

FPSE2002 Shared Hosting Flaw Workaround

Source: <http://www.derkeiler.com/Newsgroups/microsoft.public.inetserver.iis.security/2002-08/3352.html>

From: al (news@thispartisfake-13c.com)

Date: 08/28/02

From: "al" <news@thispartisfake-13c.com>

Date: Tue, 27 Aug 2002 17:57:29 -0700

I have begun a workaround for the FrontPage 2002 Extensions use of the Interactive and Network permissions for those attempting Secure Shared Hosting.

Since it is not an option to use the 2000 Extensions on the Forthcoming .NET server and I do not expect the Flaws to be fixed, I think it is in everyone's best interest to come up with a workaround that disables a minimum of features and is not based on obscurity. Please make sure you read the 'Bummers' at the bottom if you are going to make any of these modifications. Please reply to the list.

This configuration is tested on .NET Standard Server RC1 as a Stand Alone server.

1. Users And Groups:

Create a User for the SharePoint AppPool and Join it to the 'Administrators' Group.

Create a Group called 'DenyPermChange' and Add 'Everyone' to it.

Create a unique anonymous user for each Web Site Instance.

Make them members of the 'Guests' group only.

When creating FrontPage users this configuration assumes they only belong to the group 'Users'.

2. User Rights in Local Security Policy:

Remove 'Everyone', 'Users' from 'Bypass Traverse Checking'.

Add 'Network Service'.

3. SharePoint Admin Web Instance and App Pool.

Remove Anonymous Access From the Web Instance (why is it enabled by default?).

Change App Pool from Local System to use the specific user created in step 1.

When accessing SharePoint Admin site never log on as the Owner of WWWRoot and the Web Site Home Directories. The User needs to be a machine Administrator.

Turn off the ability to check server health for all FrontPage Roles

unless you want lots of error events in your security auditing logs.

4. Content Root Drive Permissions

(don't bother doing this workaround if content is on the same partition as system):

Remove all permissions except 'Administrators', 'System' Full Control

(maybe reset all child objects if this is a fresh install).

Press the Advanced Button of the Security Tab for this object

Add the Group 'Users' and give them 'Traverse Folder' on 'This folder only'

.

5. InetPub Folder Permissions:

This folder should inherit from its parent and in addition

press the 'Advanced' button of the 'Security' tab for this object

Add the Group 'Users' and give them 'Traverse Folder' on 'This folder only'

.

6. WWWRoot Folder Permissions:

This folder should inherit from its parent and in addition

press the 'Advanced' button of the 'Security' tab for this object

Add the Group 'Users' and give them 'Traverse Folder' on 'This folder only'

.

Add the Group 'DenyPermChange' created in step 1 and deny 'Change Permissions' on

'This folder and subfolders' and check the box to

'apply these permissions to this object and/or containers within this container only'.

Change Owner to a special user that will not use the FrontPage/SharePoint admin tools

and not the Group 'Administrators'.

7. New Web Site home directories:

This folder should inherit from its parent and in addition

press the 'Advanced' button of the 'Security' tab for this object

Each time you create a new directory, make sure the owner is set as in step

6.

Explicitly Add the User who will have the FrontPage Administrator role for this site and the unique Anonymous User to have read/list permissions on this folder only.

Bummers:

The root dir of each website has to be managed specially by the Built-In Administrator.

This means the user assigned the FrontPage role of Administrator cannot add users

to this dir with the FrontPage Administration Web Pages.

The Members of DenyPermChange could be reduced from everyone to a less inclusive group

than Everyone but a specific User who manages the permissions on the root dir of each

website cannot use the FrontPage/SharePoint Administration tools without messing up the permissions. Here is why:

The reason this all works is that each web site's home directory has explicit permissions and is not allowing directory traversal to the weak group 'Users' to which the anonymous users and FrontPage users belong by default (look at the members of the Group 'Users' for details).

The FrontPage/SharePoint Admin tools always try to modify the permissions on WWWRoot and the Web site's home dir and add Network and Interactive groups

thus enabling FSO or OLEDB (or other means) to access another web site's home dir . By blocking the right to Change Permissions to Everyone on these folders the tools fail in the attempt but succeed in the folders and files below. Of course the granularity of security achieved is only per web site and not per sub web. Sub web authors in the same site can still abuse each other.

Note that 'System' and the Owner still have the ability to change permissions on an object, no matter what.

This workaround does not yet take into consideration ASP.NET and its functional requirements and its similar security deficiencies for Secure Shared Hosting.

It also may impact backup solutions that use the 'Change Permissions' DAC and products that live in folders accessible to users that require directory traversal to reach without adding explicit traversal rights.

If the mail root lives in the same partition as content then CDO may be affected without adding explicit traversal rights.

Running the SharePoint App Pool

as an Administrator is no worse than the default, 'Local System'. It would be nice if

it was possible to run at a lower level but it isn't.

It is possible that performance will be degraded by removing the Bypass Traverse Checking right from Users.

Ideally for this to work for provisioning, a tool for managing the home directory of each website should be created since this is the one feature of the FrontPage/SharePoint admin

tools this workaround has broken.

Normally I wouldn't suggest such an extreme workaround since a change in Microsoft's products could easily break this, however the Flaw is about a year old and I do not expect it to be fixed in the near future.

al.NETIsNOTSecureForSharedHosting