
Blue Button Authentication Field Test

Release 1.0

Ganesh Kini

April 26, 2013

Abstract

This paper describes the Blue Button Authentication Field Test prototype software developed to provide online identification and authentication of Veterans. The application utilized a third party identity and authentication product.

Contents

1	Introduction.....	1
2	Code Walkthrough	2
2.1	OSEHRAuth.Website	2
2.2	OSEHRAuth.Core	2
2.3	OSEHRAuth.Extension	2
2.4	OSEHRAuth.UnitTests	2
2.5	Log4net.....	2
2.6	SOAPValidator	3
3	How To Use The Code.....	3
4	Conclusion.....	3

1 Introduction

This system allows users, without a DS Logon, to prove their identity through an online means. A secure website was developed that communicates with a third party identity and authentication product, and MDWS. The user accesses the website directly and completes the verification and authentication steps. Upon successful completion, the existence of the user's record in the Corporate Data Warehouse (CDW) is checked through a Medical Domain Web Service (MDWS) call. If the user authenticates successfully and his record exists in the CDW, the authentication of that user is logged as successful.

The source code associated with this system does not include the proprietary code or APIs of the third party identity and authentication product.

The solution includes the following modules, which are explained in Section 2.

- Authentication Website
- Authentication Core Module, includes the interactions between the third party authentication provider and Medical Domain Web Service (MDWS) web services
- Extensions Module

2 Code Walkthrough

The solution was developed using C#, Asp.net 4.5, MVC4, WCF (Windows Communication Foundation), Visual Studio 2012 and consists of projects described below.

2.1 OSEHRAuth.Website

This is a C#, Asp.net 4.5, MVC4 web application project. The project consists of 2 controllers (Home Controller, Person Controller), 4 main views: Getting started (index.cshtml), About You (Verify.cshtml), Quiz (Authenticate.cshtml) and Results of the Quiz (Authenticated.cshtml). The Person Controller calls functions in the Core Module (described below) to verify a person's record in the third party public records' database and to obtain a quiz to authenticate the person. The quiz answers are passed to the Core module to get the score. If the score result is "PASS", then Success View is displayed to the user. If the person passes the authentication quiz and his record exists in the CDW, then the person is considered to have authenticated successfully.

2.2 OSEHRAuth.Core

This is a C# 4 library project. The project has 2 main classes: IdentityService class and MdwsMhvService class. The IdentityService class has both 'verify' and 'authentication' functions. The Verify function is responsible for presenting an individual's demographic details to an external web service to identify a person's public record. The Authenticate function presents the quiz answers and receives the score. The newly added cdwLookupSlim function of MDWS is utilized by the MdwsMhvService to confirm the existence of an individual's record in the CDW.

Proprietary parameters of the third party authentication provider have been removed from this code. Before the code is functional, access to a third party authentication provider API is needed and the function parameters, as required by that provider, need to be added.

2.3 OSEHRAuth.Extension

This project has client inspector class to intercept the soap message and add custom security headers needed by the third party public records database web service. This class is configured in web.xml file of the website project.

2.4 OSEHRAuth.UnitTests

This is a unit test project to write unit tests.

2.5 Log4net

This is an open source logging framework, the source was included as it needed some custom modification to work in windows 7.

2.6 SOAPValidator

This is another open source project that is included, as a utility to validate the soap message to the wsdl schema.

3 How to Use the Code

Visual Studio 2012 and Asp.net MVC4 need to be installed to use the code. The entire source code is in the OSEHRAuth directory. The solution file OSEHRAuth.sln is in OSEHRAuth directory. The solution file can be opened in Visual Studio 2012 and can be compiled, deployed and run from Visual Studio 2012 IDE (Interactive Development Environment).

The functions in the Core module have been modified to remove proprietary function signatures of the authentication service provider. The code compiles but will not run in the absence of the third party web service. Web.xml file needs to be configured with the endpoint url and custom extensions as needed by the third party authentication provider's web service. The code can be used in other similar projects with suitable modifications for the third party public records web service calls.

4 Conclusion

The code was built to solve the challenge of in-person identity authentication, utilizing a web interface between an identity authentication service and the Veteran's Health Administration (VHA) to validate user information. This type of solution can be utilized where data capture and transmission are required over secure channels.