Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: 28 August 2024

N. Elkins
Inside Products, Inc.
M. Ackermann
BCBS Michigan
A. Deshpande
NITK Surathkal/Google
T. Pecorella
University of Florence
A. Rashid
Politecnico di Bari
25 February 2024

IPv6 Performance and Diagnostic Metrics Version 2 (PDMv2) Destination Option

draft-ietf-ippm-encrypted-pdmv2-06

Abstract

RFC8250 describes an optional Destination Option (DO) header embedded in each packet to provide sequence numbers and timing information as a basis for measurements. As this data is sent in clear-text, this may create an opportunity for malicious actors to get information for subsequent attacks. This document defines PDMv2 which has a lightweight handshake (registration procedure) and encryption to secure this data. Additional performance metrics which may be of use are also defined.

About This Document

This note is to be removed before publishing as an RFC.

The latest revision of this draft can be found at https://ameyand.github.io/PDMv2/draft-elkins-ippm-encrypted-pdmv2.html. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-ietf-ippm-encrypted-pdmv2/.

Discussion of this document takes place on the IP Performance Measurement Working Group mailing list (mailto:ippm@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/ippm/. Subscribe at https://www.ietf.org/mailman/listinfo/ippm/.

Source for this draft and an issue tracker can be found at https://github.com/ameyand/PDMv2.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

The current PDM is an IPv6 Destination Options header which provides information based on the metrics like Round-trip delay and Server delay. This information helps to measure the Quality of Service (QoS) and to assist in diagnostics. However, there are potential risks involved transmitting PDM data during a diagnostics session.

PDM metrics can help an attacker understand about the type of machine and its processing capabilities. Inferring from the PDM data, the attack can launch a timing attack. For example, if a cryptographic protocol is used, a timing attack may be launched against the keying material to obtain the secret.

Along with this, PDM does not provide integrity. It is possible for a Machine-In-The-Middle (MITM) node to modify PDM headers leading to incorrect conclusions. For example, during the debugging process using PDM header, it can mislead the person showing there are no unusual server delays.

PDMv2 is an IPv6 Destination Options Extension Header which adds confidentiality, integrity and authentication to PDM.

The procedures specified in RFC8250 for header placement, implementation, security considerations and so on continue to apply for PDMv2.

2. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Terminology

* Endpoint Node: Creates cryptographic keys in collaboration with a partner.

* Client: An Endpoint Node which initiates a session with a listening port on another Endpoint Node and sends PDM data.

* Server: An Endpoint Node which has a listening port and sends PDM data to another Endpoint Node.

Note: a client may act as a server (have listening ports).

* Public and Private Keys: A pair of keys that is used in asymmetric cryptography. If one is used for encryption, the other is used for decryption. Private Keys are kept hidden by the source of the key pair generator, but Public Key is known to everyone. pkX (Public Key) and skX (Private Key). Where X can be, any client or any server.

* Pre-shared Key (PSK): A symmetric key. Uniformly random bitstring, shared between any Client or any Server or a key shared between an entity that forms client-server relationship. This could happen through an out-of band mechanism: e.g., a physical meeting or use of another protocol.

* Session Key: A temporary key which acts as a symmetric key for the whole session.

4. Protocol Flow

The protocol will proceed in 2 steps.

Step 1: Creation of cryptographic secrets between Server and Client. This includes the creation of pkX and skX.

Step 2: PDM data flow between Client and Server.

These steps MAY be in the same session or in separate sessions. That is, the cryptographic secrets MAY be created beforehand and used in the PDM data flow at the time of the "real" data session.

After-the-fact (or real-time) data analysis of PDM flow may occur by network diagnosticians or network devices. The definition of how this is done is out of scope for this document.

4.1. Client - Server Negotiation

The two entities exchange a set of data to ensure the respective identities. This could be done via a TLS or other session. The exact nature of the identity verification is out-of-scope for this document.

They use Hybrid Public-Key Encryption scheme (HPKE) Key Encryption Mechanism (KEM) to negotiate a "SharedSecret".

Each Client and Server derive a "SessionTemporaryKey" by using HPKE Key Derivation Function (KDF), using the following inputs:

* The "SharedSecret".

* The 5-tuple (SrcIP, SrcPort, DstIP, DstPort, Protocol) of the communication.

* An Epoch.

The Epoch SHOULD be initialized to zero. A change in the Epoch indicates that the SessionTemporaryKey has been rotated.

When the Epoch rolls over, the SharedSecret SHOULD be re-negotiated.

The Epoch MUST be incremented when the Packet Sequence Number (PSN) This Packet (PSNTP) is rolled over. It MAY be incremented earlier, depending on the implementation and the security considerations.

The sender MUST NOT create two packets with identical PSNTP and Epoch.

The SessionTemporaryKey using a KDF with the following inputs:

* SrcIP, SrcPort, DstIP, DstPort, Protocol, SharedSecret, Epoch.

4.2. Implementation Guidelines

How should a network administrator decide whether a client should use PDM, unencrypted PDMv2, or encrypted PDMv2? This decision is a network policy issue. The administrator must be aware that PDM or unencrypted PDMv2 might expose too much information to malicious parties.

That said, if the network administrator decides that taking such a risk within their network is acceptable, then they should make the decision that is appropriate for their network.

Alternatively, the network administrator might choose to create a policy that prohibits the use of PDM or unencrypted PDMv2 on their network. The implementation SHOULD provide a way for the network administrator to enforce such a policy.

The server and client implementations SHOULD support PDM, unencrypted PDMv2, and encrypted PDMv2. If a client chooses a certain mechanism (e.g., PDM), the server MAY respond with the same mechanism, unless the network administrator has selected a policy that only allows certain mechanisms on their network.

4.2.1. Use Case 1: Server does not understand PDM or PDMv2

If a client sends a packet with PDM or PDMv2 and the server does not have code which understands the header, the packet is processed according to the Option Type which is defined in RFC8250 and is in accordance with RFC8200.

The Option Type identifiers is coded to skip over this option and continue processing the header.

4.2.2. Use Case 2: Server does not allow PDM or PDMv2

If a client sends a packet with PDM and the network policy is to only allow encrypted or unencrypted PDMv2, then the PDM / PDMv2 header MUST be ignored and processing continue normally.

The server SHOULD log such occurrences but MUST apply rate limiting to any such logs. The implementor should be aware that logging or returning of error messages can be used in a Denial of Service reflector attack. An attacker might send many packets with PDM / PDMv2 and cause the receiver to experience resource exhaustion.

The routers involved may have implemented filtering as per [RFC9288] on filtering of IPv6 extension headers which may impact the receipt of PDM / PDMv2. The organization which manages the network within

which PDM / PDMv2 is sent should take care that the filtering of extension headers is done correctly so that the desired effect is obtained.

5. Security Goals

As discussed in the introduction, PDM data can represent a serious data leakage in presence of a malicious actor.

In particular, the sequence numbers included in the PDM header allows correlating the traffic flows, and the timing data can highlight the operational limits of a server to a malicious actor. Moreover, forging PDM headers can lead to unnecessary, unwanted, or dangerous operational choices, e.g., to restore an apparently degraded Quality of Service (QoS).

Due to this, it is important that the confidentiality and integrity of the PDM headers is maintained. PDM headers can be encrypted and authenticated using the methods discussed in Section 5.4, thus ensuring confidentiality and integrity. However, if PDM is used in a scenario where the integrity and confidentiality is already ensured by other means, they can be transmitted without encryption or authentication. This includes, but is not limited to, the following cases:

a) PDM is used over an already encrypted medium (For example VPN tunnels).

b) PDM is used in a link-local scenario.

c) PDM is used in a corporate network where there are security measures strong enough to consider the presence of a malicious actor a negligible risk.

5.1. Security Goals for Confidentiality

PDM data MUST be kept confidential between the intended parties, which includes (but is not limited to) the two entities exchanging PDM data, and any legitimate party with the proper rights to access such data.

5.2. Security Goals for Integrity

An implementation SHOULD attempt to detect if PDM data is forged or modified by a malicious entity. In other terms, the implementation should attempt to detect if a malicious entity has generated a valid PDM header impersonating an endpoint or modified a valid PDM header.

5.3. Security Goals for Authentication

An unauthorized party MUST NOT be able to send PDM data and MUST NOT be able to authorize another entity to do so. Alternatively, if authentication is done via any of the following, this requirement MAY be considered to be met.

a) PDM is used over an already authenticated medium (For example, TLS session).

b) PDM is used in a link-local scenario.

c) PDM is used in a corporate network where security measures are strong enough to consider the presence of a malicious actor a negligible risk.

5.4. Cryptographic Algorithm

Symmetric key cryptography has performance benefits over asymmetric cryptography; asymmetric cryptography is better for key management. Encryption schemes that unite both have been specified in [RFC1421], and have been participating practically since the early days of public-key cryptography. The basic mechanism is to encrypt the symmetric key with the public key by joining both yields. Hybrid public-key encryption schemes (HPKE) [RFC9180] used a different approach that generates the symmetric key and its encapsulation with the public key of the receiver.

It is RECOMMENDED to use the HPKE framework that incorporates key encapsulation mechanism (KEM), key derivation function (KDF) and authenticated encryption with associated data (AEAD). These multiple schemes are more robust and significantly more efficient than other schemes. While the schemes may be negotiated between communicating parties, it is RECOMMENDED to use default encryption algorithm for HPKE AEAD as AES-128-GCM.

6. PDMv2 Destination Options

6.1. Destinations Option Header

The IPv6 Destination Options extension header [RFC8200] is used to carry optional information that needs to be examined only by a packet's destination node(s). The Destination Options header is identified by a Next Header value of 60 in the immediately preceding header and is defined in RFC 8200 [RFC8200]. The IPv6 PDMv2 destination option is implemented as an IPv6 Option carried in the Destination Options header.

6.2. Metrics information in PDMv2

The IPv6 PDMv2 destination option contains the following base fields:

SCALEDTLR: Scale for Delta Time Last Received

SCALEDTLS: Scale for Delta Time Last Sent

GLOBALPTR: Global Pointer

PSNTP: Packet Sequence Number This Packet

PSNLR: Packet Sequence Number Last Received

DELTATLR: Delta Time Last Received

DELTATLS: Delta Time Last Sent

PDMv2 adds a new metric to the existing PDM [RFC8250] called the Global Pointer. The existing PDM fields are identified with respect to the identifying information called a "5-tuple".

The 5-tuple consists of:

SADDR: IP address of the sender

SPORT: Port for the sender

DADDR: IP address of the destination

DPORT: Port for the destination

PROTC: Upper-layer protocol (TCP, UDP, ICMP, etc.)

Unlike PDM fields, Global Pointer (GLOBALPTR) field in PDMv2 is defined for the SADDR type. Following are the SADDR address types considered:

a) Link-Local

b) Global Unicast

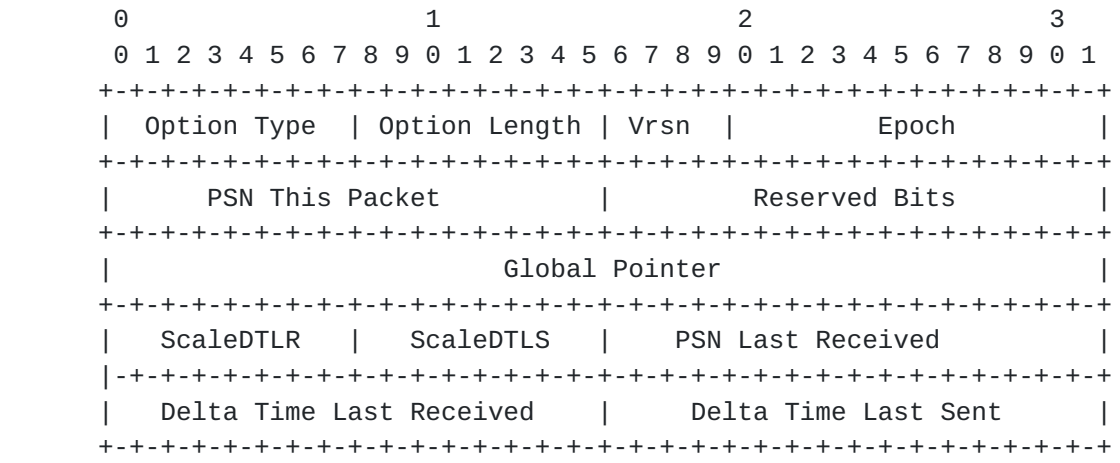
The Global Pointer is treated as a common entity over all the 5-tuples with the same SADDR type. It is initialised to the value 1 and increments for every packet sent. Global Pointer provides a measure of the amount of IPv6 traffic sent by the PDMv2 node.

When the SADDR type is Link-Local, the PDMv2 node sends Global Pointer defined for Link-Local addresses, and when the SADDR type is Global Unicast, it sends the one defined for Global Unicast addresses.

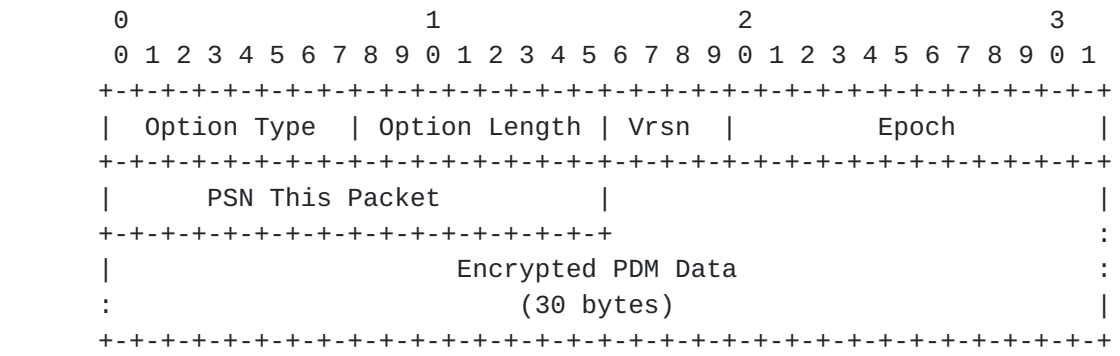
6.3. PDMv2 Layout

PDMv2 has two different header formats corresponding to whether the metric contents are encrypted or unencrypted. The difference between the two types of headers is determined from the Options Length value.

Following is the representation of the unencrypted PDMv2 header:



Following is the representation of the encrypted PDMv2 header:



Option Type

0x0F

8-bit unsigned integer. The Option Type is adopted from RFC 8250 [RFC8250].

Option Length

0x12: Unencrypted PDM

0x22: Encrypted PDM

8-bit unsigned integer. Length of the option, in octets, excluding the Option Type and Option Length fields. The options length is used for differentiating PDM [RFC8250], unencrypted PDMv2 and encrypted PDMv2.

Version Number

0x2

4-bit unsigned number.

Epoch

12-bit unsigned number.

Epoch field is used to indicate the valid SessionTemporaryKey.

Packet Sequence Number This Packet (PSNTP)

16-bit unsigned number.

This field is initialized at a random number and is incremented sequentially for each packet of the 5-tuple.

This field is also used in the Encrypted PDMv2 as the encryption nonce. The nonce MUST NOT be reused in different sessions.

Reserved Bits

16-bits.

Reserved bits for future use. They MUST be set to zero on transmission and ignored on receipt per [RFC3552].

Scale Delta Time Last Received (SCALEDTLR)

8-bit unsigned number.

This is the scaling value for the Delta Time Last Sent (DELTATLS) field.

Scale Delta Time Last Sent (SCALEDTLS)

8-bit unsigned number.

This is the scaling value for the Delta Time Last Sent (DELTATLS) field.

Global Pointer

32-bit unsigned number.

Global Pointer is initialized to 1 for the different source address types and incremented sequentially for each packet with the corresponding source address type.

This field stores the Global Pointer type corresponding to the SADDR type of the packet.

Packet Sequence Number Last Received (PSNLR)

16-bit unsigned number.

This field is the PSNTP of the last received packet on the 5-tuple.

Delta Time Last Received (DELTATLR)

16-bit unsigned integer.

The value is set according to the scale in SCALEDTLR.

Delta Time Last Received = (send time packet n - receive time packet (n - 1))

Delta Time Last Sent (DELTATLS)

16-bit unsigned integer.

The value is set according to the scale in SCALEDTLS.

Delta Time Last Sent = (receive time packet n - send time packet (n - 1))

7. Security Considerations

PDMv2 carries metadata, including information about network characteristics and end-to-end response time. This metadata is used to optimize communication. However, in the context of passive attacks, the information contained within PDMv2 packets can be intercepted by an attacker, and in the context of active attacks the metadata can be modified by an attacker.

In the following we will briefly outline the threat model and the associated security risks, using RFC3552 terminology and classification.

7.1. Limited Threat Model

We assume that the attacker does not control the endpoints, but it does have a limited control of the network, i.e., it can either monitor the communications (leading to passive attacks), or modify/ forge packets (active attacks).

7.1.1. Passive attacks with unencrypted PDMv2

Passive Attack Scenario: In a passive attack, the attacker seeks to obtain information that the sender and receiver of the communication would prefer to keep private. In this case, the attacker is not altering the packets but is intercepting and analyzing them. Here's how this can happen in the context of unencrypted PDMv2:

a. Being on the same LAN: The simplest way for an attacker to launch a passive attack is to be on the same Local Area Network (LAN) as the victim. Many LAN configurations, such as Ethernet, 802.3, and FDDI, allow any machine on the network to read all traffic destined for any other machine on the same LAN. This means that if PDM packets are sent over the LAN, the attacker can capture them.

b. Control of a Host in the Communication Path: If the attacker has control over a host that lies in the communication path between two victim machines, they can intercept PDM packets as they pass through this compromised host. This allows the attacker to collect metadata without being on the same LAN as the victim.

c. Compromising Routing Infrastructure: In some cases, attackers may actively compromise the routing infrastructure to route traffic through a compromised machine. This can facilitate a passive attack on victim machines. By manipulating routing, the attacker can ensure that PDMv2 packets pass through their controlled node.

d. Wireless Networks: Wireless communication channels, such as those using 802.11 (Wi-Fi), are particularly vulnerable to passive attacks. Since data is broadcast over well-known radio frequencies, an attacker with the ability to receive those transmissions can intercept PDMv2 packets. Weak or ineffective cryptographic protection in wireless networks can make it easier for attackers to capture this data.

Goal of Passive Attack: In a passive attack, the attacker's goal is to obtain sensitive information from intercepted packets. In the case of PDMv2, this information may include network characteristics, end-to-end response times, and potentially any other metadata that is transmitted. This information can be valuable to the attacker for various purposes, such as analyzing network performance or gaining insights into communication patterns.

In summary, within the limited Internet threat model described in RFC3552, attackers with the ability to intercept packets can conduct passive attacks to capture metadata carried in IPv6 unencrypted PDMv2 packets. This information can be useful for the attacker, even without actively altering the communication. Security measures, such as encryption and network segmentation, are important countermeasures to protect against such passive attacks.

7.1.2. Passive attacks with encrypted PDMv2

Passive Attack Scenario: An attacker is trying to seek useful information from encrypted PDMv2 packets happening between two different entities. Encrypted PDMv2 has most of the metadata fields encrypted except for PSNTP which is also used as a nonce in HPKE AEAD.

Goal of Passive Attack: In this attack, the attacker is trying to obtain the order in which the packets were sent from the sender to the receiver for different flows. The amount of information gathered by the attacker is similar, to some extent, to the ones available by inspecting the TTCP sequence number, which is also usually not protected. Therefore, we consider this information leak acceptable.

Nevertheless, this point should be noted if complete traffic obfuscation (including packet reordering) is necessary. In these cases it is suggested to use IPSec ESP [RFC4303] in tunnel mode (in which case the PDMv2 can be used unencrypted).

7.1.3. Active attacks with unencrypted PDMv2

There are also active attacks within the context of the limited Internet threat model defined in [RFC3552]. In this model, active attacks involve an attacker writing data to the network, and the attacker can forge packets and potentially manipulate the behavior of devices or networks. Let's break down how message modification, deletion, or insertion by attackers using the unencrypted IPv6 Performance and Destination option v2 (PDMv2) fits into this threat model:

1. Message Modification Attack:

In a message modification attack, the attacker intercepts a message, modifies its content, and then reinserts it into the network. This attack is significant because it allows the attacker to tamper with the integrity of the data being transmitted.

Example: Suppose an attacker intercepts an IPv6 packet containing unencrypted PDMv2 information that includes network and end-to-end response time metadata. The attacker modifies this information, such as altering the response time data or inserting false information. When this modified packet reaches its destination, the receiving device or network may act based on this malicious information, potentially leading to degraded performance, incorrect network management decisions, wrong performance data collection, etc. A direct consequence of modifying the performance data could be, for example, to hide an ongoing QoS violation, or to create a fake QoS violation, with consequences on the violation of Service Level Agreements.

2. Message Deletion Attack:

In a message deletion attack, the attacker removes a message from the network. This attack can be used in conjunction with other attacks to disrupt communication or achieve specific objectives.

Example: Consider a scenario where an attacker deletes certain IPv6 packets that contain unencrypted PDMv2 information, or deletes the PDM header from the packet. If the PDMv2 is used for network monitoring or quality of service (QoS) management, the deletion of these packets can cause the monitoring system to miss critical data, potentially leading to inaccurate network performance analysis or decisions.

3. Message Insertion Attack:

In a message insertion attack, the attacker forges a message and injects it into the network.

Example: An attacker could forge IPv6 packets containing unencrypted PDMv2 data with fake source addresses and inject them into the network. If PDM is used for making routing or resource allocation decisions, these injected packets can influence the network's behavior, potentially causing it to take suboptimal routes or allocate resources incorrectly.

All these attacks are considered active attacks because the attacker actively manipulates network traffic, and they can potentially spoof the source address to disguise their identity. In the limited

Internet threat model defined in [RFC3552], it is assumed that attackers can forge packets and carry out such active attacks. These attacks highlight the importance of securing network protocols, authenticating messages, and implementing proper security measures to protect against them.

7.1.4. Active attacks with encrypted PDMv2

Encrypted PDMv2 provides inherent protection against active attacks like Message Modification by providing integrity. If either of the sequence number or encrypted PDMv2 contents are modified then decryption will fail.

Message Deletion Attack can be performed for encrypted PDMv2 similarly to unencrypted PDMv2.

Impersonation Attack: Encrypted PDMv2 relies on a shared secret negotiated by an external protocol (e.g., TLS). If key exchange does have authentication check, then an adversary who impersonates the host can derive a key with the destination client and potentially perform all the above active attacks even on encrypted PDMv2.

7.2. Topological considerations

The same topological considerations highlighted in [RFC3552] applies in this context. Passive attacks and active attacks where the messages need to be modified or deleted are more likely if the attacker is on-path, while message insertion attacks are more likely when the attacker is off-path but can happen also when the attacker is on-path. Link-local attacks can be considered as a special case of on-path for PDM, i.e., for PDM a link-local attacker has no special privileges with respect to an on-path attacker.

7.3. Further mitigations

PDM includes cryptographic mechanisms to mitigate passive and active attacks. As a further security mechanism to protect from active attacks, it is possible for an implementation to include logging of anomalous events, e.g.:

* Missing PDM header when expected (counteracts the Message Deletion).

* Unusual variations of the PDM data (counteracts the Message Modification)

* Accept the PDM data only if the application level accepts the packet payload (counteracts the Message Insertion)

* Monitor repeated or unexpected PDM data (counteracts replay attacks).

Security considerations about HPKE are addressed in RFC 9180.
Security considerations about PDM are addressed in RFC 8250.
Security considerations about destination objects are addressed in RFC 8200.

8. Privacy Considerations

Encryption plays a crucial role in providing privacy as defined by [RFC6973], especially when metadata sniffing is a concern. [RFC6973], titled "Privacy Considerations for Internet Protocols," outlines the importance of protecting users' privacy in the context of various Internet protocols, including IPv6. When metadata like network and end-to-end response time is at risk of being observed by attackers, eavesdroppers, or observers, encryption can help mitigate the privacy risks. Here's how encryption achieves this:

a) Confidentiality: Encryption ensures that the actual content of the communication remains confidential. Even if attackers or observers intercept the data packets, they won't be able to decipher the information without the encryption key. In the case of IPv6 Performance and Destination Option (PDM), the still-visible, non-encrypted metadata is still visiblenegligible, and does not poses confidentiality

b) Content Protection: Metadata, such as network and end-to-end response time, may reveal sensitive information about the communication. By encrypting the content, encryption mechanisms help protect this sensitive data from being exposed. Observers may still see that communication is happening, but they won't be able to glean meaningful information from the metadata.

c) Integrity: Encryption often includes mechanisms to ensure data integrity. It allows the recipient to verify that the received data has not been tampered with during transit. This helps protect against attackers who might try to manipulate the metadata.

It's important to note that while encryption enhances privacy by protecting the content of communication, metadata still poses some challenges. Metadata, such as the fact that communication is occurring and the parties involved, can be revealing. To address this, additional techniques, like traffic obfuscation, may be used to hide metadata patterns. However, complete metadata privacy can be challenging to achieve, especially when communication protocols inherently require some level of metadata exchange.

Specifically, enabling PDM only for a specific set of flows can pose a risk of highlighting their presence between two parties. As a mitigation technique, it is suggested to obfuscate the events, for example by enabling PDM on more flows than strictly necessary.

9. IANA Considerations

No action is needed by IANA.

10. Contributors

The authors wish to thank NITK Surathkal for their support and assistance in coding and review. In particular Dr. Mohit Tahiliani and Abhishek Kumar (now with Google). Thanks also to Priyanka Sinha for her comments. Thanks to the India Internet Engineering Society (iiesoc.in), in particular Dhruv Dhody, for his comments and for providing the funding for servers needed for protocol development. Thanks to Balajinaidu V, Amogh Umesh, and Chinmaya Sharma of NITK for developing the PDMv2 implementation for testing.

11. References

11.1. 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/rfc/rfc2119>.

[RFC3552] Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", BCP 72, RFC 3552, DOI 10.17487/RFC3552, July 2003, <https://www.rfc-editor.org/rfc/rfc3552>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

[RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <https://www.rfc-editor.org/rfc/rfc8200>.

[RFC8250] Elkins, N., Hamilton, R., and M. Ackermann, "IPv6 Performance and Diagnostic Metrics (PDM) Destination Option", RFC 8250, DOI 10.17487/RFC8250, September 2017, <https://www.rfc-editor.org/rfc/rfc8250>.

11.2. Informative References

[RFC1421] Linn, J., "Privacy Enhancement for Internet Electronic Mail: Part I: Message Encryption and Authentication Procedures", RFC 1421, DOI 10.17487/RFC1421, February 1993, <https://www.rfc-editor.org/rfc/rfc1421>.

[RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, DOI 10.17487/RFC4303, December 2005, <https://www.rfc-editor.org/rfc/rfc4303>.

[RFC6973] Cooper, A., Tschofenig, H., Aboba, B., Peterson, J., Morris, J., Hansen, M., and R. Smith, "Privacy Considerations for Internet Protocols", RFC 6973, DOI 10.17487/RFC6973, July 2013, <https://www.rfc-editor.org/rfc/rfc6973>.

[RFC9180] Barnes, R., Bhargavan, K., Lipp, B., and C. Wood, "Hybrid Public Key Encryption", RFC 9180, DOI 10.17487/RFC9180, February 2022, <https://www.rfc-editor.org/rfc/rfc9180>.

[RFC9288] Gont, F. and W. Liu, "Recommendations on the Filtering of IPv6 Packets Containing IPv6 Extension Headers at Transit Routers", RFC 9288, DOI 10.17487/RFC9288, August 2022, <https://www.rfc-editor.org/rfc/rfc9288>.

Appendix A. Sample Implementation of Registration

A.1. Overall summary

In the Registration phase, the objective is to generate a shared secret that will be used in encryption and decryption during the Data Transfer phase. How this is to be done is left to the implementation.

A.2. High level flow

The following steps describe the protocol flow:

1. Client initiates a request to the Server. The request contains a list of available ciphersuites for KEM, KDF, and AEAD.

2. Server responds to the Client with one of the available ciphersuites and shares its pkX.

3. Client generates a secret and its encapsulation. The Client sends the encapsulation and a salt to the Server. The salt is required during KDF in the Data Transfer phase.

4. Server generates the secret with the help of the encapsulation and responds with a status message.

Appendix B. Change Log

Note to RFC Editor: if this document does not obsolete an existing RFC, please remove this appendix before publication as an RFC.

Appendix C. Open Issues

Note to RFC Editor: please remove this appendix before publication as an RFC.

Authors' Addresses

Nalini Elkins
Inside Products, Inc.
United States
Email: nalini.elkins@insidethestack.com

Michael Ackermann
BCBS Michigan
United States
Email: mackermann@bcbsm.com

Ameya Deshpande
NITK Surathkal/Google
India
Email: ameyanrd@gmail.com

Tommaso Pecorella
University of Florence
Italy
Email: tommaso.pecorella@unifi.it

Adnan Rashid
Politecnico di Bari
Italy
Email: adnan.rashid@poliba.it