Apple's Safari violates RFC2616

Today I faced an issue, where HTTP redirections didn't work out as expected on Apple's Safari browser.

This came up while I was coding up some sort of web-based login redirector, which is stacked up in three layers:

- html login form
- login preprocessor (server side scripting)
- login processor (server side scripting)

This solution was required to implement a generic way to create branded login forms, which will send their login requests to a unique, centralized login preprocessor, which will - after doing some internal magic - redirect to the final login processor.

Now, the login preprocessor was set to do redirects to the final login processor using HTTP/1.1 temporary redirects (http reply code 307) to preserve already existing POST data.
While this was working out properly on most browser, I stumbled accross Safari, which will silently discard all POST data.

Checking out the access logs revealed some interesting facts.
The first excerpt shows the access as performed by Firefox:

192.168.0.2 - - [23/Jan/2008:16:32:02 +0100] "POST / HTTP/1.1" 307 20 "https://xyz/" "Mozilla/5.0 (Windows; U; Windows NT 5.1; de; rv:1.8.1.11) Gecko/20071127 Firefox/2.0.0.11"
192.168.0.2 - - [23/Jan/2008:16:32:05 +0100] "POST /some_other_location HTTP/1.1" 200 7934 "https://xyz/" "Mozilla/5.0 (Windows; U; Windows NT 5.1; de; rv:1.8.1.11) Gecko/20071127 Firefox/2.0.0.11"
As required by the 307 temporary redirect Firefox will resubmit to the new location using a POST request (preserving existing POST data).
The first request actually refers to stage 2 (login preprocessor), the second request reflects the resubmission to the final login processor (step 3).

Now this is was Safarie does:

192.168.0.3 - - [23/Jan/2008:16:31:26 +0100] "POST / HTTP/1.1" 307 20 "https://xyz" "Mozilla/5.0 (Windows; U; Windows NT 5.1; de-DE) AppleWebKit/523.15 (KHTML, like Gecko) Version/3.0 Safari/523.15"
192.168.0.3 - - [23/Jan/2008:16:31:26 +0100] "GET /some_other_location HTTP/1.1" 200 7931 "https://xyz" "Mozilla/5.0 (Windows; U; Windows NT 5.1; de-DE) AppleWebKit/523.15 (KHTML, like Gecko) Version/3.0 Safari/523.15"
As we can see, it also receives a 307 temporary redirect from the server and follows it (which is correct as of RFC2616) in the first request (again step 2 as outlined before).
However, the second request is resubmitted using GET, which means that all previously existing POST data is lost (this is step 3 as outlined before).

In this case, Safari clearly violates RFC2616, which states in Chapter **10.3.3 302 Found**:

RFC 1945 and RFC 2068 specify that the client is not allowed
to change the method on the redirected request. However, most
existing user agent implementations treat 302 as if it were a 303
response, performing a GET on the Location field-value regardless
of the original request method. The status codes 303 and 307 have
been added for servers that wish to make unambiguously clear which
kind of reaction is expected of the client.

As a side note to this excerpt from the RFC I shall note, that status code **302 Found** as referred-to by RFC2616 used to be **302 Moved Temporarily** (as of RFC2068), which has been redefined to **307 Temporary Redirect**.

As such, while status code **303 See Other** clearly states, that GET should be used upon redirect, statements made in **RFC2616's Chapter 10.3.3 302 Found** also apply to **307 Temporary Redirect**.

In this case it means, that Safari violates the standard when it changes the access request method to GET.