HTTPS SNI Monitoring How-to

Thomas Schockaert, 2014-23-03

Hi,

You may or may not already have encountered a webserver that requires the SNI (Server Name Indication) extension in order to know which website it needs to serve you. It comes down to "if you don't tell me what you want, I'll give you a default website or even simply reset the connection". A typical IIS8.5 will do this, even with the 'Require SNI' checkbox unchecked.

So you have your F5, with its HTTPS monitors. Those monitors do not yet support SNI, as they have no means of specifying the hostname you want to use for SNI.

In comes a litle script, that will do exactly that.

Here's a few quick steps to get you started:

1. Download the script from this article (it's posted on pastebin: http://pastebin.com/hQWnkbMg).
2. Import it under 'System' > 'File Management' > 'External Monitor Program File List'.
3. Create a monitor of type 'External' and select the script from the picklist under 'External Program'.
4. Add your specific variables (explanation below).
5. Add the monitor to a pool and you are good to go.

A quick explanation of the variables:

- METHOD (GET, POST, HEAD, OPTIONS, etc. - defaults to 'GET')
- URI ("the part after the hostname" - defaults to '/')
- HTTPSTATUS (the status code you want to receive from the server - defaults to '200')
- HOSTNAME (the hostname to be used for SNI and the Host Header - defaults to the IP of the node being targetted)
- TARGETIP and TARGETPORT (same functionality as the 'alias' fields in the original monitors - defaults to the IP of the node being targetted and port 443)
- DEBUG (set to 0 for nothing, set to 1 for logs in /var/log/ltm - defaults to '0')
- RECEIVESTRING (the string that needs to be present in the server response - default is empty, so not checked)
- HEADERX (replace the X by a number between 1 and 50, the value for this is a valid HTTP header line, i.e. "User-Agent: Mozilla" - no defaults)
- EXITSTATUS (set to 0 to make the monitor always mark te pool members as up; it's fairly useless, but hey... - defaults to 1)

There is a small thing you need to know though: due to the nature of the openssl binary (more specifically the s_client), we are presented with a "stdin redirection problem". The bottom line is that your F5 cannot be "slow" and by slow I mean that if it requires more than 3 seconds to pipe a string into openssl s_client, the script will always fail. This limit is defined in the variable "monitor_stdin_sleeptime" and defaults to '3'. You can set it to something else by adding a variable named 'STDIN_SLEEPTIME' and giving it a value. From my experience, anything above 3 stalls the "F5 script executer", anything below 2 is too fast for openssl to read the request from stdin, effectively sending nothing and thus yielding 'down'. When you enable debugging (DEBUG=1), you can see what I mean for yourself: no more log entries for the script when STDIN_SLEEPTIME is set too high; always down when you set it too low.

I hope this script is useful for you,

Kind regards,

Thomas Schockaert

F5 Networks, Inc. | 401 Elliot Avenue West, Seattle, WA 98119 | 888-882-4447 | f5.com

F5 Networks, Inc. Corporate Headquarters info@f5.com	F5 Networks Asia-Pacific apacinfo@f5.com	F5 Networks Ltd. Europe/Middle-East/Africa emeainfo@f5.com	F5 Networks Japan K.K. f5j-info@f5.com