Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: December 20, 2018

A. Khare, Ed.
J. Scudder, Ed.
Juniper Networks, Inc.

June 18, 2018

BGP FlowSpec Payload Matching
draft-khare-idr-bgp-flowspec-payload-match-00

Abstract

The rise in frequency, volume, and pernicious effects of DDoS attacks has elevated them from fare for the specialist to generalist press. Numerous reports detail the taxonomy of DDoS types, the varying motivations of their attackers, as well as the resulting business and reputation loss of their targets.

BGP FlowSpec (RFC 5575, "Dissemination of Flow Specification Rules") can be used to rapidly disseminate filters that thwart attacks, being particularly effective against the volumetric type. Operators can use existing FlowSpec components to match on pre-defined packet header fields. However recent enhancements to forwarding plane filter implementations allow matches at arbitary locations within the packet header and, to some extent, the payload. This capability can be used to detect highly amplified attacks, whose attack signature remains relatively constant.

We define a new FlowSpec component, "Flexible Match Conditions", with similar matching semantics to those of existing components. This component will allow the operator to define bounded match conditions using offsets and bitmasks.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

The rise in frequency, volume, and pernicious effects of distributed denial of service ("DDoS") attacks has elevated them from fare for specialist to generalist press. Numerous reports detail the taxonomy of DDoS types, the varying motivations of their attackers, as well as the resulting business and reputation loss of their targets.

BGP FlowSpec [RFC5575] can be used to rapidly disseminate filters that thwart attacks, being particularly effective against the volumetric type. Operators can use existing FlowSpec components to match on pre-defined packet header fields. However recent enhancements to forwarding plane filter implementations allow matches

at arbitary locations within the packet header and, to some extent, the payload. This capability can be used to detect highly amplified attacks, whose attack signature remains relatively constant.

We define a new FlowSpec component, "Flexible Match Conditions", with similar matching semantics to those of existing components. This component will allow the operator to define bounded match conditions using offsets and bitmasks.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119] .

2. Flexible Match Conditions

We define a new FlowSpec component, Type TBD, named "Flexible Match Conditions".

Encoding: <type (1 octet), op, value>

It contains a single {operator, value} tuple that is used to match packets according to the rules given below.

2.1. Operator

The operator field is encoded as:

```
                      0                   1
                      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
                     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                     |v| a |u  |bit o|  byte offset  |
                     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

v - Type of value being matched, string comparison (Section 2.2.1) if this bit is set, and numeric range (Section 2.2.2) if unset.

a - Anchor. A 2-bit unsigned integer whose value indicates where in the packet the match should start. To avoid ambiguity with tunneled packets, the match SHOULD be anchored at the outermost header. An example is given below (Section 2.3) .

Value	Symbolic Name	Match start
0	d	Layer 2 (d)ata-link layer Ethernet header
1	i	Layer 3 (I)Pv4/IPv6 header
2	t	Layer 4 TCP/UDP (t)ransport header
3	p	Layer 4-specific (p)rotocol-specific payload

Anchor Field Values

u - Reserved. MUST be set to 0. MUST be ignored on receipt.

bit offset - A 3-bit unsigned integer indicating how many bits to ignore, following the byte offset.

byte offset - An 8-bit unsigned integer indicating how many bytes to ignore, after the match start as determined by the first selected anchor bit.

2.2. Value

The operator field indicates where to start matching; by contrast, the value operand indicates what to match and where to stop matching. The value operand MUST be of the type indicated by the 'v' bit, as signaled in the operator. As a result it can take on one of two forms - string vs. numeric range comparison.

The length of the numeric range is constant. It uses two 64-bit fields. A string comparison uses two 128-bit fields. Its length field indicates the extent of how much of the prefix and mask fields to use in the AND operation. This is deemed sufficient for stateless inspection and practical for efficient hardware forwarding plane implementations.

2.2.1. String Comparison

```
      0                   1                   2                   3
      0                   1                   2                   3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |    len     |                      reserved                    |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                                                               |
     +                                                               +
     |                                                               |
     +                             prefix                            +
     |                                                               |
     +                                                               +
     |                                                               |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                                                               |
     +                                                               +
     |                                                               |
     +                              mask                             +
     |                                                               |
     +                                                               +
     |                                                               |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

len - Indicates the number of corresponding bits in the prefix and mask fields to read. This length field is interpreted as (len + 1 << 1). This allows even unsigned values ranging from 2-128.

prefix - Provides a bit string to be matched. The prefix and mask fields are bitwise AND'ed to create a resulting pattern. The number of bits used in the AND operation are indicated by the preceding length field.

mask - Paired with the prefix field to create a bit string match. An unset bit is treated as a 'do not care' bit in the corresponding position in the prefix field. When a bit is set in the mask, the value of the bit in the corresponding location in the prefix field must match exactly.

Implementations MUST only extract the number of bits from the prefix and mask fields as indicated by the preceding length field.

2.2.2. Numeric Range Comparison

```
      0                   1                   2                   3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                                                               |
     +                              low                              +
     |                                                               |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                                                               |
     +                              high                             +
     |                                                               |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

low - The low value of the desired inclusive numeric range. This value MUST be numerically lower than the high value.

high - The high value of the desired inclusive numeric range. This value MUST be numerically higher than the low value.

2.3. Example

As an example, consider that the canonical Virtual eXtensible Local Area Network (VXLAN) [RFC7348] packet has the following headers:

- Outer Ethernet Header: Source MAC address of the originating VXLAN Tunnel End Point (VTEP).

- Outer IPv4/IPv6 Header: Source IP address of the originating VXLAN Tunnel End Point (VTEP).

- Outer UDP Header: Random source port used to generate entropy for load balancing, and destined to the IANA-assigned VXLAN port 4789.

- VXLAN Header: Used to identify a specific VXLAN overlay network.

- Inner Ethernet Header and payload: Original MAC frame being encapsulated.

The following table outlines where the match would start based on the anchor setting:

Anchor value	Match start
d	Outer Ethernet Header
i	Outer IPv4/IPv6 Header
t	Outer UDP Header
p	VXLAN Header

3. Error Handling

Malicious, misbehaving, or misunderstanding implementations could advertise semantically incorrect values. Care must be taken to minimize fallout from attempting to parse such data. Any well-behaved implementation SHOULD verify that the minimum packet length undergoing a match equals (match start header length + byte offset + bit offset + value length).

4. Security Considerations

This document introduces no additional security considerations beyond those already covered in [RFC5575] .

5. IANA Considerations

IANA is requested to assign a type from the First Come First Served range of the "Flow Spec Component Types" registry:

Type Value	Name	Reference
TBD	Flexible Match Conditions	this document

6. Acknowledgements

Thanks to Rafal Jan Szarecki, Sudipto Nandi, and Jeff Haas for their valuable comments and suggestions on this document.

7. Contributors

Ron Bonica
Juniper Networks, Inc.
2251 Corporate Park Drive
Herndon, VA 20171
US

Email: rbonica@juniper.net

8. References

8.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC5575] Marques, P., Sheth, N., Raszuk, R., Greene, B., Mauch, J., and D. McPherson, "Dissemination of Flow Specification Rules", RFC 5575, DOI 10.17487/RFC5575, August 2009, <https://www.rfc-editor.org/info/rfc5575>.

8.2. Informative References

[RFC7348] Mahalingam, M., Dutt, D., Duda, K., Agarwal, P., Kreeger, L., Sridhar, T., Bursell, M., and C. Wright, "Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks", RFC 7348, DOI 10.17487/RFC7348, August 2014, <https://www.rfc-editor.org/info/rfc7348>.

Authors' Addresses

Anurag Khare (editor)
Juniper Networks, Inc.
2251 Corporate Park Drive
Herndon, Virginia 20171
USA

Email: anuragk@juniper.net

John Scudder (editor)
Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, CA 94089
US

Email: jgs@juniper.net