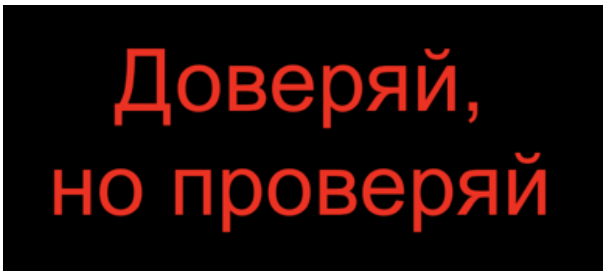
Verifying Local Traffic Policy and iRule Precedence

Jason Rahm, 2018-25-07

The BIG-IP platform has had Tcl-based iRules since the initial v9 release back in 2004. When version 11.4 released, the HTTP class was deprecated in favor of local traffic policies. The policies provide access to inspect and act on ingress and egress traffic much like iRules, but as they are TMOS-native, are more performant. Of course, iRules still provide greater flexibility, but as long as the needs are met in a policy, that should be the first choice.The interesting thing here then is what if I have a policy for the things it can handle, but an iRule for the things it can't? Which comes first, the chicken or the egg?



Knowledge article K16590 states clearly that policies are evaluated **before** iRules, so that clears it up. Article complete! Not so fast...

The point of this article isn't so much the result that policies come first, but to show you how to engage with the platform to discover some of these things for yourself, either because you just like to tinker, or like Ronald Reagan sold so well the Russian proverb to the American people back in the glory days of the 1980s, you prefer to *trust but verify*.

So how do go about verifying the policy runs before the iRule? The easy button is with logging, of course! And note that it's not so much that the policy runs first, but rather the events within the policy run before the events within the iRule. You'll see what I mean below. First, let's stub out one each of a dummy policy and iRule. Note you have more flow points of control in an iRule than you do in a policy. In the policy, I am not interested in this case in conditions, so there are none present, only the action of logging. Two final details. One, in the policy I have to log the event as it is not evident in the log entry created like it is with iRules. Two, I am using Tcl script in the policies in addition to the iRule so I can log the precision of the millisecond as the events trigger for logging the actual time.

Configuration Details

Local Traffic Policy

```
ltm policy http_event_order {
    last-modified 2018-07-24:12:00:12
    requires { http }
    rules {
        policy_rule_logging {
            actions {
                0 {
                    log
                    client-accepted
                    write
                    facility local0
                    message "tcl:Client Accepted Timestamp: [clock click -milliseconds]"
                    priority info
                }
                1 {
                    log
                    write
                    facility local0
                    message "tcl:Request Timestamp: [clock click -milliseconds]"
                    priority info
```

```
                }
                2 {
                    log
                    response
                    write
                    facility local0
                    message "tcl:Response Timestamp: [clock click -milliseconds]"
                    priority info
                }
            }
        }
    }
    status published
    strategy first-match
}
```

iRule

```
ltm rule http_event_order {
when CLIENT_ACCEPTED {
  log local0. "Timestamp: [clock clicks -milliseconds]"
}
when HTTP_REQUEST {
  log local0. "Timestamp: [clock clicks -milliseconds]"
}
when HTTP_REQUEST_RELEASE {
  log local0. "Timestamp: [clock clicks -milliseconds]"
}
when HTTP_RESPONSE {
  log local0. "Timestamp: [clock clicks -milliseconds]"
}
when HTTP_RESPONSE_RELEASE {
  log local0. "Timestamp: [clock clicks -milliseconds]"
}
}
```

And to finish the configuration off, we'll apply both the policy and the iRule to the virtual server.

```
ltm virtual testvip {
    destination 192.168.102.50:https
    ip-protocol tcp
    mask 255.255.255.255
    policies {
        http_event_order { }
    }
    pool testpool
    profiles {
        cssl {
            context clientside
        }
        http { }
        tcp { }
    }
    rules {
        http_event_order
    }
```

```
    source 0.0.0.0/0
    source-address-translation {
        type automap
    }
    translate-address enabled
    translate-port enabled
    vs-index 50
}
```

Verification Testing

Now that we have a configuration, all we have left to do is run a test! To keep everything down to a single request, I will not use a browser and keep the response to headers only so I don't clog up the log file. Curl to the rescue!

Test Log Entries

```
Jul 25 19:30:45 ltm3 info tmm1[13047]: [/Common/http_event_order/policy_rule_logging]: Client Accepte
Jul 25 19:30:45 ltm3 info tmm1[13047]: Rule /Common/http_event_order <CLIENT_ACCEPTED>: Timestamp: 15
Jul 25 19:30:45 ltm3 info tmm1[13047]: [/Common/http_event_order/policy_rule_logging]: Request Timest
Jul 25 19:30:45 ltm3 info tmm1[13047]: Rule /Common/http_event_order <HTTP_REQUEST>: Timestamp: 15325
Jul 25 19:30:45 ltm3 info tmm1[13047]: Rule /Common/http_event_order <HTTP_REQUEST_RELEASE>: Timestam
Jul 25 19:30:45 ltm3 info tmm1[13047]: [/Common/http_event_order/policy_rule_logging]: Response Times
Jul 25 19:30:45 ltm3 info tmm1[13047]: Rule /Common/http_event_order <HTTP_RESPONSE>: Timestamp: 1532
Jul 25 19:30:45 ltm3 info tmm1[13047]: Rule /Common/http_event_order <HTTP_RESPONSE_RELEASE>: Timesta
```

Verification complete! You can see that for the like events between policies and iRules, the policy event fires first every time. You can also see that you have additional points of access in iRules with the request/response ***release*** events. Now that we have verified the results of policies and iRules working together, some things to consider:

- Where you can, try to use a policy OVER using an iRule.
- Where you have to use iRules, it might make sense to put all logic in the iRule, rather than splitting the logic between them to avoid operational confusion.
- If performance is the primary driver and you have to use iRules and Policies together, plan and document well so everyone is aware of the breakdown of responsibility between them.
- Know that you can pass information from the policy actions to your iRules to enhance your solution and cut down on unnecessary logic on the iRule end. The POLICY namespace in iRules has several commands at your disposal.

Thanks to Nikhil Raj for asking the question!

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