DOTS
Internet Draft
Intended status: Standard Track
Expires: Nov 2016

T. Fu
Huawei
D. Zhang
Alibaba
L. Xia
M. Li
Huawei

June 14, 2016

IPFIX IE Extensions for DDoS Attack Detection
draft-fu-dots-ipfix-extension-01.txt

Status of this Memo

This Internet-Draft will expire on December 14, 201616.

Abstract

DDoS Open Threat Signaling (DOTS) Working Group is for developing the standard signaling mechanisms, together with the DDoS related telemetry and threat handling requests and data transmitted by them used in DDoS problem space. Although IP Flow Information Export (IPFIX), Packet Sampling (PSAMP), and Packet Selection methods are useful for network security inspection, there are still some gaps existing to identify some categories of DDoS attacks. To fill in the gaps, this document describes the connection sampling mechanism and explains why it is needed for detecting DDoS attacks. It also defines several new IPFIX Information Elements (IEs). Then, it presents some examples to show how to use these new IPFIX IEs together with the existing IPFIX IEs to detect specific DDoS attacks.

Table of Contents

1. Introduction 3
2. Conventions used in this document..... 4
2.1. Terminology 4
3. Connection Sampling and new IEs..... 5
3.1. Packet Sampling vs Connection Sampling 5
3.2. Use Cases for New IEs..... 6
3.2.1. Upstream/Downstream Counters 6
3.2.2. Fragment statistic..... 7
3.2.3. Response Time Calculation 8
3.2.4. Symptoms of Exceptions..... 8
3.2.5. Extended Value of FlowEndReason 9
3.3. Definition of New IEs..... 10
4. Application of the New IEs for Attack Detection 12
4.1. Detect ICMP Reflection Attack..... 12
4.2. Detect Fragment Attack..... 13

4.3. Detect Slowloris Attack..... 14
4.4. Detect Out-of-order Packets Attack 15
5. Security Considerations..... 15
6. IANA Considerations 15
7. References 19
7.1. Normative References..... 19
7.2. Informative References..... 19
8. Acknowledgments 20

1. Introduction

As network security issues arising dramatically nowadays, network administrators are eager to detect and identify attacks as early as possible, generate countermeasures with high agility. Due to the enormous amount of network attack types, metrics useful for attack detection are also enormous. Moreover, attacking methods are evolved rapidly, which brings challenges to designing detection mechanism.

Specifically, DOTS WG aims for developing the standard solution to fight against the DDoS attacks. The following sentence is from the DOTS WG charter:

"The aim of DDoS Open Threat Signaling (DOTS) is to develop a standards based approach for the realtime signaling of DDoS related telemetry and threat handling requests and data between elements concerned with DDoS attack detection, classification, traceback, and mitigation."

According to the above sentence, the signaling mechanisms and contents are all the essential parts of the solution, in which the contents refer to the realtime DDoS related telemetry information and threat handling messages.

The IPFIX Protocol [RFC7011] defines a generic exchange mechanism for flow information and events. It supports source-triggered exporting of information via the push model approach. The IPFIX Information Model [IPFIX-IANA] defines a list of standard Information Elements (IEs) which can be carried by the IPFIX protocol. The IPFIX requirement [RFC3917] points out that one of the target applications of IPFIX is attack and intrusion detection. Although the existing IPFIX/PSAMP protocol, packet selection methods, as well as the related standard IEs provide a rich source of data for security inspection by checking the status/events of the traffic, there are still some gaps existing to identify some categories of the DDoS attacks. More detailed gap analysis is given in the following section.

This document focuses on the DDoS related telemetry information part for DOTS, and proposes using the connection sampling method with a set of IPFIX IEs for the goal of inspecting mainly some connection-based and Zero-Day DDoS attacks, which normally are the kinds of the low & slow DDoS attack and not easy to be inspected as flood attacks. Some of these IPFIX IEs already exist; some are the new defined ones with their formats specified. The wise utilization of these IEs will improve the DDoS attack inspection and will support the offline analysis of data from different operators in the future with minimal resource consumption, which is very necessary for increasing the operators' intelligence of identifying new and unknown DDoS attacks.

This document is structured as following: Section 3 discusses the connection sampling mechanism and introduces the new IPFIX IEs derived from relevant use cases. Section 4 describes how to use these IEs to detect specific DDoS attacks.

2. Conventions used in this document

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].

2.1. Terminology

IPFIX-specific terminology (Information Element, Template, Template Record, Options Template Record, Template Set, Collector, Exporter, Data Record, etc.) used in this document is defined in Section 2 of [RFC7011]. As in [RFC7011], these IPFIX-specific terms have the first letter of a word capitalized.

This document also makes use of the same terminology and definitions as [I-D.draft-ietf-dots-requirements] and [I-D.draft-ietf-dots-use-cases].

The following are the new terms in this document.

o Connection Sampling

Connection Sampling is a connection oriented sampling mechanism. If one connection is selected for DDoS attack detection, then all the packets (if possible) of this connection will be sampled during one detection period.

o Victim

The target that suffers DDoS attack.

- Observer

Devices or software deployed at observation point defined in IPFIX.

3. Connection Sampling and new IEs

3.1. Packet Sampling vs Connection Sampling

Packet sampling selection is a widely used method to select packets from network traffic for reporting. Its selection operations include time-based selection, count-based selection, random selection, probability-based selection and so on. Although it is easy and efficient, it still has a number of limitations in inspecting some types of DDoS attack:

- Several research projects [N. DUFFIELD, 2003], [D. BRAUCKHOFF 2006] show that packet sampling impacts greater on small volumetric flows (with only few packets) due to the smaller sampling probability compared with large volumetric flows, which means that packet sampling may impair the detection performance for small volumetric flow based DDoS attacks; Connection sampling can pay more attention to small flows than packet sampling.

- The communication is 2-way between source and destination. Current packet sampling is used to select a subset of packets from all the observed packets. One of its purposes is to select the appropriate packets to estimate the whole traffic. Although packet sampling can produce flows by using property matching or hash method, it does not consider semantics of the connection (i.e. whether two flows are belonging to one connection or not). In this scenario, packet sampling is applied independently in each direction. If packets are sampled only in one direction, then it will be difficult or inaccurate to detect specific DDoS attacks such as SNMP/DNS Reflected Amplification because of the loss of the information in the opposite direction. It cannot distinguish whether there are too much requests from the victim or not.

- Although packet sampling at full line rate, i.e. with probability 100%, is not excluded in principle, resource constraints may not permit it in practice [RFC5474]. For certain DDoS attack such as HTTP Slowloris, enough packets are needed to realize the detection. In this scenario, connection sampling can provide more meaningful packets than packet sampling because all the packets it samples are belonging to the target connection.

- o Connection sampling method uses some new connection related IEs for attack analysis because of their meaning/value available for judging the status of one connection, which current packet sampling method is lack of. For example, tcpControlStateBits is an IE to record the current state of TCP session. If a packet with erroneous flag is identified in any stage of three handshakes, it should be considered as a symptom of exception.

As a consequence from the above analysis, a connection oriented sampling method is more suitable for the security application: Rather than sampling a small part of packets in the traffic between the communication peers, the connection sampling records all (if possible) TCP/UDP connection packets (including packets during connection setup and close phase if there is) between them once that connection is selected to be sampled. In nature, the connection sampling method is able to track the complete working status of the connection state machine. So, it can identify the abnormal state of the connection or the attack easily and accurately. Although the IPFIX/PSAMP also supports the connection sampling mechanism (that is the packet filtering technology for packet selection [RFC5475]), it does not explicitly discuss how to use this method for the detection of connection-based and Zero-Day DDoS attacks in a systematical way. Furthermore, if the observer (e.g. device, middle box) supports the export of the new IPFIX IEs proposed in this document, the traffic volume between exporter and analyzer can be greatly reduced compared with PSAMP, which should export the detailed packet information for further attack analysis.

3.2. Use Cases for New IEs

In this section, several use cases are discussed to identify the requirements where new IEs are desirable for the network attacks detection.

3.2.1. Upstream/Downstream Counters

Take ICMP reflection attack as an example, ICMP flow model has features such as the ICMP Echo/Echo Reply dominate the whole traffic flow, ICMP packet interval is usually not too short (normally 1 pkt/s). Usually, the normal ratio between ICMP echo to ICMP echo reply packets is around 1:1. When a DDOS reflection attack happens, a sudden burst of messages to a destination endpoint can be detected. In turn, the ratio between echo reply and echo packets will be significantly biased from the normal ratio, i.e., exceed 20:1. So the proper way to distinguish an attack from the normal communication is to check this ratio.

However, the current IPFIX IEs for ICMP contain the ICMP type and code for both IPv4 and IPv6 only for a single ICMP packet rather than statistical property of the ICMP session. Further metrics like the cumulated sum of various counters should be calculated based on sampling method defined by the Packet SAMPling (PSAMP) protocol [RFC5477]. Similar problems occur in TCP, UDP, SNMP and DNS attacks. It would be useful to calculate the number of the upstream and downstream packets for one connection separately over time in order to detect the anomalies of the network. For ICMP reflection attack, a more generic approach is to define two basic metrics icmpEchoCount and icmpEchoReplyCount as new IPFIX IEs to represent the cumulated upstream and downstream packets counter within a ICMP connection.

Note that in some case, the asymmetric routing mainly caused by the wide application of multipath technologies (e.g., load balancing, link aggregation) in network will make the bidirectional connection sampling on some network devices over the multipath to be not possible. This problem can be avoided by strategically deploying and enabling the connection sampling function in the network devices which are not located over the multipath.

3.2.2. Fragment statistic

Fragment attack employs unexpected formats of fragmentation, e.g. without last fragment or incorrect fragment offset[RFC791], which result in errors such as fragmentation buffer overrun and fragment overlapped. Existing IPFIX fragmentation metrics includes fragmentOffset, fragmentIdentification, fragmentFlags, which only indicate the attributes of a single fragment, and are not suitable for attack detection. Instead, the network attack should be observed based upon a historic, integrated view of fragmented packets of a connection. For instances, if more than 500 out of 1000 fragmented packets have fragment errors, it is likely that a fragment attack happens.

Therefore, a number of new IEs associated with fragment statistics are proposed as follows:

- fragmentPacketCount: The number of the fragmented packets of the same connection should be checked, and this metric is proposed;

- fragmentFirstTooShortCount: Attacker might intent to exclude destination port from the first fragment so as to bypass detection from firewall. This metric is proposed to indicate the number of the invalid first fragments in the observed connection;

- fragmentFlagErrorCount: This metric is proposed to detect early whether the fragment flags are incorrectly set on purpose.

- fragmentOffestErrorCount: This metric is proposed to count the number of fragments with offset error, and the value can be used to indicate attack occurs;

3.2.3. Response Time Calculation

For other DDoS attacks such as Http slowloris, there will be too many connections that should be kept in the victim (server), which lead to excessive resource consumption. As a result, the response time between client and server will increase greatly. Challenge Collapasar(CC) attack can also exhaust the resources of the server and generate the similar results. Thus, the following IEs are proposed as a symptom of these kinds of attacks:

serverResponseTime: For tcp, it denotes the time difference between the time point that the observer views the SYN packet from client to server and the time point that the observer views the SYN-ACK packet from server to client.

clientResponseTime: For tcp, it denotes the time difference between the time point that the observer views the SYN-ACK packet from server to client and the time point that the observer views the ACK packet from client to server.

sessionResponseTime: The sum of serverResponseTime and clientResponseTime. It is the Round Trip Time (RTT) between client and server.

3.2.4. Symptoms of Exceptions

In http slowloris attack the client may send packets to victim periodically which can cause the performance lost on the server. The characteristic of the attack is that there are too many connections on the victim. However, the volume for these connections is small. In order to detect this attack, the first step is to get the packets that are belonging to the same connection. The second step is to find the periodicity. Thus the two indices pktTimeInterval and pktTimeIntervalVariance are needed. The index pktTimeInterval denotes the average time difference between two successive packets and the index pktTimeIntervalVariance denotes the variance of multiple time difference. Large pktTimeInterval and small pktTimeIntervalVariance can be a symptom of slow packet attack. On the other hand, the payload size of the packets in http slowloris

attack is very small and the size difference is also small. So the index octetVariance can be used to identify the characteristic.

To degrade the performance of the victim, the malicious clients may send too many out-of-order packets, which will consume too much memory on the server. Although out-of-order packets are permit in the TCP protocol, it is possible to be leveraged to cause DDoS attack. So the index tcpOutoforderTotalCount is helpful to detect this kind of exception. For observer, it maintains one counter for each tcp connection. The initial sequence number of the client is saved in the counter. The counter increases by the sequence number of the packets it sees from client to server. If the observer sees a packet with lower sequence number than the current counter value, then the packet will be considered as an out-of-order packet.

In IPFIX, the index tcpControlBits is used to record the corresponding status bits in TCP header of the packets[IPFIX-IANA]. In order to detect the application attacks which can cause the protocol exception such as the wrong use of the tcp status bits before and after the tcp connection establishment, another index called tcpControlStateBits is needed. For example, when the observer sees the SYN packet from client to server, it sets 15th bit of tcpControlStateBits to 1; when it sees the SYN-ACK packet from server to client, it sets 14th bit to 1, and so on. If one endpoint sends the packet with wrong bits during the establishment of the connection, then the observer will identify the exception by the value of tcpControlStateBits.

3.2.5. Extended Value of FlowEndReason

Refer to [IPFIX-IANA], there are 5 defined reasons for Flow termination, with values ranging from 0x01 to 0x05:

0x01: idle timeout

0x02: active timeout

0x03: end of Flow detected

0x04: forced end

0x05: lack of resources

There is an additional reason caused by state machine anomaly. When FIN/SYN is sent, but no ACK is replied after a waiting timeout, the existing five reasons do not match this case. Therefore, a new value

is proposed to extend the FlowEndReason, which is 0x06: protocol exception timeout.

3.3. Definition of New IEs

The following is the table of all the new IEs that a device would need to export for attack statistic analysis. The recommended registrations to IANA are described in the IANA considerations section.

Field Name	Size (bits)	IANA IPFIX ID	Description
fragmentPacketCount	32	TBD	Counter of session fragments
fragmentFirstTooShortCount	32	TBD	Number of packets with first fragment too short
fragmentFlagErrorCount	32	TBD	Number of fragments with erroneous flag
fragmentOffsetErrorCount	32	TBD	Number of fragments with erroneous offset
icmpEchoCount	32	TBD	The number of ICMP echo.
icmpEchoReplyCount	32	TBD	The number of ICMP echo reply
octetVariance	64	TBD	IP packet byte variance statistic
tcpControlStateBits	16	TBD	tcp states
tcpOutoforderTotalCount	64	TBD	out of order packets statistic
pktTimeInterval	64	TBD	the average time interval between two successive packets
pktTimeIntervalVariance	64	TBD	the variance of pktTimeInterval
serverResponseTime	16	TBD	the response time of a server
clientResponseTime	16	TBD	the response time of a client
sessionResponseTime	16	TBD	the response time of a session

Table 1: Information Element Table

4. Application of the New IEs for Attack Detection

This section presents a number of examples to help for the easy understanding of the application of these new IEs for attack detection.

4.1. Detect ICMP Reflection Attack

According to previous analysis, the template for detecting ICMP reflection attack should at least contain IEs shown in Table 2.

Set ID = 2	Length = 40 octets
Template ID TBD	Field Count = 8
|0| sourceIPv4Address	Field Length = 4
|0| destinationIPv4Address	Field Length = 4
|0| protocolIdentifier	Field Length = 1
|0| packetDeltaCount	Field Length = 8
|0| icmpEchoCount	Field Length = 4
|0| icmpEchoReplyCount	Field Length = 4
|0| flowStartSeconds	Field Length = 4
|0| flowEndSeconds	Field Length = 4

Table 2: Template example for detecting ICMP attack

An example of the actual ICMP event data record is shown below in a readable form as below:

{sourceIPv4Address = 192.168.0.101, destinationIPv4Address = 192.168.0.201, protocolIdentifier = 1, packetDeltaCount = 3000, icmpEchoCount = 120, icmpEchoReplyCount = 2880, flowStartSeconds = 100, flowEndSeconds = 200}

protocolIdentifier = 1 represents the ICMP proptocol. There are 30 ICMP messages transmited per second. The ICMP Echo Reply to ICMP

Echo packet ratio is 24:1, which indicates a high possibility of ICMP reflection attack.

4.2. Detect Fragment Attack

The template for detecting fragment attack should at least contain IEs shown in Table 3. It requires the observation point to trace complete fragmented packet and accumulate the errors.

Set ID = 2	Length = 48 octets
Template ID TBD	Field Count = 10
|0| sourceIPv4Address	Field Length = 4
|0| destinationIPv4Address	Field Length = 4
|0| protocolIdentifier	Field Length = 1
|0| packetDeltaCount	Field Length = 8
|0| fragmentPacketCount	Field Length = 4
|0|fragmentFirstTooShortCount	Field Length = 4
|0|fragmentFlagErrorCount	Field Length = 4
|0|fragmentOffsetErrorCount	Field Length = 4
|0| flowStartSeconds	Field Length = 4
|0| flowEndSeconds	Field Length = 4

Table 3: Template example for detecting fragment attack

An example of the actual fragment attack record is shown below in a readable form as below:

{sourceIPv4Address = 192.168.0.101, destinationIPv4Address = 192.168.0.201, protocolIdentifier = 6, packetDeltaCount = 5000, fragmentPacketCount = 4000, fragmentFirstTooShortCount = 0, fragmentFlagErrorCount = 0, fragmentOffestErrorCount = 3000, flowStartSeconds = 100, flowEndSeconds = 200}

In this case, fragment offset errors are used to exhaust resource at the receiver.

4.3. Detect Slowloris Attack

The template for detecting resource exhausting application attack such as http slowloris attack should contain a subnet of IEs shown in Table 4.

Set ID = 2	Length = 48 octets
Template ID TBD	Field Count = 10
|0| sourceIPv4Address	Field Length = 4
|0| destinationIPv4Address	Field Length = 4
|0| protocolIdentifier	Field Length = 1
|0| serverResponseTime	Field Length = 2
|0| clientResponseTime	Field Length = 2
|0| sessionResponseTime	Field Length = 2
|0| pktTimeInterval	Field Length = 4
|0| pktTimeIntervalVariance	Field Length = 4
|0| flowStartSeconds	Field Length = 4
|0| flowEndSeconds	Field Length = 4

Table 4: Template example for detecting slowloris attack

An example of the actual record is shown below in a readable form as below:

{sourceIPv4Address = 192.168.0.101, destinationIPv4Address = 192.168.0.201, protocolIdentifier = 6, serverResponseTime = 200, clientResponseTime = 10, sessionResponseTime = 210, pktTimeInterval = 500, pktTimeIntervalVariance = 1000, flowStartSeconds = 100, flowEndSeconds = 200}

4.4. Detect Out-of-order Packets Attack

The template for detecting out-of-order packets attack should contain IEs shown in Table 5.

Set ID = 2	Length = 32 octets
Template ID TBD	Field Count = 10
|0| sourceIPv4Address	Field Length = 4
|0| destinationIPv4Address	Field Length = 4
|0| protocolIdentifier	Field Length = 1
|0| packetDeltaCount	Field Length = 8
|0| tcpOutoforderTotalCount	Field Length = 4
|0| flowStartSeconds	Field Length = 4
|0| flowEndSeconds	Field Length = 4

Table 5: Template example for detecting out-of-order attack

An example of the actual record is shown below in a readable form as below:

{sourceIPv4Address = 192.168.0.101, destinationIPv4Address = 192.168.0.201, protocolIdentifier = 6, packetDeltaCount =3000, tcpOutoforderTotalCount = 2000, flowStartSeconds = 100, flowEndSeconds = 200}

5. Security Considerations

No additional security considerations are introduced in this document. The same security considerations as for the IPFIX protocol [RFC7011] apply.

6. IANA Considerations

The following information elements are requested from IANA IPFIX registry.

Name: fragmentPacketCount

Description: This Information Element is the counter of session fragments.

Abstract Data Type: unsigned32

Data Type Semantics: TBD

Name: fragmentFirstTooShortCount

Description: This Information Element indicates the number of packets with first fragment too short.

Abstract Data Type: unsigned32

Data Type Semantics: TBD

Name: fragmentFlagErrorCount

Description: This Information Element specifies number of fragments with flag error. When the DF bit and MF bit of the fragment flag are set in the same fragment, there is an error at the fragment flag.

Abstract Data Type: unsigned32

Data Type Semantics: TBD

Name: fragmentOffsetErrorCount

Description: This Information Element specifies number of fragments with offset error.

Abstract Data Type: unsigned32

Data Type Semantics: TBD

Name: icmpEchoCount

Description: icmp Echo packets.

Abstract Data Type: unsigned32

Data Type Semantics: deltaCounter

Name: icmpEchoReplyCount

Description: icmp Echo Reply packets.

Abstract Data Type: unsigned32

Data Type Semantics: deltaCounter

Name: octetVariance

Description: IP packet byte variance statistic.

Abstract Data Type: unsigned64

Data Type Semantics: quantity

Name: tcpControlStateBits

Description: the current tcp states of the connection.

Abstract Data Type: unsigned16

Data Type Semantics: flags

Name: tcpOutoforderTotalCount

Description: out of order packets statistic.

Abstract Data Type: unsigned64

Data Type Semantics: totalCounter

Name: pktTimeInterval

Description: the average time interval between two successive packets in a flow.

Abstract Data Type: unsigned32

Data Type Semantics: quantity

Name: pktTimeIntervalVariance

Description: the variance of the time intervals between two successive packets in a flow.

Abstract Data Type: unsigned64

Data Type Semantics: quantity

Name: serverResponseTime

Description: the response time of a server.

Abstract Data Type: unsigned16

Data Type Semantics: quantity

Name: clientResponseTime

Description: the response time of a client.

Abstract Data Type: unsigned16

Data Type Semantics: quantity

Name: sessionResponseTime

Description: the response time of a session.

Abstract Data Type: unsigned16

Data Type Semantics: quantity

A new value is added to FlowEndReason:

0x06: protocol exception timeout

The flow was terminated due to protocol state machine anomaly and unexpected timeout.

7. References

7.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC7011] Claise, B., Trammell, B., and P. Aitken, "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information", STD 77, RFC 7011, September 2013.

[RFC3917] Quittek, J., Zseby, T., Claise, B., Zander, S., "Requirements for IP Flow Information Export (IPFIX)", RFC 3917, October 2004.

[RFC5474] N. Duffield Ed., D. Chiou, B. Claise, A. Greenberg, M. Grossglauser, J. Rexford, "A Framework for Packet Selection and Reporting", RFC 5474, March 2009.

[RFC5475] T. Zseby, M. Molina, N. Duffield, S. Niccolini, F. Raspall, "Sampling and Filtering Techniques for IP Packet Selection", RFC 5475, March 2009.

[RFC5476] B. Claise, Ed., A. Johnson, J. Quittek, "Packet Sampling (PSAMP) Protocol Specifications", RFC 5476, March 2009.

[RFC5477] T. Dietz, B. Claise, P. Aitken, F. Dressler, G. Carle, " Information Model for Packet Sampling Exports ", RFC 5477, March 2009.

7.2. Informative References

[IPFIX-IANA]

IANA, "IPFIX Information Elements registry",

<http://www.iana.org/assignments/ipfix>.

[I-D.draft-ietf-dots-requirements]

Mortensen, A., Moskowitz, R., Reddy, T., "DDoS Open Threat Signaling Requirements", work in progress, October, 2015.

[I-D.draft-ietf-dots-use-cases]

Dobbins, R., Fouant, S., Migault, D., Moskowitz, R., Teague, N., Xia, L., " Use cases for DDoS Open Threat Signaling", work in progress, October, 2015.

[D. BRAUCKHOFF 2006]

Daniela Brauckhoff, Bernhard Tellenbach, Arno Wagner, Martin May, and Anukool Lakhina. 2006. Impact of packet sampling on anomaly detection metrics. In Proceedings of the 6th ACM SIGCOMM conference on Internet measurement (IMC '06). ACM, New York, NY, USA, 159-164.

[N. DUFFIELD, 2003]

DUFFIELD, N., LUND, C., AND THORUP, M., Estimating Flow Distributions from Sampled Flow Statistics. In ACM SIGCOMM (Karlsruhe, August 2003).

8. Acknowledgments

The authors would thank Danping He and Yibo Zhang for their great help during the initial period of this draft.

The authors would also thank Tienan Wang for his explain about the implementation of DDoS attack solutions.

This document was prepared using 2-Word-v2.0.template.dot.

Authors' Addresses

Tianfu Fu
Huawei
Q11, Huanbao Yuan, 156 Beiqing Road, Haidian District
Beijing 100095
China

Email: futianfu@huawei.com

DaCheng Zhang
Alibaba

Email: Dacheng.zdc@alibaba-inc.com

Liang Xia (Frank)
Huawei

101 Software Avenue, Yuhuatai District
Nanjing, Jiangsu 210012
China

Email: Frank.xialiang@huawei.com

Bo Zhang (Alex)
Huawei

101 Software Avenue, Yuhuatai District
Nanjing, Jiangsu 210012
China

Email: Alex.zhangbo@huawei.com

Min Li
Huawei

Huawei Technologies Duesseldorf GmbH, European Research Center,
Riesstr. 25, 80992 Muchen, Germany
Email: l.min@huawei.com