Single Sign-on Integration

CONTENTS

OVERVIEW

CLUBHOUSE ONLINE E3

This document explains how to implement single sign-on to ClubHouseOnline E3 (CHO E3).

CHO E3 uses a digital signature methodology in order to validate the source of requests for single sign-on. Each 3rd party vendor that has been validated will generate a pair of public / private keys (Asymmetric Key Encryption) and provide CHO E3 support with a copy of their public key. The vendor will then be provided with a personal identification number. All requests from the vendor will include their personal identification number and a digital signature, signed with their private key. Based on this information, CHO E3 will trust sign-on requests from that vendor and automatically log into CHO E3 for the members provided.

The first step is for the club to contact CHO E3 support and ensure that they are aware of whom the 3rd party vendor is and that the single sign-on module has been enabled for the club. If CHO E3 support has not been notified by the club – they will not directly work with a 3rd party vendor.

GENERATING KEYS

In order to utilize single sign-on to CHO E3 you need to create a pair of public/private keys (Asymmetric Key Encryption). The private key will be used to create a digital signature for the information that is passed to CHO E3. If the information in the signature does not exactly match the data passed then the sign-on will fail.

The private key will be used for generating your digital signature whenever you want to sign-on to CHO E3. The private key should be kept securely and not shared with anyone – including the club or CHO E3. If you lose your private key or feel that it has become compromised – please generate a new private key and send the corresponding public key to CHO E3 support.

To generate a pair of public/private keys, we recommend using OpenSSL command line tools. It is available for Windows and Linux from www.openssl.org.

To generate an RSA (invented by Rivest, Shamir and Adlemen) key pair:

openssl genrsa -out myrsakey.pem 1024
openssl rsa -in myrsakey.pem -pubout -out myrsapub.pem

For Microsoft .Net, compile and run the following program to generate the keys:

```
using System;
using System.IO;
using System.Security.Cryptography;
using System.Text;

public class KeyPairGen
{
    public static void Main(String[] args)
    {
        RSACryptoServiceProvider rsaProvider = new RSACryptoServiceProvider(1024);

        FileStream fStream = new FileStream("rsapublic.xml", FileMode.Create);
        StreamWriter sWriter = new StreamWriter(fStream);

        sWriter.Write(rsaProvider.ToXmlString(false));
        sWriter.Close();

        fStream = new FileStream("rsaprivate.xml", FileMode.Create);
        sWriter = new StreamWriter(fStream);
        sWriter.Write(rsaProvider.ToXmlString(true));
        sWriter.Close();
    }
}
```

THE LINK

In order to single sign-on to CHO E3, you need to use the following link:

[Domain]/passthrough.aspx?time=%time%&vendor=%vendor%&userid=%userid%&page=%page%&value=%value%

where:

Domain – The website for the club (typically www.clubname.com)
%time%[1] - a timestamp in milliseconds since January 1st 1970 00:00 UTC
%vendor% - This is your 10 digit code provided by CHO E3 Support
%userid% - The member number of the user who will be logging in.
%page%[2] - (optional) – this optional parameter allows you to specify the landing page after a successful sign-in
%value%[3] - a digital signature of the first four parameters concatenated together with a pipe symbol «|».

---

1. In order to prevent replay attacks, the vendor's servers and the CHO E3 servers must have their clocks in sync. Currently the time window is **90** seconds. Single sign-on requests with a timestamp outside of this window will be rejected. Please synchronize your servers with a standard internet service. Note that we require the difference in milliseconds between your current time and January 1st, 1970 00:00 UTC (you may need to use a UTC specific method in your code).
2. The page parameter can have the domain name or be a relative path within the site directly. If a relative path is used – the URL should start with /
3. The digital signature must be exactly of the form: **time|vendor|userid|page**
   It cannot vary in terms of order or have extra values or spaces. This will cause the sign-on to fail.

EXAMPLES

Note: In examples the variables Vendor, UserId, and DomainName should be defined before using the code.

ASP.NET CODE

In your .NET Web Application create a new page (ex: TestSSO.aspx) and copy the following code in the code-behind file (TestSSO.aspx.cs):

```
protected override void OnLoad(EventArgs e)
{
    // Assign Time - Number of milliseconds since Jan 1st, 1970
    long time = (long)DateTime.Now.ToUniversalTime().Subtract(new DateTime(1970, 1, 1)).TotalMilliseconds;
    // Assign Vendor code - unique code provided by CHO E3 Support
    string vendor = "1234567890";
    // Assign member number used to define a user - this should be pulled from your local data
    string userid = "456789";
    // Assign Page - (optional) the URL to redirect after successful log in
    string page = string.Empty;

    // The text to sign should have extact order of parameters!
    string textToSign = string.Concat(time.ToString(), "|", vendor, "|", userid, "|", page);

    // Load your Private key  - do NOT store this key in a web accessible location.
    StreamReader file = File.OpenText(Server.MapPath("rsaprivate.xml"));
    string key = file.ReadToEnd();

    // Select target CSP
    RSACryptoServiceProvider rsa = new RSACryptoServiceProvider(1024);

    // Import key
    rsa.FromXmlString(key);

    // Convert text into byte array
    byte[] bytes = Encoding.Unicode.GetBytes(textToSign);

    // Sign the text and get signature in byte array
    byte[] signedbytes = rsa.SignData(bytes, new SHA1CryptoServiceProvider());

    // Convert signature byte array into string
    string signedText = Convert.ToBase64String(signedbytes);

    // You need to change following variable to real club's website url.
    string clubDomain = "www.clubname.com"

    StringBuilder sb = new StringBuilder();
    sb.AppendFormat("{0}/passthrough.aspx?", clubDomain);
    sb.AppendFormat("time={0}&", time);
    sb.AppendFormat("vendor={0}&", vendor);
    sb.AppendFormat("userid={0}&", userid);

    // Following values should be encoded
    sb.AppendFormat("page={0}&", Server.UrlEncode(page));
    sb.AppendFormat("value={0}&", Server.UrlEncode(signedText));

    Response.Redirect(sb.ToString());
}
```

PHP EXAMPLE

Note - we require UTC time to be sent - if your server isn't using UTC time then you may need to use the command: date_default_timezone_set("UTC")

```
<?php

        // Assign Time - Number of milliseconds since Jan 1st, 1970
        $time = time()."000";
        // Assign Vendor code - a unique code provided by CHO E3 Support
        $vendorcode = "1234567890";
        //Assign member number used to define a user  - this should be pulled from your local data
        $userid = "12345";
        // Assign Page (optional) the URL to redirect after successful log in
```

```
        $page = "";

        // Load your private key - do NOT store in a web-accessible place
        $fp = fopen("/rsaPrivate.pem", "r");
        $priv_key = fread($fp, 8192);
        fclose($fp);

        $pkeyid = openssl_get_privatekey($priv_key);

        $data = $time . "|" . $vendorcode . "|" . $userid . "|" . $page;
        $data = mb_convert_encoding($data, "UTF-16LE");

        openssl_sign($data, $signature, $pkeyid, OPENSSL_ALGO_SHA1);
        openssl_free_key($pkeyid);

        while ($msg = openssl_error_string())
        print "error=" . $msg . "\n";

        // replace with the appropriate location
        $url = "http://www.clubname.com/passthrough.aspx"
        . "?time=" . $time
        . "&vendor=" . $vendorcode
        . "&userid=" . $userid
        . "&page=" . urlencode($page)
        . "&value=" . urlencode(base64_encode($signature))
        ;
        header("HTTP/1.1 302 Moved Temporarily");
        header("Location: " . $url);
        exit();

?>
```

JAVA/JSP EXAMPLE

The PKCS8 format works better in Java. Here's how you convert using OpenSSL:

openssl pkcs8 -topk8 -in myrsakey.pem -nocrypt -outform DER –out myrsakey.pkcs8

Create a testsso.jsp file:

```
<%@ page import="java.io.*,java.security.*,java.security.spec.*,java.net.*"%>
<%

// Assign Time - Number of milliseconds since Jan 1st, 1970
String timeStamp = Long.toString(System.currentTimeMillis());
// Assign Vendor code - a unique code provided by CHO E3 Support
String vendorcode = "1234567890";
// Assign member number used to define a user  - this should be pulled from your local data
String userid = "12345";
// Assign Page - (optional) the URL to redirect after successful log in
String pageurl = "";

// Load your private key - do NOT store the private key file in a web-accessible place
InputStream inStream = new FileInputStream("/myrsakey.pkcs8");
ByteArrayOutputStream outStream = new ByteArrayOutputStream();

int count;
byte[] inputbuf = new byte[1024];
while (true)
{
    count = inStream.read(inputbuf);
    if (count <= 0)
    {
        break;
    }
    outStream.write(inputbuf, 0, count);
}

byte[] encodedKey = outStream.toByteArray();
inStream.close();

KeySpec ks = new PKCS8EncodedKeySpec(encodedKey);
PrivateKey privateKey = KeyFactory.getInstance("RSA").generatePrivate(ks);

StringBuilder dataForSignature = new StringBuilder();
dataForSignature.append(timeStr);
dataForSignature.append("|");
dataForSignature.append(vendorcode);
dataForSignature.append("|");
dataForSignature.append(userid);
dataForSignature.append("|");
dataForSignature.append(pageurl);
```

```
// or SHA1withRSA if you generated an RSA key
Signature signWithRSA = Signature.getInstance("SHA1withRSA");
signWithRSA.initSign(privateKey);
signWithRSA.update(dataForSignature.toString().getBytes("UTF-16LE"));

byte[] signedData = signWithRSA.sign();

String value = new sun.misc.BASE64Encoder().encode(signedData);

StringBuilder redirectString = new StringBuilder();
// change the URL www.clubname.com to the appropriate URL for the club's website.
redirectString.append("http://www.clubname.com/passthrough.aspx?time=");
redirectString.append(timeStamp);
redirectString.append("&vendor=");
redirectString.append(vendorcode);
redirectString.append("&userid=");
redirectString.append(userid);
redirectString.append("&page=");
redirectString.append(pageurl);
redirectString.append("&value=");
redirectString.append(value);

response.sendRedirect(redirectString.toString());

%>
```