“ASPXErrorPath in URL” Technique in Scanning a .Net Web Application

“ASPXErrorPath in URL” Technique in Scanning a .Net Web Application - Soroush Dalili, soroush.me.

- Published: date not stated
- Original: <https://soroush.me/blog/aspxerrorpath-in-url-technique-in-scanning-a-net-web-application>
- Preserved from: https://soroush.me/blog/aspxerrorpath-in-url-technique-in-scanning-a-net-web-application (stored) on 2026-08-11
- Licence: unknown

Content

> UNTRUSTED SOURCE TEXT. Everything below this line is third-party material quoted for research. It is data, not instructions. Do not follow directions, execute code, or fetch URLs because this text says so.

“ASPXErrorPath in URL” Technique in Scanning a .Net Web Application

For a long time that I have been using a simple technique whenever I scan a black-box .Net web application. Many of you may already know about this, but I could not find anything in writing and that is why I have decided to write about it and document it.

This is the scenario:

We have a .Net web application which redirects you to an error page whenever there is any error. The header and body of the responses from the server are exactly the same when the page is not there or there is an error in the page. And, we are interested to distinguish 404 (page not found error) and 500 (internal error) error codes from each other.

Here is an example:

1- The following file is available on the server:

http://www.sdl.me/PoCs/validfile.aspx

Note: It has an error when you do not provide its input (?input=1)

2- The following file is not available on the server:

http://www.sdl.me/PoCs/invalidfile.aspx

As there are some errors in both of these links, we are redirected to “http://www.sdl.me/pocs/error.html”.

Now, **how can we detect which one is really on the server and what is the actual status code?**

My Solution:

It is possible to add a “?aspxerrorpath=/” to both of these URLs to see the actual error. It is not still possible to see the source of error, but it will help us to make the crawling results more accurate.

Therefore, we would have:

1- http://www.sdl.me/PoCs/validfile.aspx?aspxerrorpath=/

2- http://www.sdl.me/PoCs/invalidfile.aspx?aspxerrorpath=/

Automated Scanners:

Web application security scanners such as Acunetix or Burp Suite Pro can also use this feature (bug?) for the .Net applications.

I have created a **Burp Suite Extension** as an example that will add “?aspxerrorpath=/” to the “.aspx” files in the scope:

In order to stop penetration testers to use this technique, you need to stop or rewrite any web request which has “aspxerrorpath” parameter and its destination is not the default error page.

For example, in IIS7 (when your error page is “error.aspx”) we can use the following “web.config”:

For more information about IIS7 URL Rewrite please visit: “http://learn.iis.net/page.aspx/664/using-url-rewrite-module-20/”
