

Description: Guide to encrypting the credentials used to run GENESIS64 Web Services.

OS Requirement: Windows Server 2003 x64, Server 2008 x64, Server 2008 R2 x64, Windows Vista x64, Windows 7 x64

General Requirement: GENESIS64 installation, Internet access to download utilities from Microsoft and good knowledge of Windows Registry editing.

Introduction

GENESIS64 uses WCF technology to run several of its Web Services. These Web Services impersonate the user you have chosen during the installation. This user's credentials must be stored somewhere for the service to use them. By default, these credentials are stored in plain text in the Web.config file at this location:

C:\Program Files\ICONICS\GENESIS64\WebSites\IcoConfig\

For most users this is not a problem as long as you protect your server file system correctly. No one except the system administrator will be able to open or even see that file.

However, for those installations where security is a top priority you may want to take an extra step to secure your credentials and encrypt them. This application note will walk you through the steps to protect your credentials.

Obtaining the tools

The reason why credentials were stored in plain text goes back to .NET 1.0, for which Microsoft released a hot-fix, a utility to encrypt credentials, and a Support Article describing the procedure to follow to work-around this issue.

This application note is a simplified version of that article, tailored for GENESIS64 users. You can find the original Microsoft article here:

<http://support.microsoft.com/default.aspx?scid=kb;en-us:329290>

You will not need the hot-fix available there, because if you are using GENESIS64 you must already have a version of .NET that includes the hot-fix. You do need the utility to encrypt the data. You can download it from the above URL, under the "More Information" box. The file to download is called "Aspnet_setreg.exe".

This utility will encrypt the credentials and store them into the Windows Registry. We will then tell the Web.config file to take the credentials from there.

Encrypting credentials

The first step is to unpack the self-extracting file we just downloaded. Unpack it wherever you want, for example in "C:\Tools". You will find a EULA file and the actual "Aspnet_setreg.exe" utility.

Make sure you are logged into the system as a user with administrator privileges and open the command prompt. Browse to the location where you unpacked your files (i.e. "C:\Tools").

From that location, run the following command:

```
c:\Tools>aspnet_setreg.exe  
-k:SOFTWARE\MY_SECURE_APP\identity  
-u:"domain\username" -p:"password"
```

Replace "domain" with either your domain or machine name and "username" and "password" with the credentials you want to encrypt. Replace "MY_SECURE_APP" with any name you want. It will be used for the registry key.

The command will output a list of instructions on how you can now use those credentials. This document will cover these steps, so you can safely ignore the command output.

Moving the credentials

You will probably not find the registry keys we just created if you try to open your Windows Registry at this point. This is because the utility we have used is a 32 bit utility, while we are actually running on a 64 bit system. Therefore those keys have been created under "WOW6432Node" instead. This is a problem for us, because IIS will not be able to get the information out of this location. We will need to move them to the correct location.

To move these keys in the folder where we need them you can export the "ASPNET_SETREG" key, as shown in Figure 1.

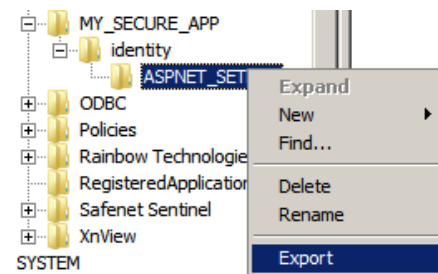


Figure 1 – Exporting registry keys

Now, open the exported file. Remove the “WOW6432Node\” part out of the location name. The part to be removed is highlighted in Figure 2:

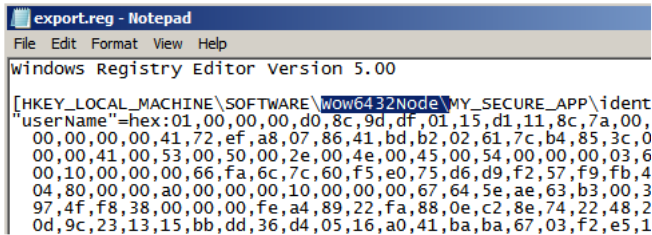


Figure 2 – Correcting the exported file

You can now save your .reg file and double click on it to import these keys back into the registry in the appropriate location.

As you can see from your .reg file (and in the above picture), your credentials are no longer in plain text, but are encrypted into a sequence of hex numbers.

Modifying the Web.config file

The last step to perform is to modify the Web.config file to get rid of the plain text credentials, and to substitute them with a link to their registry counterparts. In this way, IIS will use the new encrypted credentials instead.

Locate and open your Web.config file for editing. Look for an entry similar to:

```
<identity impersonate="true"
userName="MACHINE\Username"
password="Password" />
```

And modify it with this:

```
<identity impersonate="true"
userName="registry:HKLM\SOFTWARE\MY_SECURE_APP\ident
ity\ASPNET_SETREG,userName"
password="registry:HKLM\SOFTWARE\MY_SECURE_APP\ident
ity\ASPNET_SETREG,password" />
```

Where “MY_SECURE_APP” is the same string you have used in the previous section when encrypting your credentials. Save your Web.config file and close it.

Restart IIS and verify the services

To have IIS use the new credentials to run the services we will need to restart it, using “iisreset” from the command prompt. If you are on a domain you may also have to restart the server itself.

To make sure your services are still working properly, you can open IIS and browse to IcoConfig virtual folder under GENESIS64 and right-click on any service (i.e. Security Service) and pick Browse.

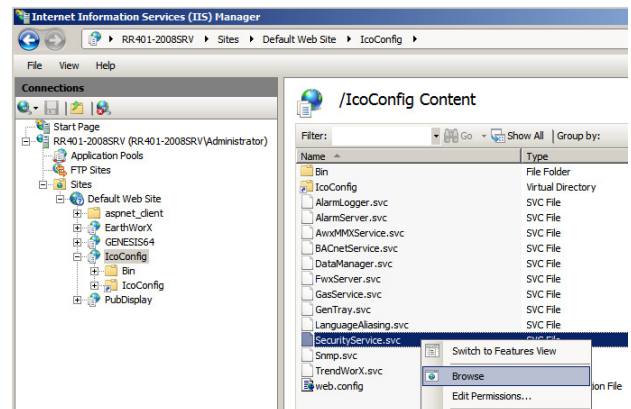


Figure 3 – Verifying the services

The service web page will open. If you can see a list of methods, it means everything is working fine using your new set of securely encrypted credentials.

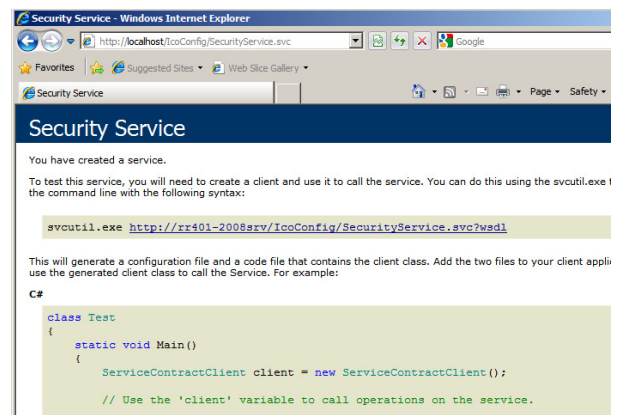


Figure 4 – A correctly running service

What to do in case of problems

This application note is a simplified version of the Microsoft Support Article we mentioned at the beginning of the document.

We have purposely left out several special cases which are not generally needed by our users. If you have any issue following this procedure you may take a look at the original Microsoft Support Article to see if that helps.