Workgroup: ipsecme
Internet-Draft:
draft-ponchon-ipsecme-anti-replay-subspaces-01
Published: 13 March 2023
Intended Status: Standards Track
Expires: 14 September 2023
Authors: P. Ponchon
Cisco Meraki
M. Shaikh
Cisco Meraki
P. Pfister
Cisco Meraki
G. Solignac
Cisco Meraki

IPsec and IKE anti-replay sequence number subspaces for traffic-engineered paths and multi-core processing

Abstract

This document discusses the challenges of running IPsec with anti-replay in multi-core environments where packets may be re-ordered (e.g., when sent over multiple IP paths, traffic-engineered paths and/or using different QoS classes). A new solution based on splitting the anti-replay sequence number space into multiple different sequencing subspaces is proposed. Since this solution requires support on both parties, an IKE extension is proposed in order to negotiate the use of the anti-replay sequence number subspaces.

Status of This Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 14 September 2023.

Table of Contents

1. Introduction

The IPsec and IKE protocol suite is very commonly used in secure overlay networks, often interconnecting thousands or tens of thousands of sites. Leveraging the high core-counts and multi-uplinks (e.g., multiple fiber/cable, cellular or MPLS uplinks) capabilities of modern systems is important to bring greater throughput, availability and quality of service.

Such scale and multi-paths requirements conflict with how anti-replay currently works. This document first describes the problems related to running IPsec with anti-replay in conjunction with traffic-engineered paths or multi-core systems, and how existing solutions are not sufficient to address these challenges. An IPsec extension is then defined. It divides the IPsec sequence number space into multiple subspaces. Finally, an IKE extension is defined

in order to enable this option only when both tunnel endpoints support it.

2. Problem Statement

While the problem is explored in more detail in [I-D.mrossberg-ipsecme-multiple-sequence-counters], this section will highlight the key issues associated with running IPsec with anti-replay in multi-core systems and environments where traffic-engineering is used, as well as the limitations of current solutions.

Scaling the current anti-replay mechanism to run on multiple cores concurrently shows performance limitations: - When receiving a packet, preventing the same IPsec packet from being accepted by two different cores in parrallel requires constant synchronization between the cores. - When transmitting a packet, sequence numbers must be allocated efficiently, and packets must be transmitted without too much re-ordering, as to not exceed the receiver's anti-replay window size. This also ends-up requiring locks and synchronization between cores.

A commonly used alternative is to assign each Child SA to a given core, but that limits the throughput that is achievable by a single tunnel and adds a performance overhead associated with passing packets across cores.

These restrictions are discussed in [I-D.pwouters-ipsecme-multi-sa-performance], which mainly focuses on high-throughput IPsec tunnels, but the problem also arises with small tunnels since multiple inner flows processed by multiple threads often need to be transmitted on the same tunnel (causing multiple threads to need to access shared resources).

A possible solution to leverage the multi-core capability of the IPsec peers for a given tunnel would be to allocate one Child SA per core. However, combined with QoS classes and multi-path capabilities, this approach shows scalability issues with both the IKE and IPsec implementations:

* Increased number of IKE negotiations and re-key operations.

* Increased IKE memory usage.

* Data-plane performance degradation due to the use of a larger number of keys.

* Data-plane reduced number of connected peers, due to a hard limit to the number of supported Child SAs.

*When PFS is enabled, the overhead of each Child SA negotiation is increased due to additional Diffie-Hellman operations.

Finally, in situations where packet reordering is present, such as with QoS or multiple uplinks, slower or lower priority packets may fall outside of the anti-replay window and be dropped. Using an extraordinarily large window size causes both performance and scalability limitations.

3. Conventions and Definitions

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 BCP14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

4. Multiple sequence number subspaces

Processing packets associated with a single Child SA on multiple cores and using a single Child SA on multiple paths or with multile QoS classes suffer from limitations due to the anti-replay mechanism.

As a result, this section describes a solution which modifies the anti-replay mechanism by allowing to split the 64 bits (with Extended Sequence Number, ESN) anti-replay sequence number space into multiple subspaces. Each core, path, or QoS class, or any combination of those, can then use their own unique anti-replay sequence number subspace. The changes needed to the ESP header and IPsec protocol are described in Section 4.1, Section 4.2 and Section 4.3.

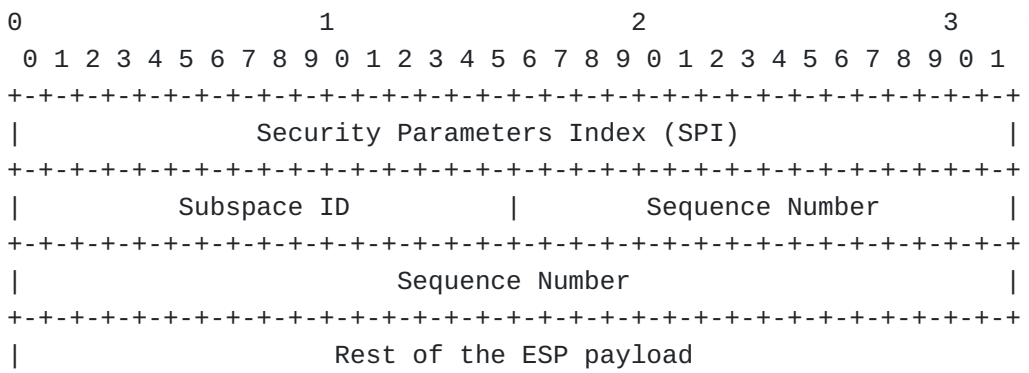
To avoid potential issues with non-standard extensions of IPsec ESP, this solution modifies only the field related to the anti-replay mechanism (i.e., the sequence number) and not the SPI field, which is intended to identify the Child SA. An IKE extension is presented in Section 4.5 to coordinate the use, or not, of this extension, which requires both IPsec peers to implement it.

4.1. Sequence number subspace encoding in IPSec

This document extends the 32-bit field of the sequence number in the ESP header to a 64-bit field, which is in turn divided into two sub-fields:

*The higher order 16 bits contain the new sequence number subspace ID.

*The lower order 48 bits continue to serve as the sequence number.



While the reduced usage of subspaces due to the restriction of the extended sequence number to 48 bits is a consideration, it is estimated that a 1 Tbps would exhaust a subspace in over 938 hours. This is for ethernet frames of 1500 bytes, T = 2^48 (pkts) * 1500 (B/pkt) * 8 / 10^12 (bps) = ~3.4 * 10^6 seconds = ~938 hours.

4.2. Sender Behavior

The sender **MAY** set the sequence number subspace ID to any value. For example, the sender would use different values per path or per processing core.

The sender **MUST** maintain one sequence number counter per sequence number subspace that it makes use of. But the sender **MAY** use only some (and as few as a single one) of the available 65,536 subspaces.

When transmitting a packet, the sender **MUST** use the sequence number counter associated with the sequence number subspace in use for that packet. The lower-order 48 bits of the sequence number counter are placed in the sequence number field, as specified in [RFC4303].

4.3. Receiver Behavior

The receiver **MUST** maintain one anti-replay window and counter for each sequence number subspace being used.

When receiving a packet, the receiver **MUST** use the anti-replay window and counter associated with the sequence number subspace identified with the subspace ID field.

Note: Since the sender may decide to only use a subset of the available subspace values, receivers **SHOULD NOT** allocate all the possible anti-replay windows per peer by default. Two mitigation mechanisms may be used to reduce the number of anti-replay windows:

* The receiver **SHOULD** limit the number of allocated anti-replay windows to the number of subspaces negotiated during the IKE Child SA creation exchange, as specified in Section 4.5.

* The receiver **MAY** reactively allocate an anti-replay window when receiving the first packet for a given subspace, since the sender may decide to not use all of the available values. When doing so, the receiver **SHOULD** first check the authenticity of the packet before allocating the new anti-replay window.

4.4. Extended Sequence Numbers (ESN) considerations

Due to the reduction of the sequence number space by using the 16 higher order bits of the field, using a 32-bit sequence number field is not a possibility. For instance, on a 1Gbps with 1500B ethernet frames, it would take less than one second for the sequence number to loop. Such a small periodicity would make it impractical to keep the peers of the IPsec tunnel in sync.

As such, the peers **MUST** use an explicit Extended Sequence Number (ESN) as a sub-second period for a resync operation (as defined in appendix A3 of [RFC4303]) would not be possible.

4.5. Negotiating sequence-number subspaces using IKE

To negotiate the use of sequence number subspaces for use with IPsec ESP, a new anti-replay subspaces transform (Section 4.5.1) is defined with two attributes:

* The number of sequence number subspaces the sender is capable of using is indicated by the 'Sequence number subspaces supported' attribute, which is 2 bytes long (Section 4.5.2).

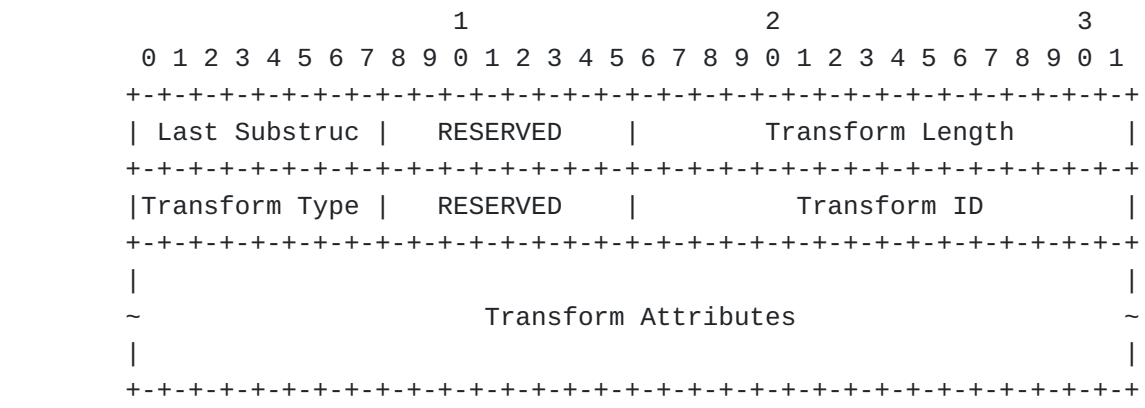
* The 'Sequence number subspaces requested' attribute indicates the number of sequence number subspaces the sender prefers to use, and is also 2 bytes long (Section 4.5.3).

If both attributes are set to 0, the sender does not support sequence number subspaces. The requested value **MUST** be lower than the supported value.

During the CREATE_CHILD_SA exchange, the sender and receiver negotiate the use of this transform. The sender indicates the number of subspaces it supports and prefers to use, while the receiver decides on the number of subspaces to use based on the sender's capabilities. This negotiation mechanism allows for flexibility in the number of subspaces used and can help optimize the performance of IPsec in different environments.

With a single Child SA negotiated between the two IPsec peers, the failure model is clean, as all requested subspaces are either available or none of them.

4.5.1. Anti-replay subspaces transform

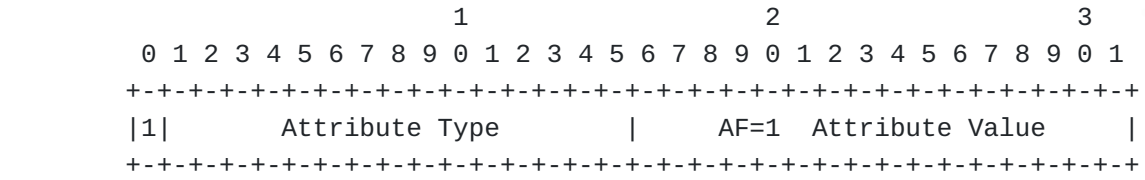


*Transform Length (2 bytes), set to 16 bytes with the two attributes each taking 4 bytes

*Transform Type (1 byte) TBD

*Transform ID (2 bytes) TBD

4.5.2. 'Sequence number subspaces supported' attribute

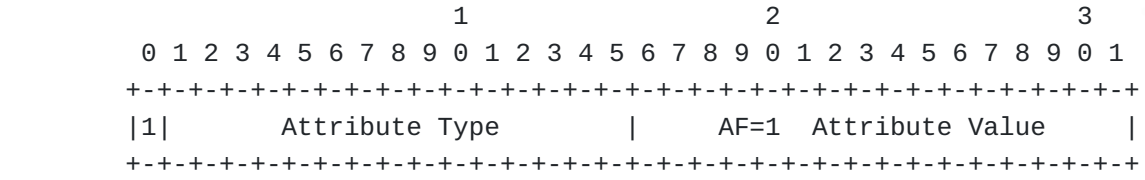


*AF (1 bit), set to 1 for Type/Value (TV) format

*Attribute Type (15 bits) TBD

*Attribute Value (2 bytes), any value between 0 and 65,535

4.5.3. 'Sequence number subspaces requested' attribute



*AF (1 bit), set to 1 for Type/Value (TV) format

*Attribute Type (15 bits) TBD

*Attribute Value (2 bytes), any value between 0 and the supported number of subspaces

4.6. Solution Analysis

As described in Section 2, anti-replay comes with implementation and scalability challenges when running in environments where IPsec peers may leverage multiple QoS policies to send packets or multiple cores to process them.

Since the anti-replay mechanism seems to be the source cause of these observed challenges, this document provides a solution which relies on a small and optional change at the anti-replay level.

By using sequence number subspaces, IPsec peers may:

* use different subspaces for different cores, which allows distributing a Child SA between cores to increase performance

* use different subspaces for different QoS classes or different paths, which avoids unwanted drops due to potential reordering of packets, either at the egress or during its flight.

* combine the above per-QoS-queue, per-path and per-core approaches without multiplying the number of required Child SAs.

The effectiveness of the subspace mechanism can be further improved with smart NICs or multiple paths to efficiently steer packets to different cores on the receiver side. However, even without these capabilities, sequence number subspaces still provide benefits for IPsec tunnels. Without subspaces, IPsec tunnels are often restricted to a single core due to the need for locking mechanisms, which can cause significant overhead. With subspaces, it is still possible to distribute the subspaces between cores by resteering packets to increase performances.

In scenarios where NATs are used to modify IP addresses or ports, the use of multiple uplinks on a single IPsec tunnel may not be feasible without additional IKE negotiation to perform NAT traversal. As a result, using multiple uplinks is recommended only in scenarios where NATs are not present.

5. Security Considerations

The sequence number is used by the anti-replay mechanism to ensure a packet could not be accepted twice by the receiver. This prevents an attacker from trying to replay one or multiple packets from an IPsec tunnel.

In this proposal, a single Child SA is associated with multiple anti-replay windows and counters. If a packet is replayed, the sequence number subspace ID remains the same since the Subspace ID field is authenticated. As a result, the receiver will use the same

anti-replay state when processing the replayed packet as the one used when the first packet was initially received. This ensures that a replayed packet will be detected and dropped by the receiver.

The use of a subspace ID as part of the 64-bit sequence number ensures that the usage limit of cryptographic materials is evenly distributed among the subspaces without the need for an additional mechanism. This means each of the 2^16 subspaces can encrypt 2^48 packets, fully utilizing the 2^64 usage limits of the cryptographic keys.

6. Implementation Considerations

When a single sequence number space is used within a given Child SA, encryption and decryption operations must always happen on the same core (locking anti-replay structures or using contended atomic operations has a dramatic performance hit).

- On reception, this requires packets which are received (and load-balanced to cores) to be often resteered to a different thread for processing.

- On transmisson, multiple flows, processed by different cores, need to be transmitted using the same Child SA. This requires the packets to be resteered to the thread in charge of the given Child SA.

To avoid the performance degradation caused by packet resteering, each thread may use its own sequence number subspace:

- On transmission, the core will always select the subspace it is assigned when generating the ESP header.

- On reception, the subspace ID could be used to load-balance the packets to their proper thread.

Similarly, when multiple paths are used:

- On transmission, a different sequence number subspace is used for each packet path. Ensuring that out-of-order packets are not dropped by the anti-replay mechanism.

- On reception, the 5-tuple based packet steering would provide a decent level of load-balancing between threads, since different IP paths would use different 5-tuples.

If a combination of both multi-path and multi-core load-balancing is needed, the subspace field could be used partly to encode a path ID, partly to encode a core ID. But this is purely implementation specific and does not require coordination between the peers.

6.1. Initialization Vector (IV) Considerations

Depending on the cryptographic mode of operations, the Initialization Vector (IV) comes with specific requirements.

Some modes (e.g., CBC) make use of random IV values. When implementing this specification, each thread independently generates its independent stream of random values, ensuring the IV randomness property. Care must be taken as to limit the global number of transmitted packets using the same Child SA in order to avoid birthday paradox attacks. A lockless counter, or batched token bucket mechanism, may be used to efficiently implement this process without performance degradation.

Other cryptographic modes (e.g., GCM) do not have randomness requirements over the IV, but the IV values must only be used once. RFC4106 Section 3.1 states that "The most natural way to implement this is with a counter, but anything that guarantees uniqueness can be used, such as a linear feedback shift register (LFSR). Note that the encrypter can use any IV generation method that meets the uniqueness requirement, without coordinating with the decrypter." . One simple way to implement this specification is to divide the IV into a subspace field, which reuses the ESP sequence number subspace value, and a variable IV part, which is simply incremented for each encrypted packet. To ensure compatibility with implicit IVs from [RFC8750], only the 48-bit sequence number field must be initialized to zero, while the 16-bit subspace ID can be used for its intended purpose.

Author's note: Are there other cryptographic modes with different requirements over the IV ?

7. Operational Considerations

TBD

8. IANA Considerations

TBD

9. References

9.1. 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>.

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

[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/info/rfc8174>.

[RFC8750] Migault, D., Guggemos, T., and Y. Nir, "Implicit Initialization Vector (IV) for Counter-Based Ciphers in Encapsulating Security Payload (ESP)", RFC 8750, DOI 10.17487/RFC8750, March 2020, <https://www.rfc-editor.org/info/rfc8750>.

9.2. Informative References

[I-D.mrossberg-ipsecme-multiple-sequence-counters]
Rossberg, M., Klassert, S., and M. Pfeiffer, "Problem statements and uses cases for lightweight Child Security Associations", Work in Progress, Internet-Draft, draft-mrossberg-ipsecme-multiple-sequence-counters-00, 27 February 2023, <https://datatracker.ietf.org/doc/html/draft-mrossberg-ipsecme-multiple-sequence-counters-00>.

[I-D.pwouters-ipsecme-multi-sa-performance]
Antony, A., Brunner, T., Klassert, S., and P. Wouters, "IKEv2 support for per-queue Child SAs", Work in Progress, Internet-Draft, draft-pwouters-ipsecme-multi-sa-performance-05, 8 November 2022, <https://datatracker.ietf.org/doc/html/draft-pwouters-ipsecme-multi-sa-performance-05>.

Authors' Addresses

Paul Ponchon
Cisco Meraki

Email: pponchon@cisco.com

Mohsin Shaikh
Cisco Meraki

Email: mohsisha@cisco.com

Pierre Pfister
Cisco Meraki

Email: ppfister@cisco.com

Guillaume Solignac
Cisco Meraki

Email: gsoligna@cisco.com