

Disable HTTP Processing For Selective HTTP Methods

Use Case:

HTTP is the most common layer 7 based communication protocol in industry today. Most of us use it for our everyday jobs to get done. HTTP RFC has defined common set of methods and operations to be used. While there are some special methods and operations which are not expected to be used in normal use cases. This example shows how you can disable HTTP processing and specific functions for these methods.

F5 iRules:

```
when CLIENT_ACCEPTED {
    # Enable HTTP processing for all requests by default
    HTTP::enable
}
when HTTP_REQUEST {
    # selectively disable HTTP processing for specific request methods
    switch [HTTP::method] {
        "MOVE" -
        "COPY" -
        "LOCK" -
        "UNLOCK" -
        "PROPFIND" -
        "PROPPATCH" -
        "MKCOL" { HTTP::disable }
    }
}
```

NetScaler Solution:

```
add patset method_filter
bind patset method_filter "MOVE"
bind patset method_filter "COPY"
bind patset method_filter "LOCK"
bind patset method_filter "UNLOCK"
bind patset method_filter "PROPFIND"
bind patset method_filter "PROPPATCH"
bind patset method_filter "MKCOL"

add rewrite policy dont_process HTTP.REQ.METHOD.EQUALS_ANY("method_filter")
NO Rewrite
```

This policy will make sure that NetScaler will not process HTTP requests coming in with one of these methods through Rewrite layer. You can use a Responder policy in case you want to redirect all such requests to a different host, or send error response or just reset the connection.