Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: March 4, 2021

A. Khare, Ed.
Ciena Corporation
P. Bergeon, Ed.
Nokia
J. Scudder
Juniper Networks, Inc.
L. Jalil
Verizon
K. Kasavchenko
NetScout
August 31, 2020

BGP FlowSpec Payload Matching
draft-khare-idr-bgp-flowspec-payload-match-07

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 attacks, the varying motivations of their attackers, as well as the resulting impact for their targets ranging from internet or business services to network infrastrutures.

BGP FlowSpec (RFC 5575, "Dissemination of Flow Specification Rules") can be used to rapidly disseminate filtering rules to mitigate (distributed) denial-of-service (DoS) attacks. Operators can use existing FlowSpec components to match typical n-tuple criteria in pre-defined packet header fields such as IP protocol, IP prefix or port number. Recent enhancements to IP Router forwarding plane filter implementations also allow matches at arbitrary locations within the packet header or payload. This capability can be used to essentially match a signature for the attack traffic and can be combined with traditional n-tuple filter criteria to mitigate volumetric DDoS attacks and reduce false positive to a minimum.

To support this new filtering capability 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 a new match condition using a combination of offset and pattern values.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Copyright Notice

Table of Contents

1. Introduction 3
2. Definitions of Terms Used in This Memo 3
3. Motivation 4
3.1. Machine analysis of DDoS attacks 4
3.1.1. Matching based on payload 4
3.1.2. Matching based on any protocol header field or across fields 5
3.2. Tunneled traffic 5
3.3. Non-IP traffic 5
4. Specification 6
4.1. Offset-type 6
4.2. Offset-value 7
4.3. Pattern-type 7
4.4. Pattern-value 8

4.4.1. Bitmask match 8
4.4.2. Regular expression string match 8
5. Flexible Match Conditions boundaries and additional considerations 8
6. Error Handling 9

7. Security Considerations 9
8. IANA Considerations 9
9. Acknowledgements 10
10. References 10
10.1. Normative References 10
10.2. Informative References 10
10.3. URIs 11
Authors' Addresses 11

1. Introduction

BGP FlowSpec [RFC5575] can be used to rapidly disseminate filtering rules to mitigate (distributed) denial-of-service (DoS) attacks. Operators can use existing FlowSpec components to match typical n-tuple criteria in pre-defined packet header fields such as IP protocol, IP prefix and port number.

Recent enhancements to IP Router forwarding plane filter implementations also allow matches at arbitrary locations within the packet header or payload. This capability can be used to essentially match a signature for the attack traffic and can be combined with traditional n-tuple filter criteria to mitigate volumetric DDoS attacks and reduce false positive to a minimum.

To support this new filtering capability 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 a new match condition using a combination of offset and pattern values.

2. Definitions of Terms Used in This Memo

AFI - Address Family Identifier.

SAFI - Subsequent Address Family Identifier.

NLRI - Network Layer Reachability Information.

Flow specification controller - BGP speaker sending the flow specification rules to the IP edge routers (e.g. DDoS controllers).

Maximum Readable Length - The packet length in bits that a forwarding implementation can parse and make available for filtering. Abbreviated as MRL.

Maximum Pattern Length - The pattern length in bits that a forwarding implementation can match against the packet header or payload. Abbreviated as MPL.

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].

3. Motivation

BGP FlowSpec couples both the advertisement of NLRI-specific match conditions, as well as the forwarding instance to which the filter is attached. This makes sense since BGP FlowSpec advertisements are most commonly generated, or at least verified, by human operators. The operator finds it intuitive to configure match conditions as human-readable values, native to each address family.

It is much friendlier, for instance, to define a filter that matches a source address of 192.168.1.1/32, than it is to work with the equivalent binary representation of that IPv4 address. Further, it is easier to use field names such as 'IPv4 source address' as part of the match condition, than it is to demarc that field using byte and bit offsets.

However, there are a number of use cases that benefit from the latter, more machine-readable approach.

3.1. Machine analysis of DDoS attacks

Volumetric DDoS attacks can severely impact services and network operator infrastrutures but are also easily mitigated once identified. The challenge lies in fishing out a generally unvarying attack signature from a data stream. Machine analysis can be particularly useful here given the size of input involved in order to identify a pattern within the attack traffic flows.

Below we illustrate the need for the suggested approach with two use cases.

3.1.1. Matching based on payload

Volumetric DDoS attacks can either directly send traffic to a target or use reflection/amplification protocols to overload that target.

Reflection/amplification attacks are often identified by the UDP source port of a service that reflects and amplifies the attack traffic. However, blocking traffic based on source port can lead to further service interruption and eventually complete the attack

especially in case of essential protocols such as NTP. There alo exist DDoS attack methodologies such as SSDP Diffraction or Bittorent amplification where values in most of layer 3 and layer 4 header fields, including source and destination UDP ports, are varied. That makes it challenging to mitigate based on existing Flow Specification components. At the same time these attacks often have a constant pattern in payload. Using the pattern in payload as a matching criteria would help in mitigating such DDoS attacks.

Direct attacks may also use a constant pattern in payload which can be used as a match criteria in filtering rules.

3.1.2. Matching based on any protocol header field or across fields

BGP FlowSpec [RFC5575] defines 12 Flow Specification component types that can be used to match traffic. However, a DDoS attack might result in illegitimate traffic with a specific pattern in a layer 3 or layer 4 header, and this pattern may not have a respective FlowSpec component type defined. Examples of this are the Time to Live field of IP header or Window field of TCP header. The flexible match patterns defined in this document avoid extending BGP FlowSpec [RFC5575] with all theoretically possible header fields and also

allow matching accross fields for any bitmask combinations.

3.2. Tunneled traffic

Tunnels continue to proliferate due to the benefits they provide. They can help reduce state in the underlay network. Tunnels allow bypassing routing decisions of the transit network. Traffic that is tunneled is often done so to obscure or secure. Common tunnel types include IPsec [RFC4301], Generic Routing Encapsulation (GRE) [RFC2890], et al.

By definition, transit nodes that are not the endpoints of the tunnel hold no attendant control or management plane state. These very qualities make it challenging to filter tunneled traffic at non-endpoints and it is usually infeasible to filter based on the content of this passenger protocol's header since BGP FlowSpec does not provide the operator a way to address arbitrary locations within a packet.

3.3. Non-IP traffic

Not all traffic is forwarded as IP packets. Layer 2 services abound, including flavors of BGP-signaled Ethernet VPNs such as BGP-EVPN, BGP-VPLS, FEC 129 VPWS (LDP-signaled VPWS with BGP Auto-Discovery).

Ongoing efforts such as [I-D.ietf-idr-flowspec-l2vpn] offer one approach, which is to add layer 2 fields as additional match conditions. This may suffice if a filter needs to be applied only to layer 2, or only to layer 3 header fields.

4. Specification

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

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

The length is a one octet unsigned interger field that contains the length of the Value field in octets.

The Value field itself is encoded using offset-type, offset-value, pattern-type and pattern-value.

Encoding: <offset-type (4 bits), offset-value (2 octets), pattern-type (4 bits), pattern-value (variable)>

The value field is 0 padded for byte alignement.

4.1. Offset-type

The combination of offset-type and offset-value defines where the match should begin for the pattern-value. This document defines the following offset types:

Value	Offset Type
0	Layer 3 - IP Header
1	Layer 4 - IP Header Data
2	Payload - TCP/UDP Data

Offset Types

The offset-type 0 for 'layer 3' is defined as the start of the IP header.

The offset-type 1 for 'layer 4' is defined as the start of the data portion of the IP header after the IP options.

The offset-type 2 for 'payload' is defined as start of the TCP or UDP data. For TCP, the offset-type payload represents the beginning of the TCP data after any TCP options. Note that Flow Specification

NLRI using the Flexible Match Condition component with offset-type 2 will result in not matching the pattern value in this component in case of non-first fragmented packet or in case it is combined with component type 2 IP Protocol other than 6 (TCP) and 17 (UDP).

4.2. Offset-value

The offset-value is a 2 octets unsigned integer field defining the

number of bytes to ignore from the earlier offset-type to match the pattern value.

Examples:

- The combination of offset-type 0 (Layer 3) and offset-value 0 defines an offset at the very beginning of the IP header.

- The combination of offset-type 1 (Layer 4) and offset-value 2 defines an offset two bytes after the beginning of the data portion of the IP header (after any IP options). Example, in the case of a UDP packet, this offset defines the beginning of the destination port header field.

- The combination of offset-type 2 (Payload) and offset-value 10 defines an offset ten bytes after the beginning of the TCP/UDP data payload.

4.3. Pattern-type

The pattern-type defines how the pattern value is matched. The following pattern-types are defined:

Value	Pattern Type
0	Bitmask match
1	POSIX Regular expression (regex) string match
2	PCRE Regular expression (regex) string match

Pattern Types

Pattern-type 0 MUST be implemented.

Pattern-type 1 and 2 for regular expressions are typically dedicated to hardware-accelerated and software-only forwarding planes or appliances that may be able to filter on more complex criteria. There is a plethora of regular expression engines and their supported flavor. The two flavors introduced in this document are:

o POSIX regular expression string match: This type refers to

extended regular expression (ERE) as defined by [IEEE.1003-2.1992].

- PCRE regular expression string match: This type refers to Perl compatible regular expression as defined by PCRE documentation [1].

4.4. Pattern-value

4.4.1. Bitmask match

If the pattern-type bitmask is selected, the pattern-value is encoded as {prefix, mask}, of equal length.

prefix - Provides a bit string to be matched. The prefix and mask fields are bitwise AND'ed to create a resulting pattern.

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.

4.4.2. Regular expression string match

If a regular expression pattern-type is selected, the pattern-value is encoded following the appropriate regular expression string match.

5. Flexible Match Conditions boundaries and additional considerations

The beginning of the match boundary is aligned with the FlowSpec AFI/SAFI to which the flexible match rule belongs. For instance, with FlowSpec for IPv4 traffic, the smallest offset can only start at the first bit of the IPv4 header.

The end of the match boundary MUST be the lesser of either the last bit in a packet or the Maximum Readable Length (Section 2) that a forwarding implementation can parse from a packet and make available for filtering. As the MRL will be implementation-dependent, it needs to be known to the Flow Specification controller. That can be communicated out-of-band via configuration or signaled using future BGP or IGP extensions.

The Maximum Pattern Length (Section 2) for the pattern-value can also be forwarding implementation dependant and may need to be known to the Flow Specification controller or communicated out-of-band.

It is not required that all nodes in a filtering domain have a common or minimum MRL and MPL. This does not remove the need for a Flow Specification controller to take MRL and MPL into account when creating flexible filters. This can be useful if the Flow Specification controller does not have direct BGP peering with all FlowSpec enforcers and may not receive a BGP Notification if it advertises a flexible match that exceeds the MRL or MPL of a given node.

6. 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 from the offset + pattern-value length).

7. Security Considerations

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

8. 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

Reference: this document

Registry Owner/Change Controller: IESG

Registration procedures:

Range	Registration Procedures
0-127	IETF Review
128-249	First Come First Served
250-254	Experimental

255	Reserved

Note: a separate "owner" column is not provided because the owner of all registrations, once made, is "IESG".

9. Acknowledgements

We wish to thank Michael Gallagher, Ron Bonica, Jeff Haas, Sudipto Nandi, Brian St Pierre and Rafal Jan Szarecki for their valuable comments and suggestions on this document.

10. References

10.1. 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>.

10.2. Informative References

[I-D.ietf-idr-flowspec-l2vpn] Weiguo, H., Eastlake, D., Litkowski, S., and S. Zhuang, "BGP Dissemination of L2 Flow Specification Rules", draft-ietf-idr-flowspec-l2vpn-15 (work in progress), May 2020.

[IEEE.1003-2.1992] Institute of Electrical and Electronics Engineers, "Information Technology - Portable Operating System Interface (POSIX) - Part 2: Shell and Utilities (Vol. 1)", IEEE Standard 1003.2, 1992.

[RFC2890] Dommety, G., "Key and Sequence Number Extensions to GRE", RFC 2890, DOI 10.17487/RFC2890, September 2000, <https://www.rfc-editor.org/info/rfc2890>.

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, DOI 10.17487/RFC4301, December 2005, <https://www.rfc-editor.org/info/rfc4301>.

10.3. URIs

[1] https://www.pcre.org/original/pcre.txt

Authors' Addresses

Anurag Khare (editor)
Ciena Corporation

Email: ak@ciena.com

Philippe Bergeon (editor)
Nokia
600 March Road
Ottawa, Ontario K2K2E6
CA

Email: philippe.bergeon@nokia.com

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

Email: jgs@juniper.net

Luay Jalil
Verizon

Email: luay.jalil@one.verizon.com

Kirill Kasavchenko
NetScout

Email: Kirill.Kasavchenko@netscout.com