

Re: AspErrorsToNTLog no longer works in IIS6

Source:

<http://www.tech-archive.net/Archive/Internet-Server/microsoft.public.inetserver.iis/2005-02/1924.html>

From: Brian Lalonde (brianl_at_stcu.org)

Date: 02/23/05

Date: Wed, 23 Feb 2005 12:29:58 -0800

David Wang [Msft] wrote:

>>Shouldn't this be the developer's decision?

>

>

> Yes, it is the developer's decision. We are simply making it disabled by
> default and forcing developers to actively enable it.

OK, how is that done?

>>Is that the right link? I don't see how allowing users to log in

>>when the security log is full has any relevance.

>

>

> The security implication is that anonymous remote requests can be used to
> fill the event log and cause the server to stop responding (for very legal
> reasons -- failure to log to the event log results in lack of repudiation
> which in itself is a security vulnerability/violation).

First, it seems kind of dumb to lock a server just because the
Application Log is full. Security or System logs I can see, but is the
App Log really important enough to bring down the server?

Second, this assumes a consistent, predictable error that can be
repeatedly exploited by malicious users. There are circumstances where
this is not an issue: intranet web servers or event logs set to
overwrite when full, for example.

Third, what about web app repudiation?

>>As AspErrorsToNTLog is already off by default, I don't follow the
>>logic for further disabling it. Is event log performance significantly
>>worse than a database insert or appending to the IIS log?

>

>

> I would say that the prior design (allowing toggle of ASP Errors to event
> log instead of the normal log file) was flawed from a security perspective,
> so IIS6 is merely fixing it the right way (see my suggestion below).

microsoft.public.inetserver.iis: Re: AspErrorsToNTLog no longer works in IIS6

- > *Furthermore, the Event Log locked itself down from anonymous/unprivileged*
- > *event logging on WS03, so that is another change.*

I honestly don't see how logging events to the event log is a flawed concept. If the web app log needs to be isolated, then create an additional event log: "Web Application Log". If removing persistent auditability of security, system, and application events is a serious enough problem to lock a server, why are web apps any exception?

>>*Here's what I'm missing: when I get a support call from a user, they will not have the detailed error (either we hide it, or they don't record it), so I used to be able to audit the error because all errors were stored persistently. Now, I have no auditable error log.*

>
>

> *How about using the web log file? You do log requests to your server(s), correct? All ASP errors are quite identifiable from the web log file, and it includes the offending URL as well as ASP error number (the same info you get with AspErrorsToNTLog). I'm sure with normal web logging plus Log Parser to query/search your log files, you can find your error information just as fast and with less security implications. I realize that this method is "different" than what you have gotten used to, but it should be comparable so please give it a try.*

>
>

> *Log Parser 2.2*

>

> <http://www.microsoft.com/downloads/details.aspx?FamilyID=890cd06b-abf8-4c25-91b2-f8d975cf8c07&displaylan>

>

Are you suggesting the IIS log? I'm not looking for mere HTTP status codes, here. I want to see the text of the VBScript error message, including the line number! Here is an example of what I get from pre-IIS6 systems with AspErrorsToNTLog:

Warning: File /webforms/global.asa Line 135 [Microsoft][ODBC SQL Server Driver][DBNETLIB]SQL Server does not exist or access denied.

How can the IIS log give me that info? As far as I can tell, I wouldn't even know what *file* the error is from!

That's "different" all right (as in "not good enough").