SFC
Internet-Draft
Intended status: Standards Track
Expires: October 11, 2015

T. Reddy
P. Patil
S. Fluhrer
P. Quinn
Cisco

April 9, 2015

Authenticated and encrypted NSH service chains
draft-reddy-sfc-nsh-encrypt-00

Abstract

This specification adds data origin authentication and optional encryption directly to Network Service Headers (NSH) used for Service Function Chaining.

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 http://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 October 11, 2015.

Copyright Notice

Table of Contents

1. Introduction

Service function chaining (SFC) [I-D.ietf-sfc-architecture] involves steering traffic flows through a set of service functions in a specific order, such an ordered list of service functions is called a Service Function Chain (SFC). The actual forwarding path used to realize an SFC is called the Service Function Path (SFP). Network Service Headers (NSH) [I-D.ietf-sfc-nsh] provides a mechanism to carry metadata between service functions. The NSH structure is defined in [I-D.ietf-sfc-nsh] and NSH data can be divided into two parts:

- Path information used to construct the SFP.
- Metadata carrying the information about the packets being chained

NSH data is unauthenticated and unencrypted, forcing a service topology that requires security to use a transport encapsulation that support such features (e.g. IPsec). This draft adds authentication and optional encryption directly to NSH. This way NSH data does not have to rely on underlying transport encapsulation for security and confidentiality.

This specification introduces new TLVs to carry fields necessary for Authenticated and Encrypted NSH, and is hence only applicable to NSH MD-Type 2 defined in [I-D.ietf-sfc-nsh].

2. Notational Conventions

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].

This note uses the terminology defined in [I-D.ietf-sfc-problem-statement].

2.1. Definitions and Notation

KMS: Key Management Service.

Ticket: A Kerberos like object used to identify and deliver keys over an untrusted network.

NSH imposer: Imposes NSH header including Service Path ID, Service Index and metadata.

SF : Service function.

3. Design considerations

SFC [I-D.ietf-sfc-architecture] removes the constraint of strict ordering of service functions and allows dynamic ordering of service functions. Service function paths (SFP) could vary for different traffic and it is not possible to pre-determine peer service functions in service function paths and pre-distribute credentials for security association between all possible combinations of peer service functions for authentication and encryption of NSH data.

The keying material should be unique for each SFP so that only the authorized service functions participating in the SFP can act on the NSH data. A trusted KMS can be used to propagate keying material to authorized service functions as and when needed and avoids the use of pair-wise keys. A KMS based on symmetric keys has particular advantages, as symmetric key algorithms are generally much less computationally intensive than asymmetric key algorithms and the size of cipher-text generated using symmetric key algorithm is smaller compared to the cipher-text generated using asymmetric encryption algorithm. Systems based on a KMS require a signaling mechanism that allows peers to retrieve other peers dynamic credentials. A convenient way to implement such a signaling scheme is to use a ticket concept, similar to that in Kerberos [RFC4120] to identify and

deliver keys. The ticket can be forwarded in the NSH data. The NSH imposer requests a ticket from the KMS and sends the ticket in NSH data. The service function in SFP gets the ticket from NSH, requests KMS to provide the keying material associated with the ticket. If the service function is authorized then KMS returns the corresponding keys.

Note: Key management services may introduce a single point of (security) failure. The security requirements on the implementation and protection of the KMS may therefore, in high-security applications, be more or less equivalent to the requirements of an AAA (Authentication, Authorization, and Accounting) server or a Certification Authority (CA).

KMS is used in GDOI [[RFC6407]], MIKEY-TICKET [[RFC6043]], end-to-end encryption key management service [I-D.abiggs-saag-key-management-service] etc.

4. Overview

The service functions do not share any credentials; instead, they trust a third party, the KMS, with which they have or can establish shared credentials. These pre-established trust relations are used to establish a security association between service functions.

The NSH imposer requests keys and a ticket from the KMS. The request message also includes identities of the service functions authorized to receive the keying material associated with the ticket. Each SF is referenced using an identifier that is unique within an SF-enabled domain. If the request is authorized then KMS generates the encryption and message integrity keys (referred to as ENC and MAC keys), ticket, and returns them in a response message. The ticket could be self-contained (key encrypted in the ticket) or just a handle to some internal datastructure within the KMS. The need to encrypt NSH metadata is determined based on the classification decision and the metadata conveyed in NSH. The encryption and authentication algorithms will either be negotiated between the NSH imposer and KMS or determined by the KMS and conveyed to the NSH imposer.

The NSH imposer includes the ticket in NSH data. The NSH data is protected using the MAC key and optionally NSH metadata is encrypted using the ENC key. Service functions in the SFP forward the ticket to the KMS and request the KMS to provide the keying material. If the service function is authorized and the ticket is valid then the KMS retrieves the keys and algorithms associated with the ticket and conveys them to the service function. The other alternative

technique is that KMS implicitly pushes the keying material to service functions authorized by the NSH imposer.

If the SFP for a flow changes then NSH imposer requests new keys and a new ticket from KMS. The request message from NSH imposer to KMS includes identities of the service functions authorized to receive the keying material associated with the new ticket. For subsequent packets of the flow the new ticket will be conveyed in the NSH data, NSH data will be integrity protected using the new MAC key and NSH metadata encrypted using the new ENC key.

Figure 1 shows an example of NSH imposer requesting keys and ticket from the KMS. The request message includes identifiers of SF1 and SF2 service functions authorized to receive keying material associated with the ticket. KMS returns the ENC key, MAC key and ticket in the response message. The NSH imposer includes the ticket in the NSH data. SF1 in the SFP forwards the ticket to the KMS and requests the KMS for keying material associated with the ticket (In Ticket resolve request message). If SF1 is authorized and the ticket is valid then KMS retrieves the keys and algorithms associated with the ticket and conveys them to the SF1 (In Resolve response message). Similarly, SF2 retrieves the keying material associated with the ticket from KMS.

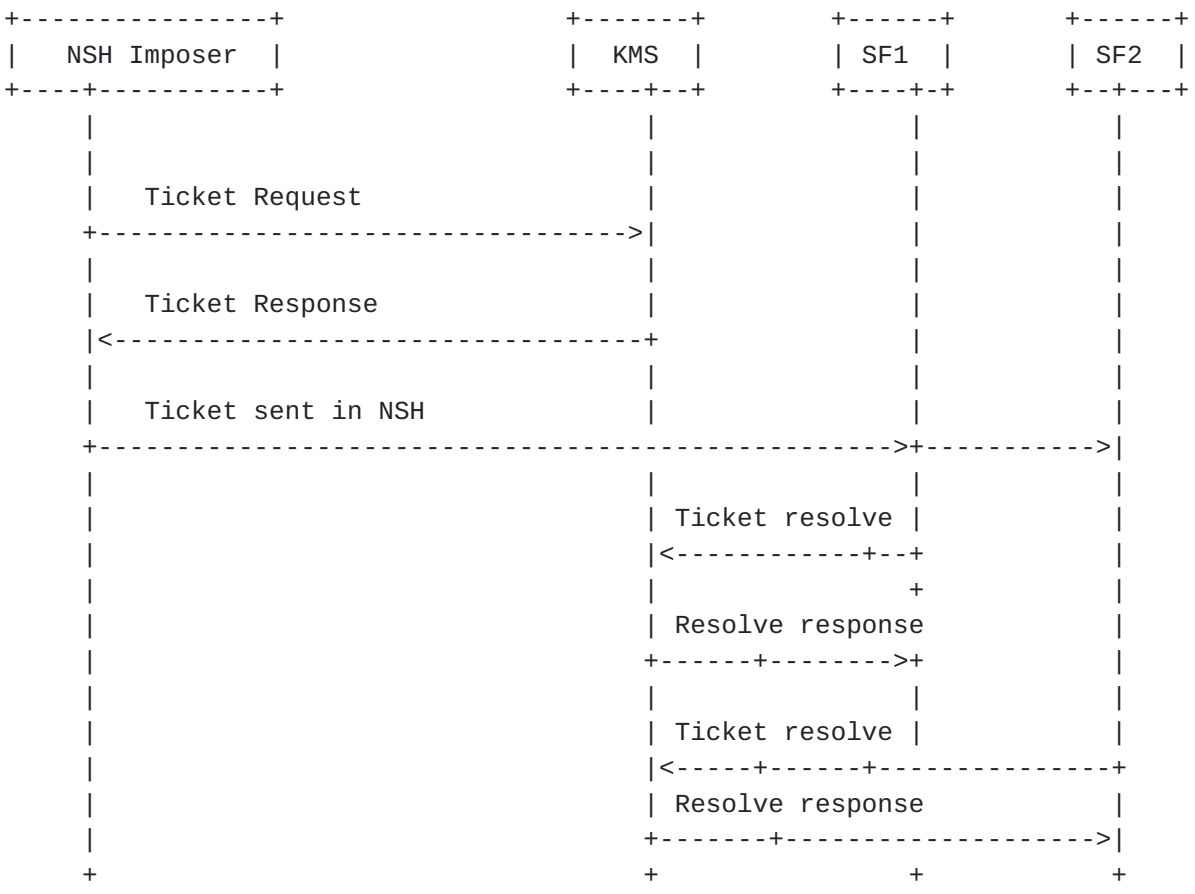
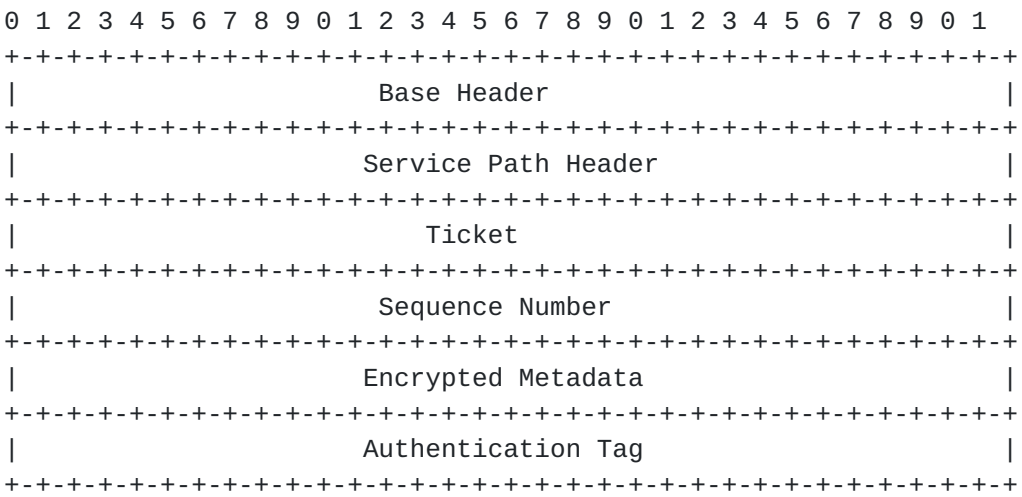


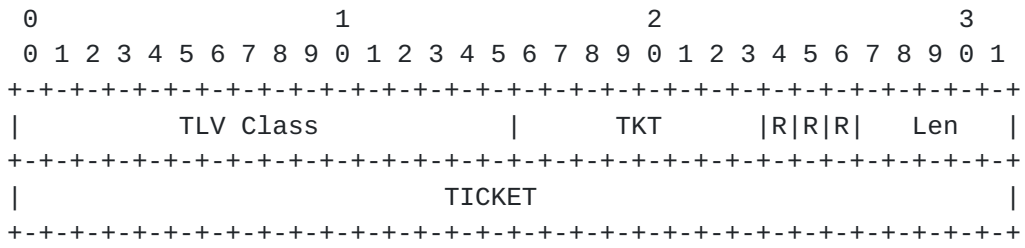
Figure 1: Interaction with KMS

5. NSH Format



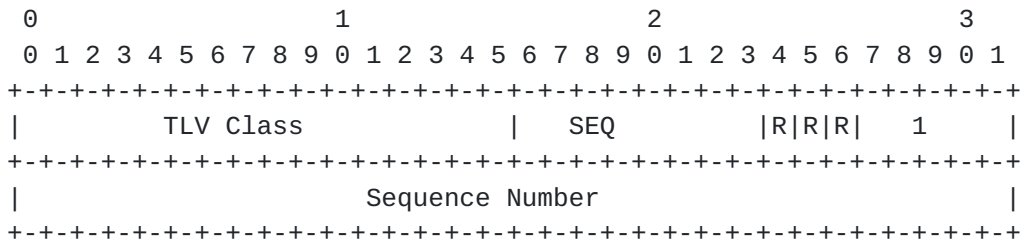
5.1. Ticket TLV

A variable length Kerberos-like object used to identify and deliver keys over an untrusted network to service functions. This is a mandatory TLV that MUST be present if an authenticated and encrypted NSH solution is desired.



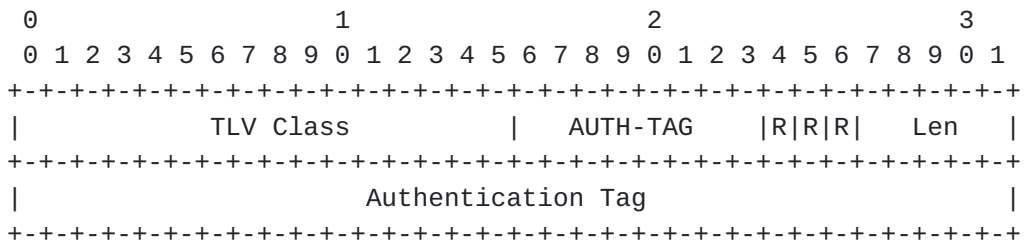
5.2. Sequence Number TLV

A 32-bit sequence number per ticket. In this solution, a sequence number needs to be incremented every time NSH is included by the NSH imposer. The number should not be incremented if an existing NSH is being updated. This is a mandatory TLV that MUST be present if an authenticated and encrypted NSH solution is desired.



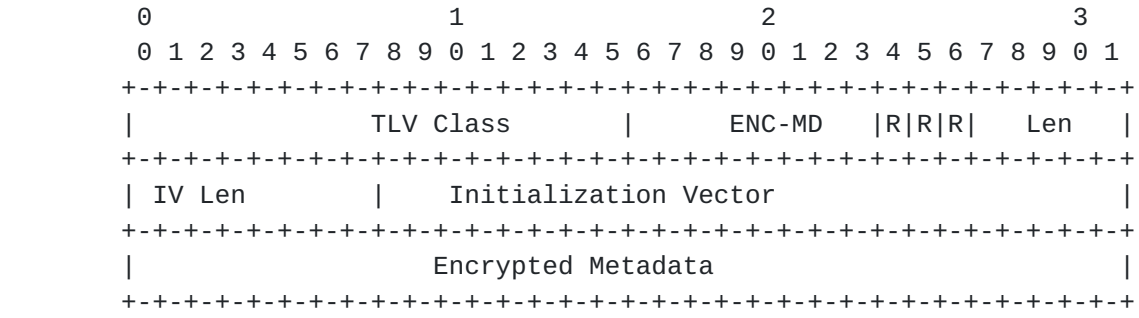
5.3. Authentication Tag TLV

A variable-length TLV that carries the hash based Message Authentication Codes for the entire NSH calculated using the MAC key. If Authenticated Encryption with Associated Data (AEAD) algorithm defined in [RFC5116] is used then there is no need explicitly compute HMAC and include this TLV.



5.4. Encrypted Metadata

A variable-length TLV that carries the metadata encrypted using ENC key obtained from the KMS. The C bit in the Type field MUST be set to 1 indicating that the TLV is mandatory for the receiver to understand and process.



Randomly generated Initialization Vector (IV) prevents generation of identical cipher-text from packets which have identical metadata, use of IV in AES CBC is explained in [RFC3602].

If AEAD algorithm is used

- The Initialization Vector field will carry the nonce and the length of the nonce for AEAD algorithms is specified in [RFC5116].

- The associated data MUST be the entire NSH data excluding the metadata to be encrypted and the nonce value.

If one or more service functions in the SFP are authorized to validate the message integrity of NSH data and update the unencrypted NSH data but not decrypt the encrypted metadata then AEAD algorithm MUST NOT be used and these service functions MUST only be given access to the MAC key.

6. Processing rules

The following sections describe processing rules for authenticated and encrypted NSH.

6.1. Encrypted NSH metadata Generation

An NSH imposer can encrypt all NSH metadata or only a subset of metadata i.e., encrypted and unencrypted metadata may be carried simultaneously. Using the ENC key and encryption algorithm obtained from the KMS, the imposer encrypts metadata of choice and inserts the resulting payload in the encrypted metadata TLV.

An authorized entity in the service path that intends to update encrypted metadata, MUST also do the above.

If NSH encryption is desired, encryption is performed first, before the integrity algorithm is applied. This order of processing facilitates rapid detection and rejection of bogus packets by the receiver, prior to decrypting the metadata, hence potentially reducing the impact of denial of service (DoS) attacks.

6.2. Authenticated NSH data Generation

An NSH imposer inserts an Authentication Tag TLV for data origin authentication and integrity protection. After requesting ENC and MAC keys from the KMS, the imposer computes the message integrity for the entire NSH data using the MAC key and HMAC algorithm. It inserts the result in the AUTH-TAG TLV. The length of the Authentication Data field is decided by the HMAC algorithm adopted for the particular ticket.

An entity in the service function path that intends to update NSH MUST do the above to maintain message integrity of the NSH for subsequent validations.

6.3. Sequence number validation for replay attack

A Sequence Number is an unsigned 32-bit counter value that increases by one for each NSH created and sent from the NSH imposer, i.e., a per-ticket packet sequence number. The field is mandatory and MUST always be present. Processing of the Sequence Number field is at the discretion of the receiver, but all implementations MUST be capable of validating that the Sequence Number that does not duplicate the Sequence Number of any other NSH received during the life of the ticket.

The NSH imposer's counter is initialized to 0 when a new ticket is to be used . The sender increments the Sequence Number counter for this ticket and inserts the 32-bit value into the Sequence Number TLV. Thus the first NSH sent using a given ticket will contain a Sequence Number of 1. The imposer checks to ensure that the counter has not cycled before inserting the new value in the Sequence Number TLV. In other words, the sender MUST NOT send a packet on a ticket if doing so would cause the Sequence Number to rollover. Sequence Number counters of all participating nodes MUST be reset by establishing a new ticket prior to the transmission of the 2^32nd packet of NSH for a particular ticket.

6.4. NSH data Validation

When an SFC node receives an NSH header with encrypted metadata, it MUST first ensure that all mandatory TLVs required for NSH data authentication exist. The node MUST discard NSH if mandatory TLVs are absent or if the sequence number is invalid (described in Section 6.3). The node should then go on to do data validation. The node calculates the message integrity for the entire NSH data using the MAC key and HMAC algorithm obtained from the KMS for the ticket being carried in NSH. If the value of the newly generated digest is identical to the one in NSH, the node is certain that the header has not been tampered and validation succeeds. Otherwise, the NSH MUST be discarded.

6.5. Decryption of NSH metadata

After NSH data validation is complete, an SFC node decrypts metadata using the ENC key and decryption algorithm obtained from the KMS for the ticket in NSH. If AEAD algorithm is used then it has only a single output, either a plaintext or a special symbol FAIL that indicates that the inputs are not authentic.

7. IANA Considerations

TODO

8. Security Considerations

The interaction between the Service functions and the KMS requires Transport Layer Security (TLS) with a ciphersuite offering confidentiality protection. The ENC and MAC keys MUST NOT be transmitted in clear since this would completely destroy the security benefits of the proposed scheme.

9. Acknowledgements

Authors would like to thank Dan Wing and Jim Guichard for their comments and review.

10. References

10.1. Normative References

[I-D.ietf-sfc-architecture]
Halpern, J. and C. Pignataro, "Service Function Chaining (SFC) Architecture", draft-ietf-sfc-architecture-07 (work in progress), March 2015.

[I-D.ietf-sfc-nsh]
Quinn, P. and U. Elzur, "Network Service Header", draft-ietf-sfc-nsh-00 (work in progress), March 2015.

[I-D.ietf-sfc-problem-statement]
Quinn, P. and T. Nadeau, "Service Function Chaining Problem Statement", draft-ietf-sfc-problem-statement-13 (work in progress), February 2015.

[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.

[RFC4120] Neuman, C., Yu, T., Hartman, S., and K. Raeburn, "The Kerberos Network Authentication Service (V5)", RFC 4120, July 2005.

[RFC5116] McGrew, D., "An Interface and Algorithms for Authenticated Encryption", RFC 5116, January 2008.

[RFC6043] Mattsson, J. and T. Tian, "MIKEY-TICKET: Ticket-Based Modes of Key Distribution in Multimedia Internet KEYing (MIKEY)", RFC 6043, March 2011.

[RFC6407] Weis, B., Rowles, S., and T. Hardjono, "The Group Domain of Interpretation", RFC 6407, October 2011.

10.2. Informative References

[I-D.abiggs-saag-key-management-service]
Biggs, A. and S. Cooley, "Key Management Service Architecture", draft-abiggs-saag-key-management-service-00 (work in progress), November 2014.

Authors' Addresses

Tirumaleswar Reddy
Cisco Systems, Inc.

Email: tireddy@cisco.com

Prashanth Patil
Cisco Systems, Inc.

Email: praspati@cisco.com

Scott Fluhrer
Cisco Systems, Inc.

Email: sfluhrer@cisco.com

Paul Quinn
Cisco Systems, Inc.

Email: paulq@cisco.com