK sends SMS messages via the global ESET SMS gateway. You can configure your own SMS Gateway by implementing the `ITextMessageSender` interface