Naming Requirements for Hostnames and Domains

From time to time we find that either authentication via Single Sign On or stateful application fail. This is due to the fact that the host and/or domain names are not standard conform. BSP attempts to set a session cookie in the browser to store the specific information, that is then silently discarded by the browser, as the location information in the cookie (based on host and domain names) are not valid. On the next HTTP roundtrip, this information will not be available, and the there will be either a request for new authentication or a new session will be opened (or both).

There are Internet standard that govern the naming convention of domains. Note that this issue depends very much on how strict the browser implements the rules. Specifically reference SAP Note URL requirements due to Internet standards.

In summary, the rules are:

- Domains with the extension "com", "edu", "net", "org", "gov", "mil" or "int" must include at least one additional domain component (usually the name of the company or organization)
- Any domain with a different extension (including the national top-level domains in particular, for example, "de", "uk", "fr", and so on) must consist of at least two additional domain parts. Specifically with top-level country domains, we have seen that the browsers often accept also only one extra domain part. But for other non-standard top-level domains, the two additional parts seems to be the rule.

Additionally, what we have seen in customer installations, is that once a host name of the domain name contains a '_' character, that the browser refuses to accept the cookies that are set by the server. Specifically reference SAP Note Hostnames and Cookies.

For example: we have a setup where the development system is called dev_sys and the quality system is qsys. Then given the fully qualified domain names, we have the following:

```
  dev_sys.company.co.xx   --> FAILED
  qsys.company.co.xx      --> WORKED
```

Troubleshooting

Have either non-standard domain names, or somewhere a space in host name or any of the domain parts? Then it could be possible that cookies will not work, causing failure in either session management or Single Sign On.

To test, use the following small test program (will be shipped in later SPs as IT05/test.htm). Create a new BSP page, with one page attribute "counter TYPE string", and paste the following code into the layout.
BSP Application: it05, stateful
Page: test.htm

Page Attributes: counter TYPE string

runtime->keep_context = 1.

DATA: host TYPE string, domain TYPE string.
host = request->get_header_field( 'Host' ).
SPLIT host AT ':' INTO host domain.

domain = host.
WHILE STRLEN( domain ) > 1 AND domain(1) <> '.
   domain = domain+1(*).
ENDWHILE.
domain = domain+1(*).

DATA: testCookieHost TYPE string, testCookieDomain TYPE string.
request->get_cookie( EXPORTING name   = 'testCookieHost'
IMPORTING value  = testCookieHost ).
request->get_cookie( EXPORTING name   = 'testCookieDomain'
IMPORTING value  = testCookieDomain ).

<html><body><form>
<br><input type=submit value="<%=counter%>">
<br>testCookieHost: <%=testCookieHost%>  <% IF testCookieHost <> counter.%> ERROR! <% ENDIF.%>
<br>testCookieDomain: <%=testCookieDomain%>  <% IF testCookieDomain <> counter.%> ERROR! <% ENDIF.%>
</form></body></html>

<% counter = 1 + counter. 
response->set_cookie( name   = 'testCookieHost'
path   = runtime->page_url
value  = counter
domain = host ).

response->set_cookie( name   = 'testCookieDomain'
path   = runtime->page_url
value  = counter
domain = domain ).
%>
All that this program does, is run stateful and increments a counter for every roundtrip. This is checked and confirmed by the counter that is only stored on the server, where the value is incremented. In addition, it sets and reads per roundtrip a cookie on both host (as is done for session cookies) and on sub domain (as is usually done for SSO cookies). To test, just run the program and keep pressing the button.

If the host and domain names are correct, and the cookies work correctly, then the values of the cookies will increment in lock step with the cookies. If host name or domain is not standard conform (to whatever degree enforced by the browser!), the either one or both of the cookies will fail! In case of failure, also the BSP runtime will fail!

Expected results in in working and failure case:

**WORKING**

![HTTP Trace Working](image)

**FAILED!**

![HTTP Trace Failed](image)

This problem can also easily be verified with a HTTP trace. In the case over everything working, we will see the two set cookie sequences in the HTTP response (Set-Cookie headers), and then with the next HTTP request the two cookies will be send to the server again (one Cookie header).
However, once the cookies start to fail (silently), we will see for each and every HTTP response a new sequence of Set-Cookie headers (at least for the new session ID). For the next HTTP request, our test cookies will just not be there. They were never placed into the cookie jar.

Notice in this example above, we changed the host name to contain spaces. We see how the cookies are set, but with the next request there are no cookies transmitted (no Cookie headers!).

**Solution Space**

There is unfortunately **NO** solution for this problem. It is the way that the Internet standards are defined, and enforced by the browsers. The only approach will be to reconfigure the WebAS to have a valid host and domain name.

One quick workaround, that can be done on each and every client machine, is to temporarily define a pseudo name, and to update the local `/etc/hosts` file to have an IP address for this name. On Windows XP this file is stored in path: `\windows\system32\drivers\etc\hosts`. Load the file with notepad, and add a new entry:
Above, the IP address must be that of the WebAS server. The name can be any valid standard Internet conform address. In the browser, now use URLs of the form:

```
http://myhost.mydomain.com/sap/...
```

With this approach, the browser will resolve the pseudo name via the `/etc/hosts` file to a valid IP address.

## References

- Fully Qualified Domain Name (FQDN)
- Domain restrictions in a portal environment
- URL requirements due to Internet standards
- Error message for domain name with underscore
- Hostnames and Cookies