

Re: Access Control Best Practices for shared hosting seem at odds with Web Site Starters

Source: <http://www.derkeiler.com/Newsgroups/microsoft.public.inetserver.iis.security/2005-09/0077.html>

From: David Wang [Msft] (someone_at_online.microsoft.com)

Date: 09/07/05

Date: Wed, 7 Sep 2005 06:37:16 -0700

I think you misunderstand what "anonymous write" means:

http://blogs.msdn.com/david.wang/archive/2005/08/20/Why_can_I_upload_a_file_without_IIS_Write_Permission.aspx

How the two types of "write" differ is this.

In the case of HTTP PUT, it means that anyone who can send a PUT request to the server can write a file somewhere. This is clearly dangerous if authentication is not required or unprivileged user is authorized to send PUT, and it is the "anonymous write" that is clearly warned.

In the case of HTTP POST or any other request to a server-side application who subsequently decides to write a file, this is less dangerous because it is up to the server-side application to decide how read/write works. IIS has very little say in this instance, if you read and understand my blog post. Security depends on the application itself.

Yes, there is risk in allowing IUSR Write ACLs in the URL namespace, but that is something for the hoster to decide for their web applications. Personally, security is not absolute, and best practices are not perfect --- from a security perspective, I would never make it into a series of check-boxes --- I would seek to understand the web app and then make my choices.

Security is NOT absolute. It is relative and a balance between functionality and risk. There is no way that a single check list of Best Practices is going to functionally apply to all situations; no way. For the same reason, the list of recommend ACLs that you are asking for also does not exist and needs to be generated by yourself for your situation.

--

//David

IIS

<http://blogs.msdn.com/David.Wang>

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

//

"M. M. Rafferty" <mmr@vistagrande.com> wrote in message

news:OxSbv0nsFHA.912@TK2MSFTNGP11.phx.gbl...

The MS Shared Hosting Deployment Guide lists among best practices:

Ensure strong permissions are used on Web content
Use separate anonymous (IUSR) accounts for each Web site
Never allow anonymous user (IUSR) Write permission

The document also describes Isolated Shared Web Hosting where each customer has their own application pool with a unique identity. It states that the host should "Ensure that the Customer-specific identity has the minimal necessary permissions to system resources" but exactly what that means in terms of ACLs apparently has been left as an exercise for the reader.

And therein lies the problem. Or at least part of it.

We are looking at the web site starter kits -- DotNetNuke and the Community Server. It appears that both require write access for the application pool identity below the web root. This has also come up in a few other applications such as shopping carts we have encountered. Usually, the requirement is to allow content management features, generally image uploads, via the browser.

How is this not anonymous write access? Why would Microsoft recommend this to its hosting partners?

Also, another quirk that appears to be the case, at least from what we encountered in some experiments with the Community Server applications, is that one seems to require the application pool identity have list permission starting at the root of the drive all the way down to the web folder. This seems like a bit of a privacy breach at best since it would rely only on security through obscurity in our folder naming.

Can someone explain this apparent contradiction?

Ideally, can someone lay out the recommended ACLs for this scenario for a web host to have in place so that customers' sites are secured and isolated... and still able to run real ASP.NET applications?

Thanks,

Mary M. Rafferty