

## Re: Asp.net 2.0 deployment with encryption

---

*Source:*

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework.aspnet/2006-04/msg02847.html>

---

- *From:* Chuck P <Chuck@xxxxxxxxxxxxxxxxxxxx>
  - *Date:* Wed, 19 Apr 2006 11:35:17 -0600
- 

Thanks, Steven

I had read the web farm stuff, but didn't think of using it since I don't have a web farm.

I guess I will create a rsa key on the production server.

Export the public xml/key to a common location on some server.

Write a batch file on the development machine that compiles the app and then encrypts the web.config using the xml file on the production server.

That way any developer can deploy the app and I don't have to give the aspnet account any write permissions.

On Wed, 19 Apr 2006 10:37:15 GMT, stcheng@xxxxxxxxxxxxxxxxxxxxxx (Steven Cheng[MSFT]) wrote:

Hi Chuck,

Thank you for posting and glad to see you again.

As for the ASP.NET 2.0 configuration section protection, it provides two encryption approaches, DPAPI and RSA. I think the current approach you're using is the DPAPI one which is mentioned in the following article, correct?

#How To: Encrypt Configuration Sections in ASP.NET 2.0 Using DPAPI  
<http://msdn.microsoft.com/library/en-us/dnpag2/html/PAGHT000005.asp?frame=true>

As for this data protection, it is something like a symmetric data encryption which uses a single shared session key to encrypt and decrypt the data. Also, this session key is machine specific (or user store specific) which makes it not portable from machine to machine. So when you're using this approach (DPAPI) to protect the configuration section, we should do the

Re: Asp.net 2.0 deployment with encryption

final encrypting work on the deployment server rather than on the development server (where you compile the application). And normally, the work (execute the aspnet\_regiis tool from commandline to encrypt the application's configuration section) is done by the deployment server's administrator.

Then, what shall we do if we want to make the encrypting work done at before the application be deployed to the target deployment server (on development server)? Well, this brings out the second option----- RSA data encryption approach. Actually you can also find the above article (about DPAPI approach mentioned this in the final section, about protect configuration data in WEBFARM scenario).

The RSA approach is just based on RSA asymmetric data encryption/decryption which use a public/private key pair. So when we want to make multiple web servers share the protection key setting (e.g do the encryption on the web.config file on one server, and when deploy it to other servers, also want the protected data be usable without additional work), we can create a custom RSA key pair, and on the development server, we still use the aspnet\_regiis tool to encrypt the web.config use the created RSA key pair's public key, and export the private key (which is necessary for decrypting the data) to other servers which will want to decrypt the data (for your scenario, it's the deployment server). And all the tasks mentioned here like creating the RSA key pair, encrypt through it, or export it can be done via the aspnet\_regiis tool.

Here is another MSDN article which mentioned using RSA approach to do the configuration protection (also be referenced in the above article):

#How To: Encrypt Configuration Sections in ASP.NET 2.0 Using RSA  
<http://msdn.microsoft.com/library/en-us/dnpag2/html/paght000006.asp?frame=true>

Hope this helps you.

Regards,

Steven Cheng  
Microsoft Online Community Support

=====

When responding to posts, please "Reply to Group" via your newsreader so that others may learn and benefit from your issue.

=====

This posting is provided "AS IS" with no warranties, and confers no rights.

Re: Asp.net 2.0 deployment with encryption