ЧТО ТАКОЕ HTTP. ЧАСТЬ IX. ПОЛИТИКИ И ПРАВИЛА IRULES

Джейсон Рамм (Jason Rahm), архитектор решений, F5 Networks, США

В предыдущей статье цикла «Что такое HTTP?» мы рассмотрели методы оптимизации HTTP-трафика за счёт сжатия и кэширования. В этой статье мы несколько отклонимся от описания протокола, чтобы продемонстрировать, каким образом можно наладить взаимодействие с протоколом HTTP в BIG-IP: речь пойдёт о политиках и правилах iRules. На DevCentral есть множество публикаций о том, что такое политики и как их использовать, поэтому я не буду здесь повторять сказанное. Мы сосредоточимся на обсуждении возможностей каждого из подходов и рассмотрим пару общих примеров. Но пока мы не начали, сперва быстренько прочтите статьи Чейза Эббота (Chase Abbott) To iRule or not to iRule: Introduction to Local Traffic Policies («iRule или не iRule: введение в политики локального трафика») и Стива Маккарти (Steve McCarthy) LTM Policy («Политика LTM»). Я серьёзно: прочтите их прямо сейчас!

Пример 1: простое перенаправление HTTP-трафика

Иногда владелец приложения переносит свои ресурсы с одного места в другое внутри приложения: из-за изменения базовой платформы, либо для удобства навигации, либо же — из-за пересмотра карты сайта. Или же владелец приложения, возможно, просто хочет, чтобы рандомные URL-адреса появлялись в рекламе, чтобы упростить маркетинг и отслеживание необходимых параметров. Для перенаправления клиентов приложения на новые локации ресурсов, вы можете просто переписать URL перед их отправкой на сервер — и клиент этого даже не заметит, либо перенаправить их на новый ресурс, в результате чего клиент получит HTTP-ответ и отправит на BIG-IP второй запрос — уже обращаясь к «правильному» ресурсу.

Код состояния 300 используется «сервером» для того, чтобы сообщить клиенту о необходимых дополнительных действиях. Два основных кода перенаправления — это 301, сообщающий клиенту о том, что ресурс перенесен навсегда, и 302, сообщающий клиенту о временном переносе ресурса.

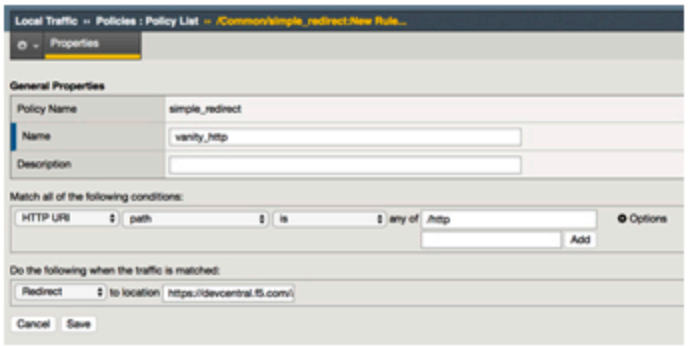
Решение на основе политики

Каждая политика включает в себя одно или несколько правил. В этой политике нам нужно лишь одно правило, где условием является наличие строки /http в составе пути, а действие заключается в перенаправлении обращений к страницам, соответствующим данному условию, на указанный URL.

```
ltm policy simple_redirect {
    controls { forwarding }
    last-modified 2017-12-14:23:22:51
    requires { http }
    rules {
        vanity_http {
            actions {
                0 {
```

```
                http-reply
                redirect
                location https://devcentral.f5.com/articles/sid/8311
            }
        }
        conditions {
            0 {
                http-uri
                path
                values { /http }
            }
        }
    }
  }
  status published
  strategy first-match
}
```

Написание политик в виде текста — задача не всегда приятная, но, как показано ниже, их можно настраивать и при помощи графического интерфейса. Обратите внимание, что в версии 12.1 и последующих версиях политику можно использовать лишь после ее публикации, а, при изменении политик, перед активацией каких-либо изменений, нужно сделать черновую копию.



Решение на основе iRules

Решение на основе iRule довольно короткое.

```
when HTTP_REQUEST {
  if { [HTTP::path] eq "/http" } {
    HTTP::redirect "https://devcentral.f5.com/articles/sid/8311"
  }
}
```

Но, несмотря на краткость и простоту, оно вряд ли будет работать так же эффективно, как политика. Тем не менее, благодаря простоте iRule, экономия производительности может не оправдать сложность использования политик, особенно если эту работу приходится делать с использованием командной строки. Следует также отметить, что, если требуется постоянное перенаправление, то использование правила iRule становится обязательным, так как обеспечить принудительную выдачу кода 301 при помощи одной лишь политики — невозможно. Для постоянного перенаправления вам придётся заменить команду HTTP::redirect на HTTP::respond и указать 301 с заголовком месторасположения.

```
when HTTP_REQUEST {
  if { [HTTP::path] eq "/http" } {
    HTTP::respond 301 Location "https://devcentral.f5.com/articles/sid/8311"
  }
}
```

Существует гибрид политики и iRule для перенаправления по коду 301, но если вам требуется iRule политики, то я бы порекомендовал использовать решение только на основе правила iRule.

Пример 2: замена X-Forwarded-For на IP-адрес клиента

Пользователи-злоумышленники могут попытаться фальсифицировать IP-клиентов в заголовке X-Forwarded-For (XFF), что может привести к неприятным последствиям. В этом примере мы уберём все заголовки XFF из запроса клиента и вставим туда новый.

Решение на основе политики

В этой политике нас не интересуют условия: нам нужно одно общее решение, заменяющее заголовок XFF, поэтому нам нужно прописать только действие, а не условие. Вы сразу увидите tcl: перед вызовом правил iRules [IP::client_addr]. Эти tcl-выражения поддерживаются в политиках для достижения компромисса между производительностью и дополнительной гибкостью.

```
ltm policy xff {
    last-modified 2017-12-15:00:39:26
    requires { http }
    rules {
        remove_replace {
            actions {
                0 {
                    http-header
                    replace
                    name X-Forwarded-For
                    value tcl:[IP::client_addr]
                }
            }
```

Этот вариант отлично работает, если заголовок XFF только один. Но что же делать, если их несколько? Посмотрим, что нам покажет wireshark по результатам запуска curl для того, чтобы проставить два заголовка с IP-адресами 172.16.31.2 и 172.16.31.3.

```
▽ Hypertext Transfer Protocol
  ▷ GET / HTTP/1.1\r\n
    Host: 192.168.102.62\r\n
    User-Agent: curl/7.54.0\r\n
    Accept: */*\r\n
    X-Forwarded-For: 172.16.31.2\r\n
    X-Forwarded-For: 192.168.102.1\r\n
    \r\n
```

Как видите, заменить удалось лишь один из двух заголовков XFF, то есть поставленную задачу мы не выполнили. Вместо этого нам нужно прописать удаление и добавление в политике вот таким образом:

```
ltm policy xff {
    last-modified 2017-12-15:00:35:35
    requires { http }
    rules {
        remove_replace {
            actions {
                0 {
                    http-header
                    remove
                    name X-Forwarded-For
                }
                1 {
                    http-header
                    insert
                    name X-Forwarded-For
                    value tcl:[IP::client_addr]
                }
            }
        }
    }
    status published
    strategy first-match
}
```

Как показывает wireshark по тем же двум IP-адресам, которые были вставлены в заголовки XFF из запроса curl, удаление и вставка позволили нам выполнить поставленную задачу.

```
▽ Hypertext Transfer Protocol
  ▷ GET / HTTP/1.1\r\n
    Host: 192.168.102.62\r\n
    User-Agent: curl/7.54.0\r\n
    Accept: */*\r\n
    X-Forwarded-For: 192.168.102.1\r\n
    \r\n
```

Решение на основе iRules

Здесь правило iRule очень похоже на политику тем, что мы снова удаляем и вставляем заголовки.

```
when HTTP_REQUEST {
  HTTP::header remove "X-Forwarded-For"
  HTTP::header insert "X-Forwarded-For" [IP::client_addr]
  # You could also use replace since it will insert if missing
  # HTTP::header replace "X-Forwarded-For" [IP::client_addr]
}
```

И снова коротко и удобно! Как и в приведенном выше примере решения на основе политики, использование в заголовке выражения HTTP::header remove удаляет все заголовки XFF, а не только последний, как это делает HTTP::header replace.

Решение на основе профиля

А теперь сделаем шаг назад и посмотрим, нужны ли нам iRule или политика в принципе. Помните, в пятой статье этого цикла упоминались настройки профиля HTTP для заголовка X-Forwarded-For? Там я отметил, что подключение этой настройки просто добавляет ещё один заголовок X-Forwarded-For, не заменяя уже имеющиеся. Но тут можно применить изящное обходное решение (очень похожее по смыслу на пример решения на основе политики, но полностью реализованное в профиле HTTP): мы добавим заголовок в поля remove и insert с указанными ниже значениями.

Request Header Erase	X-Forwarded-For	☑
Request Header Insert	X-Forwarded-For: [IP::client_addr]	☑

Помните, однако, что удалить и вставить можно только один заголовок, поэтому данное решение здесь, конечно, сработает, но если у вас больше параметров, то лучше применить политику или iRule.

Профили, политики и iRules... Так где же моя точка управления?

Эта проблема не относится к HTTP, но было бы ошибкой не уделить ей внимание, раз уж зашла речь о решениях. Соображения производительности играют важную роль в развертывании системы, но далеко не единственную

Есть ещё одна важная сфера, работу в которой требует тщательно спланировать - определение точек управления, удобных для группы эксплуатации. Если логика приложения разбросана по профилям, политикам и правилам iRules, а изменения приходится вносить не в одно приложение, а во все, то возникает значительный риск невозможности устранения проблем в том случае, если решения не оформлены документально. Да, логику вполне можно распределять, но в этом случае каждый должен знать о том, что XYZ регулируется в iRules, ABC — в политиках, а DEF — в профилях. В ином случае специалистам по эксплуатации придётся потратить кучу времени на то, чтобы разобраться, что к чему относится, и понять потоки приложений в BIG-IP.
Я говорю всё это, чтобы донести до читателей ту мысль, что iRules — это вполне приемлемое решение, если на те потребности, которые удовлетворяют политики, приходится лишь незначительная доля из общего количества прикладных служб, настроенных в BIG-IP. Вполне приемлемым решением может оказаться стандартизация политик с требованием к командам приложений перенести те потребности, которые политики не в состоянии решить, обратно в сами приложения. Классно, когда есть варианты!

Дополнительные ресурсы

- **Getting Started with iRules** («Правила iRules: с чего начать»)
- **iRule Recipe 1: Single URL Explicit Redirect** («iRule, рецепт 1: явное перенаправление одного URL»)
- **Codeshare: iRules & HTTP** («Код-шеринг: iRules и HTTP»)
- **Getting Started with Policies** (AskF5 Manual) («Политики: с чего начать. Пособие AskF5»)
- **LTM Policy Recipes I** («Политики управления локальным трафиком: рецепты, часть I»)
- **LTM Policy Recipes II** («Политики управления локальным трафиком: рецепты, часть II»)
- **Automatically Redirect http to https on a Virtual Server** (Ask F5 Solution) («Автоматическое перенаправление http на https на виртуальном сервере. Решение от Ask F5»)
- **Forwarded HTTP Extension Insertion - RFC 7239** (Codeshare iRule) («Вставка перенаправленного расширения HTTP — RFC 7239»)

Группа компаний БАКОТЕК – официальный дистрибьютор F5 Networks в Украине, Азербайджане, Республике Беларусь, Грузии, Армении и Молдове.
bakotech.com, f5@bakotetch.com, +38 044 273 33 33.

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

Americas
info@f5.com

Asia-Pacific
apacinfo@f5.com

Europe/Middle-East/Africa
emeainfo@f5.com

Japan
f5j-info@f5.com

©2017 F5 Networks, Inc. All rights reserved. F5, F5 Networks, and the F5 logo are trademarks of F5 Networks, Inc. in the U.S. and in certain other countries. Other F5 trademarks are identified at f5.com. Any other products, services, or company names referenced herein may be trademarks of their respective owners with no endorsement or affiliation, express or implied, claimed by F5. 0113