

Remove X-* Headers from Response

Use Case:

X-* headers are added by the intermediate proxies on the way while a request/response go from point A to point B. These headers can have useful information like preserving original Client IP where request originated from. But at times the overhead of carrying these headers can be significant and thus you will need ways to remove all such headers in single attempt.

F5 iRule:

```
when HTTP_RESPONSE {

    # Remove all instances of the Server header
    HTTP::header remove Server

    # Remove all headers starting with x-
    foreach header_name [HTTP::header names] {

        if {[string match -nocase x-* $header_name]}{

            HTTP::header remove $header_name
        }
    }
}
```

Netscaler Solution:

We are using Policy Extensions (custom written LUA function) to achieve this functionality

LUA Script:

```
local function split(str, pat)
  local t = {} -- NOTE: use {n = 0} in Lua-5.0
  local fpat = "(.*)" .. pat
  local last_end = 1
  local s, e, cap = str:find(fpat, 1)
  while s do
    if s ~= 1 or cap ~= "" then
      table.insert(t, cap)
    end
    last_end = e+1
    s, e, cap = str:find(fpat, last_end)
  end
  if last_end <= #str then
    cap = str:sub(last_end)
    table.insert(t, cap)
  end
  return t
end
```

```
function NSTEXT:remove_headers() : NSTEXT
```

```
  local out = ""
  local input = self
  local lines = split(input, "\\r\\n")
  local key,value
  for key,value in pairs(lines) do
    if ((value:find("x[-].*:") or (value:find("X[-].*:") or (value:lower():find("server")))) then
      --Do Nothing
      --Lua does not have continue or next statements, so keeping it blank
    else
      out = out .. value .. "\\r\\n"
    end
  end
  return(out:sub(0,-1))
```

```
end
```

The above script should be named as `remove_headers.lua` and placed in `/var/tmp` directory in NetScaler for the below configs to work.

Configuration:

```
import ns extension local:remove_headers.lua remove_headers
add ns extension remove_headers.lua
add rewrite action act1 replace http.res.full_header http.res.full_header.remove_headers
add rewrite policy pol1 true act1
Bind the above policy to response side Global or LB
```

This configuration invokes the LUA script while processing HTTP response coming from backend.