

IIS Security Tips

Hal Flynn 2000-09-25

IIS Security Tips

by Marc Maiffret, Rain Forest Puppy, Mark Burnett and Ben Greenbaum

last updated Monday, September 25 2000

Mark Maiffret - eEye Digital Security

<http://www.eeye.com>

marc@eeye.com

1. Do not ever use the "default web" always create a brand new web in a completely different path on the hard drive.
2. Do not use ANY ISAPI filters unless you must... I can not stress enough **MUST MUST MUST**.
3. Do not sleep easy at night because there is a new IIS hole bound to come out at least once a month.
4. The only way to barely make it in the world of IIS web servers is to guard yourself from the attacks that are yet to come. Basically it does not matter how many service packs or hotfixes you install there will always be that one new hole that comes out that bites you in the ass. We can learn from past IIS holes though and create "CHAM (Common Hacking Attack Method) Filters" that will effectively protect you from future IIS attacks that might pop-up. For example you should never have a request to your website that has `"../../../../../../../../"` somewhere within it. There are a lot of "path attacks" against various ISAPI filters... they all have the same basic characteristics so just stop those types of information from entering the server in the first place and you will effectively block against future ISAPI filters or cgi's that might be vulnerable to path attacks.

Rain Forest Puppy - RFP Labs

<http://www.wiretrip.net/rfp/>

rfp@wiretrip.net

1. Treat your include files with care. Many people put extra authentication information in

include files. This is fine, as it centralizes sensitive information and reduces the number of total possible exposures. However, people feel the uncanny need to use '.inc' as an included file extension; the problem is that this returns the file without processing, so if you were to leave .inc files laying around, anyone can view them. You can code your include files so they are full (but non-functional) .asp files themselves, and use an .asp extension--this stops casual viewing. Also, you should try to keep sensitive information outside the web root, if possible. If it's not possible, make sure the ACLs on that directory prevent access.

2. Be aware of text/web editors that make '.bak' backups of files. This, coupled with a massive directory FTP upload, might leave '*.asp.bak' files laying around, which would allow retrieval of source, since .bak is a benign file extension. If your site does exhibit this problem frequently, you may consider mapping the '.bak' extension to the '.asp' handler, just to be safe.

3. While the extra overhead of ODBC (compared to OLEDB) is great, the benefit is that connection information is stored in the DSN, and not the .asp file itself. This means even if someone retrieved the source of your .asp file, they can't start connecting to the database with the information, since there's nothing but a DSN to be had.

4. Make sure you are aware of what DSNs and database drivers your systems had available. Like extension mappings, you should review each one, and keep them to a minimal. For instance, MDAC 1.5 (Jet 3.5) is known to be vulnerable; MDAC 2.0 updates Jet (to 4.0), but leaves a mechanism to invoke the older Jet 3.5. MDAC 2.1 is overall fixed, but you could still be vulnerable if you applied MDAC 2.0 before MDAC 2.1.

Mark Burnett - Xato Network Security

<http://www.xato.net>

mburnett@xato.net

Know your web site -- Know you web site in and out. Know the function of every file. Know the ACL' of every file. Delete, rename, or quarantine every file you don't know. Know all the virtual roots and where they physically reside. Frequently open MMC and frequently produce directory listings. Frequently list the most recently changed files in your web root. Keep copies of your site offline on read-only media for base reference.

Default ACL's are usually wrong -- Review the ACL's for every sensitive file on your site.

Remove read permissions on scripts. Check the files as well as the directories themselves. Be

wary of allowing files that have both write and read access. Keep executable directories to a minimum and keep them all in one place.

Get your web settings right -- Eliminate extension mappings and http methods you do not explicitly use. Don't enable parent paths. Don't send detailed ASP error messages to the client. Don't index a directory unless you intend to allow someone to search that directory. Add a few more default documents like index.htm and index.html. Always make applications first check that a file exists.

Consider obscurity -- Obscurity isn't good security but it is better than using defaults. There is nothing wrong with not installing software in their default directories to foil script kiddies or the easily discouraged hacker. Don't save orders to orders.txt. Don't put logs in a directory named logs. With so many other sites for a hacker to hack sometimes a little obscurity can be enough to save you.

Your site is never secure -- Don't brag about how secure your site is on your privacy page. Don't even mention how secure you are because you will start believing it yourself and let your guard down. Security is a process that you engage in, not a state that you attain.

Ben Greenbaum - Securityfocus.com

<http://www.securityfocus.com>

bgreenbaum@securityfocus.com

You will never know every directory traversal and buffer overflow vulnerability. To mitigate the possible damage from future or undisclosed problems of this nature:

Put your webroot on a separate partition. That way if an attacker does exploit some unpatched directory traversal bug, they're still stuck on the dedicated partition.

Lock down the IUSR_machinename account. Give it 'No Access' permissions on everything outside of the webroot you can get away with, and Read-only wherever applicable on what's left. Take away any of its rights you wouldn't want any random anonymous user to have to your network. However, it would be a good idea to test your changes before deployment, as this is an easy way to break things if you make a mistake.

Also, if you must have ftp and http access to the same folder, try to think of a way not to. If

you still have to have this, by all means ensure that no folder has both write and execute abilities available to the anonymous account.

[Privacy Statement](#)

Copyright 2006, SecurityFocus