Light-Weight Implementation Guidance (lwig)
Internet-Draft
Intended status: Standards Track
Expires: January 3, 2015

D. Migault (Ed)
Orange
T. Guggemos
Orange / LMU Munich
D. Palomares
Orange / LIP6

July 2, 2014

Minimal ESP
draft-mglt-lwig-minimal-esp-01.txt

Abstract

This document describes a minimal version of the IP Encapsulation Security Payload (ESP) described in RFC 4303 which is part of the IPsec suite.

ESP is used to provide confidentiality, data origin authentication, connectionless integrity, an anti-replay service (a form of partial sequence integrity), and limited traffic flow confidentiality.

This document does not update or modify RFC 4303, but provides a compact description of the minimal version of the protocol. If this document and RFC 4303 conflicts then RFC 4303 is the authoritative description.

Status of This Memo

Copyright Notice

Table of Contents

1. Requirements notation

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 [RFC2119].

2. Introduction

ESP [RFC4303] is part of the IPsec suite protocol [RFC4301] . It is used to provide confidentiality, data origin authentication, connectionless integrity, an anti-replay service (a form of partial sequence integrity), and limited traffic flow confidentiality.

Figure 1 describes an ESP Packet. Currently ESP is implemented in the kernel of IPsec aware devices. This document provides a minimal ESP implementation so that smaller devices like sensor without kernel and with hardware restriction can implement ESP on their own and benefit from IPsec.

For each field of the ESP packet represented in Figure 1 this document provides recommendations and guidance for minimal implementations.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ ----
|               Security Parameters Index (SPI)                 | ^Int.
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |Cov-
|                      Sequence Number                          | |ered
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ | ----
|                    Payload Data* (variable)                   | |   ^
~                                                               ~ |   |
|                                                               | |Conf.
+               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |Cov-
|               |     Padding (0-255 bytes)                     | |ered*
+-+-+-+-+-+-+-+-+               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |   |
|                               |  Pad Length   | Next Header   | v   v
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ ------
|         Integrity Check Value-ICV   (variable)                |
~                                                               ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1: ESP Packet Description

3. Security Parameter Index (SPI) (32 bit)

According to the [RFC4303], the SPI is a mandatory 32 bits field and is not allowed to be removed.

The SPI is used to index the Security Association. The SPI MUST be unique so that any incoming ESP packet can appropriately be bound to its association. Uniqueness of the SPI may be provided by random functions. However, the SPI does not need to be unpredictable. As a result, if random functions are too costly for some constraint devices, the SPI can be generated using predictable functions or even fixed values.

If a constraint device is expected to establish a single ESP connection with a single device, it can use a fix value for the SPI. Similarly, if a constraint device establish a single ESP connection with multiple devices, it may use the IPv4 or the interface ID for IPv6 addresses for example.

Values 0-255 SHOULD NOT be used. Values 1-255 are reserved and 0 is only allowed to be used internal and it must not be send on the wire.

[RFC4303] mentions :

- "The SPI is an arbitrary 32-bit value that is used by a receiver to identify the SA to which an incoming packet is bound. The SPI field is mandatory. [...]"

- "For a unicast SA, the SPI can be used by itself to specify an SA, or it may be used in conjunction with the IPsec protocol type (in this case ESP). Because the SPI value is generated by the receiver for a unicast SA, whether the value is sufficient to identify an SA by itself or whether it must be used in conjunction with the IPsec protocol value is a local matter. This mechanism for mapping inbound traffic to unicast SAs MUST be supported by all ESP implementations."

4. Sequence Number(SN) (32 bit)

According to [RFC4303], the sequence number is a mandatory 32 bits field in the packet.

The SN is set by the sender so the receiver can implement anti-replay protection. The SN is derived from any strictly increasing function that guarantees: if packet B is sent after packet A, then SN of packet B is strictly greater then the SN of packet A.

In IoT, constraint devices are expected to establish communication with specific devices, like a specific gateway, or nodes similar to them. As a result, the sender may know whereas the receiver implements anti-replay protection or not. Even though the sender may know the receiver does not implement anti replay protection, the sender MUST implement a always increasing function to generate the SN.

Usually, SN is generated by incrementing a counter for each packet sent. A constraint device may avoid maintaining this context. If the device has a clock, it may use the time indicated by the clock has a SN. This guarantees a strictly increasing function, and avoid storing any additional values or context related to the SN.

[RFC4303] mentions :

- "This unsigned 32-bit field contains a counter value that increases by one for each packet sent, i.e., a per-SA packet sequence number. For a unicast SA or a single-sender multicast SA, the sender MUST increment this field for every transmitted packet. Sharing an SA among multiple senders is permitted, though generally not recommended. [...] The field is mandatory and MUST always be present even if the receiver does not elect to enable the anti-replay service for a specific SA."

5. Next Header (8 bit)

According to [RFC4303], the Next Header is a mandatory 8 bits field in the packet.

[RFC4303] mentions :

- "The Next Header is a mandatory, 8-bit field that identifies the type of data contained in the Payload Data field, e.g., an IPv4 or IPv6 packet, or a next layer header and data. [...] the protocol value 59 (which means "no next header") MUST be used to designate a "dummy" packet. A transmitter MUST be capable of generating dummy packets marked with this value in the next protocol field, and a receiver MUST be prepared to discard such packets, without indicating an error."

6. ICV

The ICV is an optional value with variable length. Although optional, we recommend strongly to use the ICV. Furthermore, [RFC4303] allows combined encryption and authentication ciphers, which enables the use of modes like GCM [RFC4106] , CCM and AES-CTR which make ICV mandatory.

IoT devices may allow weak security by removing the ICV, and gateways wanting to connect to IoT devices SHOULD be able to deal with NULL authentication.

[RFC4303] mentions :

- "The Integrity Check Value is a variable-length field computed over the ESP header, Payload, and ESP trailer fields. Implicit ESP trailer fields (integrity padding and high-order ESN bits, if applicable) are included in the ICV computation. The ICV field is optional. It is present only if the integrity service is selected and is provided by either a separate integrity algorithm or a combined mode algorithm that uses an ICV. The length of the field is specified by the integrity algorithm selected and associated with the SA. The integrity algorithm specification MUST specify the length of the ICV and the comparison rules and processing steps for validation."

7. Encryption

[RFC4303] specifies AES in CBC mode [RFC3602] as mandatory for implementing ESP. For maximum interoperability with any gateway, it is recommended to implement AES in CBC mode.

AES and CBC mode has a 128 bit alignment which for small packets of a few bytes length generates a large overhead in term of extra padding bytes. For constraint devices especially those relying on battery, sending these unnecessary bytes may be avoided as it reduces there life time. The AES-CBC mode may then be abandoned, and modes like CCM [RFC4309] or AES CTR [RFC3686] may be used instead. These modes are not mandatory, but are part of the main IPsec distributions. As these IoT devices do not require to be accessed by all nodes on the Internet, implementation of AES-CTR only or CCM only may be considered.

8. Padding

[RFC4303] does not specify any way on how Padding bytes should be generated. These bytes may for example, be generated randomly or each byte may be numbered from \x01 to \xpad-length. A simplified implementation may consider a fix value, and consider all Padding bytes set to zero.

Note that Padding can also be defined by the encryption algorithm like AES in CBC mode [RFC3602]. In that case, Padding MUST be performed as described in [RFC3602]. However, [RFC3602] does not specify how Padding bytes are generated, and AES in CTR or GCM or CCM mode do not consider Padding. As a result, currently, setting padding byte to zero can be proceeded.

9. IANA Considerations

There are no IANA consideration for this document.

10. Security Considerations

Security considerations are those of [RFC4303].

Using a fix value for SPI may isolate the device, as it will not be able to set a communication with the peer if that SPI value is not available.

11. Acknowledgment

12. Normative References

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

[RFC3602] Frankel, S., Glenn, R., and S. Kelly, "The AES-CBC Cipher Algorithm and Its Use with IPsec", RFC 3602, September 2003.

[RFC3686] Housley, R., "Using Advanced Encryption Standard (AES) Counter Mode With IPsec Encapsulating Security Payload (ESP)", RFC 3686, January 2004.

[RFC4106] Viega, J. and D. McGrew, "The Use of Galois/Counter Mode (GCM) in IPsec Encapsulating Security Payload (ESP)", RFC 4106, June 2005.

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.

[RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, December 2005.

[RFC4309] Housley, R., "Using Advanced Encryption Standard (AES) CCM Mode with IPsec Encapsulating Security Payload (ESP)", RFC 4309, December 2005.

Appendix A. Document Change Log

[RFC Editor: This section is to be removed before publication]

-00: First version published.

Authors' Addresses

Daniel Migault
Orange
38 rue du General Leclerc
92794 Issy-les-Moulineaux Cedex 9
France

Phone: +33 1 45 29 60 52
Email: daniel.migault@orange.com

Tobias Guggemos
Orange / LMU Munich
Am Osteroesch 9
87637 Seeg, Bavaria
Germany

Email: tobias.guggemos@gmail.com

Daniel Palomares
Orange / LIP6
10, Rue du Moulin
92170 Vanves, Ille-de-France
France

Email: daniel.palomares@orange.com