

Re: Auto deploy from W2K machine w/IIS Lockdown applied

Source:

<http://www.derkeiler.com/Newsgroups/microsoft.public.dotnet.framework.aspnet.security/2002-12/2301.html>

From: Mike Moore [MS] (michmo@online.microsoft.com)

Date: 12/24/02

From: michmo@online.microsoft.com ("Mike Moore [MS]")

Date: Tue, 24 Dec 2002 00:30:30 GMT

Hi Norm,

* Check the configuration for LockDown.

LockDown comes with a help file named iislockd.chm. If you double-click it, you can see help information. It also comes with RunLockdUnattended.doc.

Find the file iislockd.ini on your machine and open it in NotePad. You will see that it has several sections. At the top is the [info] section. If the info section contains "Unattended=TRUE:", then the setting "UnattendedServerType" should designate which of the several sections below is currently active on your machine. If Unattended is set to FALSE, then look in the folder that contains iislockd.exe. In that directory, look for Oblt-log.log. This should show the settings that were used when LockDown was installed. Then you can determine which of the sections within iislockd.ini is the active section.

If you cannot determine which section is active, you can experiment to find it. If you can browse ASP files, then sections disabling ASP are not active. Of the remaining sections, set them consecutively to disable ASP until ASP stops working. That's the active section.

In the active section, review the settings to see if any of them prevent activities that you want to allow, or if any allow activities that you want to prevent. Change the settings as appropriate and experiment with any settings you don't know. For all settings, make note of what they were previously so you can put them back if needed.

* Check the configuration for UrlScan

UrlScan comes with UrlScan.doc and urlscan_unattend.txt and readme.txt. It also comes with multiple INI files.

Within the active section of iislockd.ini, look for an entry named UrlScan_IniFileLocation. This will indicate which of the UrlScan INI files is active. If you have trouble determining which INI file is active,

experiment by setting the INI files consecutively to disable ASP. When ASP is actually disabled, that is the active INI file. It will probably be "urlscan_dynamic.ini".

Examine the settings within the INI file and change them as needed.

NOTE: the UrlScan INI file has multiple sections and only some of them are active. At the top of the file, look for:

- UseAllowVerbs
- UseAllowExtensions

Further down you will see sections for

[AllowVerbs]
[DenyVerbs]
[DenyHeaders]
[AllowExtensions]
[DenyExtensions]
[DenyUrlSequences]

If "UseAllowVerbs" is set to zero, then all verbs will be allowed except those listed in the DenyVerbs section. If UseAllowVerbs is set to 1, then all verbs will be denied except those listed in AllowVerbs.

"UseAllowExtensions" is similar. The other two sections, "DenyHeaders" and "DenyUrlSequences" are active regardless of the settings at the top of the INI file.

Change the settings as appropriate and experiment with any settings you don't know. For all settings, make note of what they were previously so you can put them back if needed.

* Check NTFS permissions for the EXE (and its related files such as application.config). Change the NTFS permissions to allow your visitors to read, but not execute the EXE (if you want to, you can also allow them execute, but that will allow them to run the EXE on the server, compared to allowing them to only download it and run it on their own machines). You need to be sure to allow access to the user account actually being used by your visitors. With anonymous access, this will usually be either ASPNET or Iusr_machine, where "machine" is the machine name of the server. With non-anonymous access and with impersonation, this account can vary. What ever it is, that's the account(s) you need to grant access.

If it still fails after all of the above, then try some experiments. If this server is connected to the internet, then modify your IIS settings to restrict access, such as restrict access by IP address.

* grant "everyone" full access to your whole directory
* make backups of the above INI files and change the settings to allow maximum access (such as, use DenyVerbs and leave the DenyVerbs section empty). You can even change the IIS settings to remove the LockDown and UrlScan filters.

*** Remember to put these things back.

--

If it's still failing, then repost with the following additional

microsoft.public.dotnet.framework.aspnet.security: Re: Auto deploy from W2K machine w/IIS Lockdown applied

```
>> This is the executable that hosts out-of-process applications.
DLLHOST.EXE
>> is the executable that the members of the "Web Applications" group have
to
>> start.
>>
>>
>> Thanks,
>> Bassel Tabbara
>> Microsoft, ASP.NET
>>
>> This posting is provided "AS IS", with no warranties, and confers no
rights.
>> -----
>> | Content-Class: urn:content-classes:message
>> | From: "Norm Dotti" <normd@knorrassociates.com>
>> | Sender: "Norm Dotti" <normd@knorrassociates.com>
>> | Subject: Auto deploy from W2K machine w/IIS Lockdown applied
>> | Date: Fri, 20 Dec 2002 07:28:28 -0800
>> | Lines: 11
>> | Message-ID: <048001c2a83c$71ecbe80$cef82ecf@TK2MSFTNGXA08>
>> | MIME-Version: 1.0
>> | Content-Type: text/plain;
>> | charset="iso-8859-1"
>> | Content-Transfer-Encoding: 7bit
>> | X-Newsreader: Microsoft CDO for Windows 2000
>> | X-MimeOLE: Produced By Microsoft MimeOLE V5.50.4910.0300
>> | Thread-Index: AcKoPHHspc8v/fjcQXWJI2RzC/hfQ==
>> | Newsgroups: microsoft.public.dotnet.framework.aspnet.security
>> | NNTP-Posting-Host: TK2MSFTNGXA08 10.40.1.160
>> | Path: cpmsftngxa09!TK2MSFTNGP08!cpmsftngxa06
>> | Xref: cpmsftngxa09
microsoft.public.dotnet.framework.aspnet.security:3449
>> | X-Tomcat-NG: microsoft.public.dotnet.framework.aspnet.security
>> |
>> | I can't seem to get autodeploy to work from a W2K Server
>> | machine w/the IIS Lockdown applied. I keep getting a 404
>> | when I try to get the exe (e.g. http://webserver/app.exe).
>> | If I turn on directory browsing I can see the exe file
>> | there so I know I'm asking for it correctly. I've got the
>> | app set up for Script-only in IIS. I've got anonymous
>> | access set up. I've removed .config from the list of files
>> | to not download. Does the lockdown tool somehow prevent
>> | the detection of a .net exe? I'm not all that familiar
>> | w/what the lockdown tool does behind the scenes. Any help
>> | would be appreciated.
>> |
>
```