

Warning broken users with JavaScript

Introduction

The main concern for a web site operator that wants to deploy dual-stack service on his site, is the existence of *broken users*, i.e., users that could access his site fine while it was available over IPv4 only, but for some reason have problems accessing it when it is made available over IPv6 as well. A list of the most common causes for this is found in [Customer problems that could occur](#)

At the time of writing, measurements done by several organisations indicate that at least 1 in 2000 users currently have this problem, see:

[Google's presentation at RIPE 61](#)

[Redpill Linpro's presentation at RIPE 61](#)

[Yahoo's presentation at IETF 77](#)

One thing a web site operator can do about this problem, is to automatically identify these users whenever they visit his web site (which presumably is IPv4-only at that point), and display a warning of some kind that informs the users that they have a problem, and preferably how to fix it, too. That way, when the site is finally dual-stacked, the users will have had ample warning.

It is also useful to warn broken users in this way even after the site has been dual-stacked, as they will in most cases be able to load a dual-stacked page if they are patient enough to endure an initial connection timeout over IPv6.

A much more comprehensive (and probably better) solution for testing the user's IPv6 connectivity is available from [falling-sky](#), the open-source variant of Jason Fesler's [test-ipv6.com](#).

JavaScript code

Add this code near the bottom of your HTML pages. It should work as-is, but see below for some more discussion about how to adapt it better to your site.

```
<!-- these are the test elements that will be loaded by the JavaScript below -->
<IMG id="1x1_ds_testing_1" class="1x1_ds_testing" width="1" height="1">
<IMG id="1x1_ds_testing_2" class="1x1_ds_testing" width="1" height="1">
<IMG id="1x1_ds_testing_3" class="1x1_ds_testing" width="1" height="1">
<IMG id="1x1_v4_testing_1" class="1x1_v4_testing" width="1" height="1">
<IMG id="1x1_v4_testing_2" class="1x1_v4_testing" width="1" height="1">
<IMG id="1x1_v4_testing_3" class="1x1_v4_testing" width="1" height="1">

<!-- Remove the line below if you're already loading the jQuery library -->
<SCRIPT
src="https://ajax.googleapis.com/ajax/libs/jquery/1.4.2/jquery.min.js"></SCRIPT>

<SCRIPT language="javascript" type="text/javascript">
$(window).load(function() {
    // Defines for how many milliseconds we're willing to wait for the test elements
load before
    // considering it a failure. Should not be more than 21 seconds, as that's
Microsoft Windows'
    // connect() timeout.
    var timeout = 5000;

    // Records the number of dualstack and ipv4-only successes.
    var ds_ok = 0;
    var v4_ok = 0;

    $('1x1_ds_testing').load(function() {
ds_ok++;
    });
});
```

```
    $('#1x1_v4_testing').load(function() {
v4_ok++;
    });

    // Load the test images while trying to avoid caches.
    var rand = Math.random();
    $('#1x1_ds_testing_1').attr('src', 'http://<dualstacked hostname
1>/lib/1x1.png?rand=' + rand);
    $('#1x1_ds_testing_2').attr('src', 'http://<dualstacked hostname
2>/lib/1x1.png?rand=' + rand);
    $('#1x1_ds_testing_3').attr('src', 'http://<dualstacked hostname
3>/lib/1x1.png?rand=' + rand);
    $('#1x1_v4_testing_1').attr('src', 'http://<ipv4-only hostname
1>/lib/1x1.png?rand=' + rand);
    $('#1x1_v4_testing_2').attr('src', 'http://<ipv4-only hostname
2>/lib/1x1.png?rand=' + rand);
    $('#1x1_v4_testing_3').attr('src', 'http://<ipv4-only hostname
3>/lib/1x1.png?rand=' + rand);

    // Check the result of the test after the timeout have expired.
    var t = setTimeout(function() {
        // We take the conservative approach here. 100% success with
        // IPv4 combined with 0% success with dualstack is necessary
        // to give user a warning about broken IPv6.
        if(v4_ok == 3 && ds_ok == 0) {
            alert("You have a problem accessing dual-stack sites!");
        }
    }, timeout);
```

```
} );  
</SCRIPT>
```

Improving the warning

The code above will just throw a JavaScript *alert()* pop-up, which is likely not what you want. You probably want to replace this with a nicer-looking error box/banner that gives a warning in the same language as the rest of the site, and giving pointers to further information so that the user will be able to solve the problem himself, or alternatively contact you for help.

Hosting of the test elements

You should make sure to use different hostnames for each URL, preferably pointing to different IP addresses. This prevents browsers from doing clever optimisations which could trick the test.

Changing the timeout and number of test elements

The timeout setting should be higher than the time required by a normal client to load the test images, but lower than the systemic connect() timeout of the common operating systems. The lowest such timeout is, to the best of my knowledge, 21 seconds (Microsoft Windows).

You can also modify the number of test elements loaded if you feel that six of them makes the test too heavy-weight. I would not go below one ipv4-only PNG though, because otherwise you might be testing whether or not the machine that hosts the test elements is up or not; or below two dual-stacked PNGs, because a single load failure could be caused by a number of unrelated things, such as the user having a congested Internet connection.

Author and copyright information

The code was written by me, [Tore Anderson](#), and is released into the public domain. Do with it what you wish.

Sites who have this or something similar in production

Please add your site here if you add such a test, so that others can have a look at it for inspiration!

- [Brian Carpenter's home page](#)
- [Consulintel web site](#)