

Re: openssl from tcl

Source: <http://coding.derkeiler.com/Archive/Tcl/comp.lang.tcl/2008-04/msg00495.html>

- *From:* vitick@xxxxxxxxxx
 - *Date:* Wed, 9 Apr 2008 07:03:52 -0700 (PDT)
-

On Mar 26, 12:09 am, Erik Leunissen <l...@xxxxxxxxxxxxxxxxxxxxxx> wrote:

vit...@xxxxxxxxxx wrote:

```
set fp [open "|openssls_server -accept 9009" r+]; # (you will need a
server.pem file in the current working directory).
```

When I connect to localhost:9009 with a web browser, I am able to use "gets \$fp" to retrieve the request from the web browser. However, when I send a response back to the browser, using "puts \$fp 'html data'", the browser will not receive the response until I close the connection (close \$fp). I tried "flush \$fp", that didn't work. Any ideas how to fix that?

Hello vitick,

I'm investigating this issue with the help of Alexandre Ferrieux. Currently, we believe that a race condition related to the SSL secured channel is causing the misbehaviour.

In my case, I use the Tcl tls1.5 extension. You appear not to use tls1.5 and nevertheless experience the same type of symptom as I do. Therefore, the problem may be inside theopenssllibrary which is a common factor in both our cases. However, I'm unsure about the SSL functionality that is employed at the client side of your setup.

In order to find this out, I would very much like to know about your "openssls_server" exercise above:

- which browser you have been using?
- which platform were you running on?

Because you made the client connect to localhost, I assume that server process and client process were both running under the same (non-windows) operating system. If that wasn't always the case, please

Re: openssl from tcl

let me know.

I'd be grateful for your cooperation,

Erik Leunissen

—

leunissen@ nl | Merge the left part of these two lines into one,
e. hccnet. | respecting a character's position in a line.

I have finally gotten back to the TLS code and there is definitely some kind of a race condition problem where often you would get lots of empty data in the beginning of the request. I have spent some time figuring a way out of it and the only way I could deal with it is to retry "gets sock" after each empty data until I get some good data. I don't like that solution at all because it might last indefinitely, so I had to set up a counter and stop after some number of [gets sock]. The counter has not exceeded 500 so far but you never know.

So, basically, I cannot use the following code:

```
while {[gets $sock data] >= 0} {...} --- which is the code used most of  
the time in examples on how to get the request data from the client.
```

I instead ended up using something like this:

```
set counter 0  
while {[eof $sock]} {  
  incr counter  
  if {$counter >= 1000} {break}  
  if {[gets sock data] >= 0} {  
    continue  
  }  
  ....  
  process the data  
  ....  
}
```

I hope this will get fixed, because timing out is not a good solution (not happen yet to me with the time out count set to 1000, but in some cases it probably will happen). I want to read and process every request correctly.

Of course, there is probably a better solution. That's where people like Alex Ferrieux often come to the rescue. ;)

.

Re: openssl from tcl