Packet Content Filter for BGP FlowSpec

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

Yong Cui, Tsinghua University

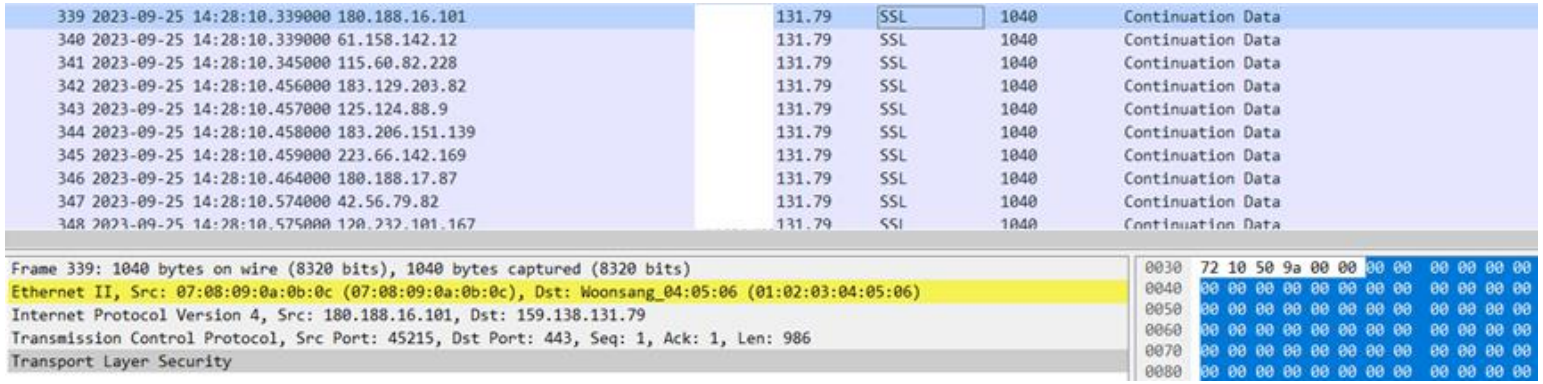
Yujia Gao, Zhongguancun Laboratory

Susan Hares, Hickory Hill Consulting

IETF 120 Vancouver

Problem Statement

- Carrier network cannot defense 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

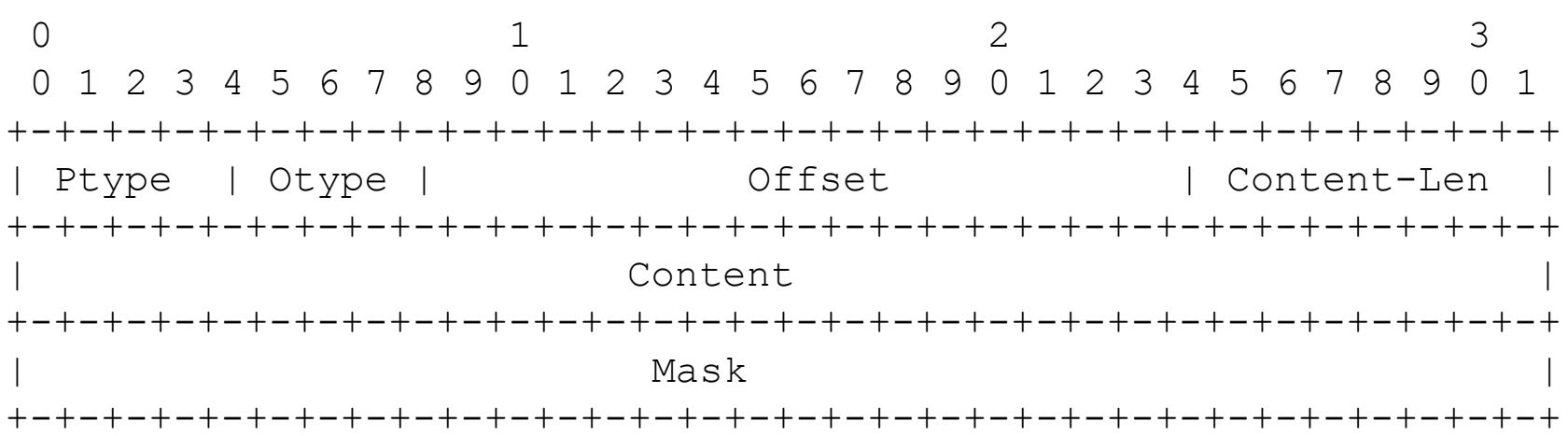


- Existing FlowSpec filter cannot match the traffic; Cleaning device is costly and limited in resources; Firewalls is slow to respond

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

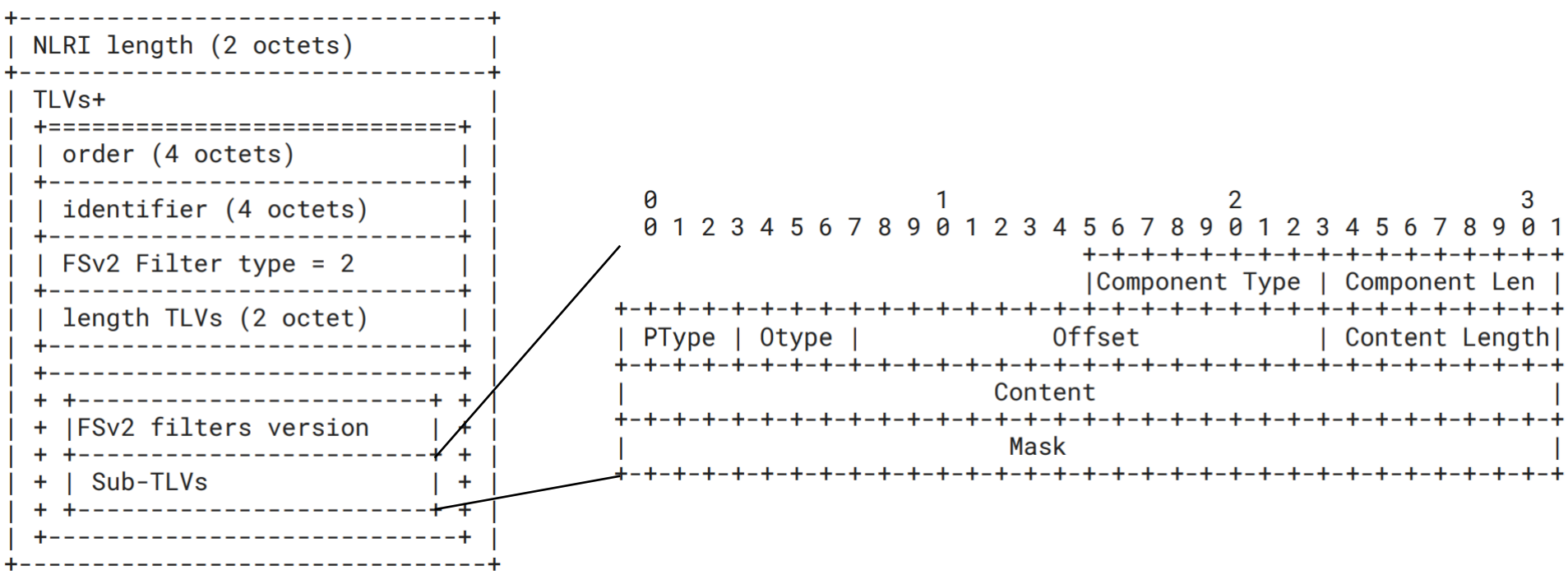
- Component Encoding: <Type, Value>
- Value Encoding:



- **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:



- Ordering rules:

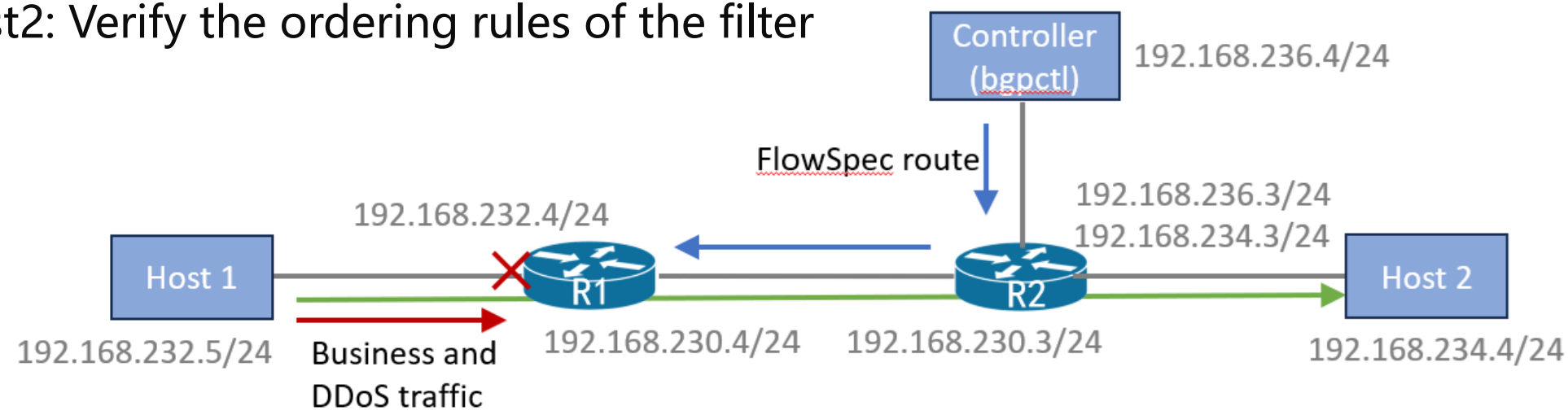
a) Filters with a specific user order number would be ordered by the user order
b) Filters with same (or no) user order would be ordered by the default order:

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

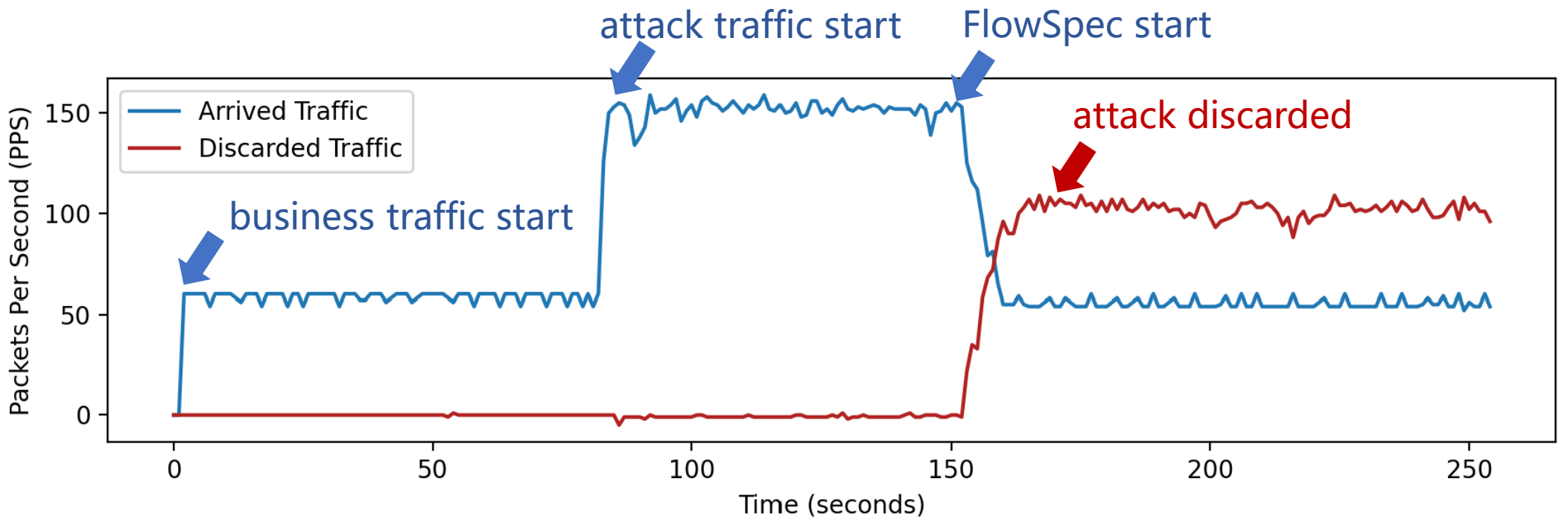


- Development：OpenBGPD-8.3-portable, FRRouting-10.2-dev
  - Develop 1300+ lines of code
  - Add new filter rule definition and announcement function
  - Add announcement receive and analyze function
  - Add traffic handling function using netfilter
- Main Developer: Rui Xu and Yannan Hu
  - Github Project: 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

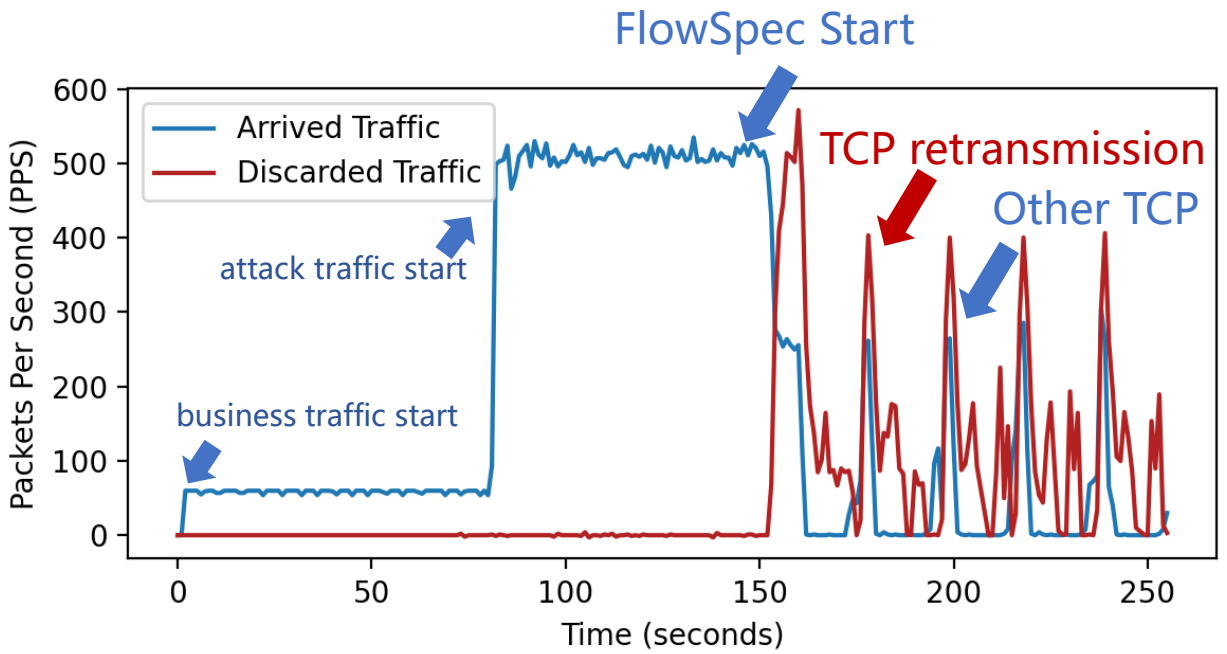
Client

	TCP	74 47082 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TS
	TCP	74 80 → 47082 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 S
192.168.234.4	TCP	66 47082 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=363282461
192.168.234.4	HTTP	434 GET /aaaaaaaaV6cza HTTP/1.1
192.168.234.4	TCP	66 47082 → 80 [FIN, ACK] Seq=369 Ack=1 Win=64256 Len=0 TSval=36
192.168.232.5	TCP	78 [TCP Window Update] 80 → 47082 [ACK] Seq=1 Ack=1 Win=65280 L
192.168.234.4	TCP	434 [TCP Retransmission] 47082 → 80 [PSH, ACK] Seq=1 Ack=1 Win=6
192.168.234.4	TCP	434 [TCP Retransmission] 47082 → 80 [PSH, ACK] Seq=1 Ack=1 Win=6
192.168.234.4	TCP	434 [TCP Retransmission] 47082 → 80 [PSH, ACK] Seq=1 Ack=1 Win=6
192.168.232.5	TCP	78 80 → 47082 [FIN, ACK] Seq=1 Ack=1 Win=65280 Len=0 TSval=1102
192.168.234.4	TCP	66 47082 → 80 [ACK] Seq=370 Ack=2 Win=64256 Len=0 TSval=3632854
192.168.232.5	TCP	78 80 → 47082 [RST, ACK] Seq=2 Ack=1 Win=65280 Len=0 TSval=1102

HTTP get request and TCP retransmission can be discarded

Web Server

	TCP	74 49560 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=146
	TCP	74 80 → 49560 [SYN, ACK] Seq=0 Ack=1 Win=65160 Le
192.168.234.4	TCP	66 49560 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 T
192.168.234.4	TCP	66 [TCP Previous segment not captured] 49560 → 80
192.168.232.5	TCP	78 [TCP Window Update] 80 → 49560 [ACK] Seq=1 Ack
192.168.232.5	TCP	78 80 → 49560 [FIN, ACK] Seq=1 Ack=1 Win=65280 Le
192.168.234.4	TCP	66 49560 → 80 [ACK] Seq=370 Ack=2 Win=64256 Len=0
192.168.232.5	TCP	78 80 → 49560 [RST, ACK] Seq=2 Ack=1 Win=65280 Le



- 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

	Order	Number of arrived packets	
		Traffic1	Traffic2
	Rule1、Rule2	260	240
√	Rule2、Rule1	500	0

Rule1 matches traffic1 and traffic2
Rule2 matches traffic2

Ordering: 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

UDP and ICMP headers are both 8 octets

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 1 8 4 0x5858 0xffff - Discard

	Order	Number of arrived packets	
		Traffic1	Traffic2
√	Rule1、Rule2	500	0
	Rule2、Rule1	0	0

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

	Order	Number of arrived packets
		Traffic
	Rule1、Rule2	500
√	Rule2、Rule1	0

→ 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