Packet Content Filter for BGP FlowSpec

draft-cui-idr-content-filter-flowspec-02

Yong Cui, Tsinghua University

Yujia Gao, Zhongguancun Laboratory

Susan Hares, Hickory Hill Consulting

IETF 120 Vancouver
Problem Statement

- Carrier network cannot defend well in face of some **volumetric DDoS attacks**, especially for over hundreds of Gbps or Tbps-level attacks

- Example: An **ACK Flood carpet-bombing attack** packet captured on the carrier network
  - Large packet and the destination port is 443, the packet content are **all zeros**

- Existing FlowSpec filter cannot match the traffic; Cleaning device is costly and limited in resources; Firewalls is slow to respond

  It is necessary to propose a **new type of FlowSpec filter** to implement the defense against **volumetric attacks with fixed packet characteristics**
Packet Content Filter

- **Component Encoding:** `<Type, Value>`

- **Value Encoding:**

  - **Ptype** = (4 bits) Packet type
    - 1 = IPv4
    - 2 = IPv6
  - **Otype** = (4 bits) Offset type
    - 0 = IP Header
    - 1 = IP Payload
    - 2 = TCP/UDP Payload
  - **Offset** = (2 octets) Offset from the beginning field specified in the Otype
  - **Content-Len** = (1 octet) Length of Content and Mask field
  - **Content** = (variable) string to be matched
  - **Mask** = (variable) string for masking (contains 0 or 1)
Packet Content Filter for FSv2

- NLRI Encoding:

  - TLVs:
    - NLRI length (2 octets)
    - Order (4 octets)
    - Identifier (4 octets)
    - FSv2 Filter type = 2
    - Length TLVs (2 octets)
    - FSv2 filters version
    - Sub-TLVs

- Ordering rules:
  a) Filters with a specific user order number would be ordered by the user order.
  b) Filters with same (or no) user order would be ordered by the default order:

  Content-length(↓) → Otype(↓) → Offset (↓) → Content(↑)  (↓ = higher value has higher precedence)
Development and Deployment

- Test bed
  - Test1: Effectiveness of the filter
  - Test2: Verify the ordering rules of the filter

- Development: OpenBGPD-8.3-portable, FRRouting-10.2-dev
  - Develop 1300+ lines of code
  - Add new filter rule definition and announcement function
  - Add announcement receive and analyze function
  - Add traffic handling function using netfilter

- Main Developer: Rui Xu and Yannan Hu
  - Github Project: https://github.com/Flowspec-extension/Packet-Content-Filter-Demo
Test 1 - Effectiveness of Filter

- Objective: Using packet content filters to defend network layer DDoS attacks
- Traffic generator:
  - Business traffic: HTTP (Web request)
  - Attack traffic: UDP Flood, ICMP Flood, ACK Flood
- Effectively defend simple network layer volumetric attack in network device
- Reduce the defense costs of cleaning devices
Test 1 - Effectiveness of Filter

- **Objective:** Using packet content filters to defend application layer DDoS attacks

- **Traffic generator:**
  - Business traffic: HTTP (Web request)
  - Attack traffic: HTTP Flood

  **Effectively defend** 25-55% HTTP Flood attack

  **Combined with cleaning device can achieve full traffic defense**
Test 2 – Component Ordering

- Situation 1: Simultaneous Match Conflict – Substring

| Traffic1: UDP Flood, otype 2, offset 0, payload 5858 |
| Traffic2: UDP Flood, otype 2, offset 0, payload 58585a5a |

Rule1: to 192.168.234.4/32 payload 1 2 0 2 0x5858 0xffff - Rate limiting
Rule2: to 192.168.234.4/32 payload 1 2 0 4 0x58585a5a 0xffffffff - Discard

<table>
<thead>
<tr>
<th>Order</th>
<th>Number of arrived packets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Traffic1</td>
<td>Traffic2</td>
</tr>
<tr>
<td>Rule1、Rule2</td>
<td>260</td>
</tr>
<tr>
<td>Rule2、Rule1</td>
<td>500</td>
</tr>
</tbody>
</table>

Rule1 matches traffic1 and traffic2
Rule2 matches traffic2

Ordering: \texttt{content-length(↓)→otype(↓)→offset (↓) →content(↑)}

Component with larger content-length take precedence
Test 2 – Component Ordering

• Situation 2: Simultaneous Match Conflict – Different Otype

Traffic1: UDP Flood, otype 2, offset 0, payload 5858
Traffic2: ICMP Flood, otype 1, offset 8, payload 5858

Rule1: to 192.168.234.4/32 payload 1 2 0 2 0\times5858 0xffff - Rate limiting
Rule2: to 192.168.234.4/32 payload 1 1 8 4 0\times5858 0xffff - Discard

<table>
<thead>
<tr>
<th>Order</th>
<th>Number of arrived packets</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Traffic1</td>
</tr>
<tr>
<td>Rule1, Rule2</td>
<td>500</td>
</tr>
<tr>
<td>Rule2, Rule1</td>
<td>0</td>
</tr>
</tbody>
</table>

Rule1 matches traffic1
Rule2 matches traffic1 and traffic2

Ordering: content-length(↓) → otype(↓) → offset (↓) → content(↑)

Component with larger otype take precedence
Test 2 – Component Ordering

• Situation 3: Simultaneous Match Conflict – Different Offset

Traffic: UDP Flood, otype 2, offset 0, payload 5858; offset 20, payload 4343

Rule1: to 192.168.234.4/32 payload 1 2 0 2 0x5858 0xffff - Rate limiting
Rule2: to 192.168.234.4/32 payload 1 2 20 2 0x4343 0xffff - Discard

<table>
<thead>
<tr>
<th>Order</th>
<th>Number of arrived packets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Traffic</td>
<td></td>
</tr>
<tr>
<td>Rule1, Rule2</td>
<td>500</td>
</tr>
<tr>
<td>Rule2, Rule1</td>
<td>0</td>
</tr>
</tbody>
</table>

Rule1 and Rule2 match traffic

Ordering: content-length(↓)→otype(↓)→**offset (↓)**→content(↑)
Component with larger offset take precedence
Summary & Next Steps

• Summary of packet content filter

  - Implement simple volumetric DDoS attack defense in network device, reducing the pressure and cost of carrier network defense
  - Provide FSv1 and FSv2 compatible definition formats
  - Provide ordering rules to handle typical component conflicts
  - Implement the filter on open-source projects OpenBGPD and FRRouting

• Next Steps

  - Any questions and comments are welcomed
  - Actively participate in the FSv2 work
  - Validate filters in hardware devices and carrier networks
Thanks!

Contact information: gaoyj@zgclab.edu.cn