

HTML Comment Scrubber

Use Case:

HTTP payload many times contains lots of comments and information which could be on sensitive nature. These comments would be added by the Web/App developers while writing the code and later are not removed from the HTML files which go to end clients. Hence removing comments is a dual benefit operation as it reduces the bandwidth taken while communicating with client and also ensures that no sensitive information is passed along.

F5 iRules:

```
when HTTP_REQUEST {
    # Don't allow data to be chunked
    if { [HTTP::version] eq "1.1" } {
        if { [HTTP::header is_keepalive] } {
            HTTP::header replace "Connection" "Keep-Alive"
        }
        HTTP::version "1.0"
    }
}

when HTTP_RESPONSE {
    if { [HTTP::header exists "Content-Length"] && [HTTP::header "Content-Length"]
    < 1000000} {
        set content_length [HTTP::header "Content-Length"]
    } else {
        set content_length 1000000
    }
    if { $content_length > 0 } {
        HTTP::collect $content_length
    }
}

when HTTP_RESPONSE_DATA {
    # Find the HTML comments
```

```

    set indices [regexp -all -inline -indices {<!--(?:[^-]|--[^-]-[^-])*?--\s*?>}
[HTTP::payload]]

    # Replace the comments with spaces in the response

    #log local0. "Indices: $indices"

    foreach idx $indices {
        set start [lindex $idx 0]
        set len [expr {[lindex $idx 1] - $start + 1}]
        log local0. "Start: $start, Len: $len"
        HTTP::payload replace $start $len [string repeat " " $len]
    }
}

```

NetScaler Solution:

```

add rewrite action replace_comment_action replace_all
"HTTP.RES.BODY(HTTP.RES.CONTENT_LENGTH)" "" "" -search regex(re/<!--+[\w|s|d]+-->/)
add rewrite policy replace_comment_pol
'HTTP.RES.BODY(HTTP.RES.CONTENT_LENGTH).REGEX_MATCH(re/<!--+[\w|s|d]+-->/)' replace_comment_action

```

NetScaler can look through the entire stream of response data through single policy using regex and can replace the comments using similar expression and replace_all action.