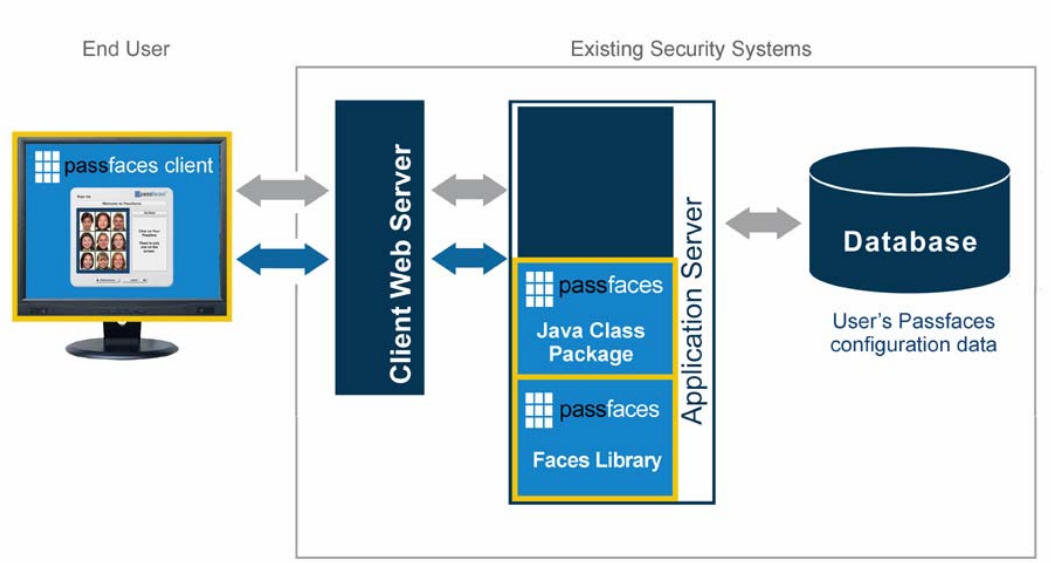
Passfaces SDK Overview

Toolkit Version 3.0
Document revision 3.0
March 2006

Introduction

This document describes the Passfaces SDK – a collection of software components and example HTML pages that provide an easy and flexible path to providing user authentication using Passfaces within your own web based application. The Software Developers' Kit (SDK) consists of the following primary components:

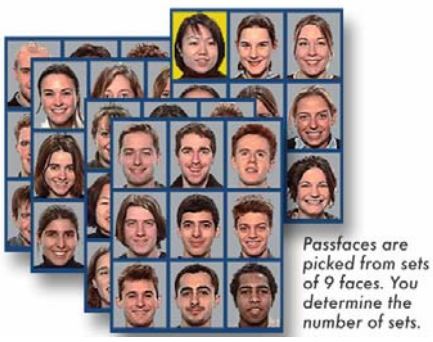
- A server-side Java class package;
- Passface Client (user interface);
- Passfaces Library (database of face images).
- Detailed integration information



Although Passfaces can run on dedicated hardware, it is designed to run on existing Windows and UNIX based security systems so no additional hardware is required. The reference implementation provided is a simple JSP application that will run on Java servlet capable application servers. The Passfaces Library is a database of serialized .JPG files from which a user's Passface authentication set is assigned. The User Interface, or *Passfaces Client,* is offered as a highly parameterized JavaScript, Active X component and Java applet that represent the core of the Passfaces system.

Functional Flow

Passfaces is based on the human brain's ability to recognize individual faces. The simplest way to think of Passfaces is that faces are used instead of alphanumeric characters as an access code. As with passwords, new users are asked to sign-up for Passfaces log on access. Users are assigned a set of Passfaces typically consisting of 5 separate images. The number of images assigned is determined in advance by the system administrator and ranges between 3 and 7.



Once Passfaces are assigned, users are taken through an optimized enrollment process designed to familiarize them with their Passfaces. This one-time process takes about 3 minutes; it can be thought of as placing a Passfaces cookie in the user's brain. Once the exercise is complete, users can use their Passfaces to log on anytime, anywhere.

During the logon process, users are asked to select their assigned Passfaces from 3 by 3 grids each containing one Passface and 8 decoys of the same general appearance. The faces appear in random positions within the grid each time. This process is repeated until all of the assigned Passfaces are identified. Once correctly identified the user is logged on.

Operational Flow

Passfaces and decoys for each registered user are stored in a database. During logon, all associated images, decoys and Passfaces, are sent to the end user and presented through the user's web browser. Images are presented one grid at a time.

Once the user has clicked on one face from every grid, the Passfaces Client sends a one time positional reference to the server as an HTTP form. This result is an ASCII string (two characters for each Passface). The result may be used in place of a password. It may also be compressed or passed through a one-way function using existing databases or directory services.

Face reference numbers of the Passfaces and 'decoy' faces are stored at the application server. For example, if a user's account is configured to require 5 Passfaces, 45 face references are recorded and stored. The list of face references is presented to the Passfaces Client as a configuration parameter during the logon session. The Passfaces Client then reads and displays the corresponding JPEG face images within the user's browser.

As with passwords, the messages from client to server contain the user's authentication secret and should be encrypted to prevent interception. The user must be sure they are authenticating to a trusted party and not an impostor. Using SSL achieves both of these requirements without the need for any additional cryptography. It is reassuring to note that the authentication process cannot commence without the presentation of user-specific configuration data. The presentation of this specific information further authenticates the site without requiring any additional action from the end user.

Software Developers' Kit

The Passfaces SDK contains the software and information required to integrate Passfaces into an existing authentication framework. Using an SDK approach gives organizations the ability to configure Passfaces to fit the current operating environment, limiting or eliminating the need to make any substantial changes to the existing system.

The following items are included in the Passfaces SDK.

- **Passfaces Client** – Passfaces web browser user interface (Active X, Java, or JavaScript) is used to set control parameters and facilitate interaction with the end user.
- **Passfaces Image Database** - Collection of face images.
- **Reference Implementations** – Examples and demonstration code used for integrating Passfaces into an existing security application.
- **HTML Files** – These files demonstrate how to replace passwords in a web based environment.

Passfaces Client

The Passfaces Client runs in the user's browser and provides the user interface for enrolling and logging on. The client can run using Active X, Java, or JavaScript. The JavaScript version will run in any modern browser. The Active X version will run in Active X enabled browsers. The Java applet version will run in any browser (including older browsers) as long as a Java Virtual Machine is available. The most obvious advantage of the Java and JavaScript version is that anyone can run them regardless of hardware platform, operating system or browser. The advantages of Active X are the speed of launching the application as well as more controlled caching of the Active X component and of the face images. It is recommended that your application determines the appropriate client to use based on the user's browser version and operating system combination. Example code is provided to achieve this.

Customizable Skins

Operational details such as the number of Passfaces per user and how they are assigned are configurable within the Client. The look and layout of the Client is customizable through a set of parameters contained in the Client's HTML page.

Passfaces offers two options for skin customization. The existing skins can be modified by changing a few graphic images to include a company logo and colors. Because the client is HTML, you can create a complete customized Web page that reflects the look and feel of your Web site. Changes to the page are made using a combination of Passfaces control parameters located on the Client page and standard HTML formatting. Among other things, control parameters determine Passfaces positioning on the page, text messages within the log on and signup pages, the look and behavior of the buttons, and font characteristics. All of the Passfaces client options (ActiveX, Java, and JavaScript) support the same set of configuration parameters.

Reference Implementation

A reference implementation using Java servlet technology is provided to demonstrate the server-side components of a complete Passfaces enrollment and authentication system. The salient features translate easily to any server-side development environment. New to version 3.0 are Java classes that implement the recommended feature set, including phishing/pharming countermeasures. These include the randomized selection of Passfaces grids per user and the ability to present only a random subset of the user's Passfaces during a given authentication session. For more detail on strategies to implement these features, see the white paper Implementation Security Strategies included with the SDK.

The code may also be used as the basis of a simple web site authentication system. For simplicity, the example implementation stores the user enrollment data as ASCII text files within a simple file system based 'database'. A real implementation would most likely use an existing user database or directory.

Face Database

A collection of face images is provided as part of the Passfaces SDK. These images are in JPEG format and have unique filenames that are used to identify the face image. These faces have been pre-grouped into grids to assure similarity of general appearance.

An optional server-side component, the Passfaces Face Server, is provided as a source code distribution or built executable. This is optimized to reduce the time taken to download the face images. When configured to use the Face Server, the Passfaces Client requests a number of face images (typically 9 – a complete grid) to be encoded and delivered by the Face Server as one JPEG image strip. This reduces the download time by significantly reducing the number of image requests made by the user's browser. *Example: 5 Passfaces require 45 small images, approximately 2K bytes each. Using the Face Server reduces this download to 5 larger images, approximately 15K bytes each.*

Summary

Because Passfaces leverages existing security architectures and hardware, implementation can be accomplished in days as opposed to months. The flexible nature of the software gives you the ability to create a customized authentication solution. Detailed information on strategies for implementation can be found in the Implementation Security Strategies paper and detailed Passfaces integration assistance is available in the Client Configuration Guide and in the Java Class Documentation.