IIS Application vs. Folder Detection During Blackbox Testing

IIS Application vs. Folder Detection During Blackbox Testing - Soroush Dalili, soroush.me.

- Published: date not stated
- Original: <https://soroush.me/blog/iis-application-vs-folder-detection-during-blackbox-testing>
- Preserved from: https://soroush.me/blog/iis-application-vs-folder-detection-during-blackbox-testing (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.

IIS Application vs. Folder Detection During Blackbox Testing

When testing a website on IIS, it is sometimes important to know whether a path is an application or a folder (or a virtual folder). I am intruding a new sneaky method using some ASP.NET features that can be used to verify this in a blackbox assessment.

You can check https://docs.microsoft.com/en-us/iis/get-started/planning-your-iis-architecture/understanding-sites-applications-and-virtual-directories-on-iis if you are not familiar with virtual directory and application terms in IIS.

Why knowing this is useful?

I explain this using the following examples.

If we can upload a file in an IIS app rather than a folder, we can do a lot more to gain RCE by uploading a web.config or a resource file for example as this is normally as effective as uploading a file on the root of a website.

We also need this information when we have the keys to create the ViewState otherwise we have to use trial and error. See the Exploiting Deserialisation in ASP.NET via ViewState post for more details.

How can we do this?

By sending one of the following GET requests to a path, if the application responds with the status code `200 Ok` and some JavaScript code, it is an application. If it responds with the status code `500 Internal Error`, it is a folder:

Copy

```
1http(s)://target/path1/path2/profile_json_appservice.axd/jsdebug

2

3or

4

5http(s)://target/path1/path2/profile_json_appservice.axd/js
```

The `Role_JSON_AppService.axd` or `Authentication_JSON_AppService.axd` can also be used instead of `Profile_JSON_AppService.axd`.

This would work even if these services are disabled (default). Here is an example that shows `/start/` is an application:

https://office.live.com/start/profile_json_appservice.axd/jsdebug

`/stat/` is a folder/virtual folder:

https://office.live.com/stat/profile_json_appservice.axd/jsdebug

How did I find it?

I found this whilst I was reviewing a portion of ASP.NET Framework: https://referencesource.microsoft.com/#system.web.extensions/Script/Services/WebServiceData.cs

Side notes:

When these services (Profile, Authentication, and Role) are enabled, it is also possible to send POST requests to their endpoints. Perhaps they should be reported as informational issues in an assessment in order to make sure they are really needed for the operation of the website. These web services might lead to information disclosure or password guessing attacks as well. The following HTTP request shows a sample request to the `login` endpoint of the Authentication service:

Copy

```
1POST /someapppath/authentication_json_appservice.axd/login HTTP/1.1

2Host: target

3Content-Length: 69

4Content-Type: application/json

5

6{"userName":"guest","password":"guest","createPersistentCookie":true}
```