

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/2005-09/0082.html>

From: M. M. Rafferty (*mmr_at_vistagrande.com*)

Date: 09/08/05

Date: Wed, 7 Sep 2005 22:20:59 -0700

Hi David,

I understand the HTTP PUT. And that it is different. (An IIS thing rather than NTFS.)

However, the full context of the bullet I quoted appears to be the HTTP POST situation. Here is the longer version:

"Never Allow Anonymous User (IUSR) Write Permission

Do not allow anonymous user (IUSR) to have write permission. Allowing write permission means that if the attacker can gain a method to upload the content to the server, then they can write anything onto the server. Guest books and forum software/Web pages are typical applications that require anonymous write access. More secure alternatives are applications that store data in an external database such as Microsoft SQL Server, rather than in database files stored in the Web content directory. There are many free or inexpensive guest books and forums that support using a database in this way."

I always thought "Best Practices" were sort of like a 10 for a gymnast -- the ideal one strives to achieve -- not the practical reality one accepts because that is all they can manage under the circumstances.

As the server administrator, I must assume that an application will be insecure. Attempting to make a decision to the contrary on a case by case situation is not a scalable solution. And it is basically impossible with ASP.NET applications where only the DLLs are uploaded.

As far as I can tell, in order to run most web applications, we need Full Trust allowed. (Database access and a few other things required it?) I don't grasp what that means in terms of what the application pool users can do. Just because I can't do more than the first "hello world" sample in ASP.NET doesn't mean someone else can't create an application that uses features of Windows I didn't even realize existed. So I am worried that if we were to loosen up security for these sorts of applications, there could be unpleasant results.

For instance, if a client application (unintentionally) allows visitors to upload anything to any place on the website, what then? Other than burning CPU and disk space with the traditional uses of compromised sites (warez, file sharing, spamming) will the risks be confined to that Application pool and the associated web structure? Is there enough isolati