Network Working Group
Internet-Draft
Intended status: Standard Track

L. Yong
Huawei USA
T. Herbert
Facebook

Expires: November 2015
May 20, 2015

Generic UDP Encapsulation (GUE) for Secure Transport
draft-hy-gue-4-secure-transport-02

Abstract

This document specifies the ability of Generic UDP Encapsulation (GUE) [GUE] to provide secure transport over IP networks and the Internet, including GUE header integrity protection and origin authentication and GUE payload encryption. These are optional features of GUE.

Status of This Document

Copyright Notice

Table of Contents

1. Introduction

Generic UDP Encapsulation [GUE] is the protocol that specifies tunneling a network protocol over an IP network or Internet and a UDP tunnel. The tunneled network protocol is encapsulated in GUE header [GUE] at a tunnel encapsulator, transported as a regular IP packet, and decapsulated at the tunnel decapsulator.

This draft specifies the security capabilities for GUE. One security capability is to provide origin authentication and integrity protection of the GUE header at tunnel end points to guarantee isolation between tunnels and mitigate Denial of Service attacks. Another capability is payload encryption that prevents the payload from eavesdropping, tampering, or message forgery. These security capabilities are specified as optional features of GUE.

In theory, uses of GUE could leverage other existing tunnel mechanisms that provides secure transport over Internet such as DTLS [RFC6347] and IPsec[RFC4301]. Section 6 discusses the weakness to rely on another tunnel mechanism for GUE secure transport.

2. Terminology

The terms defined in GUE [GUE] are used in this document.

2.1. Requirements Language

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].

3. GUE Security

This document proposes to allocate two flag bits from GUE optional flag field as the Security flag for GUE integrity protection and authentication validation. GUE header format with Security Flag is shown in Figure 1. The second and third bits in GUE optional flag field are allocated for Security mechanism.

```
     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
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |        Source port            |      Destination port         |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |           Length              |          Checksum             |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |Ver|C|  Hlen   |  Proto/Ctype  |V|SEC|  Undefined Flags      |E|
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                Virtual Network ID (Optional)                  |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    ~                    Security Field                             ~
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1 GUE Header Format with Security Flag

- 'V' Virtualization flag: This flag is designated for network virtualization overlay (NVO)[GUE4NVO].

- 'SEC' Security flags: Indicates presence of security field. To provide security capability, the flags MUST be set. Different sizes are allowed to allow different methods and extensibility. The use of the security field is expected to be negotiated out of band between two tunnel end points. Potential uses of the security field are discussed in Section of Security Considerations.

  - 00 - No security field

  - 01 - 64 bit security field

  - 10 - 128 bit security field

  - 11 - 256 bit security field

The usage of the key fields in the GUE header [GUE] for the security mechanism is described as below:

- o Ver: Set to 0x0. Security option is designed for GUE version 0.

- o 'C' Control flag: When it set, control message presents and control processing MUST occur after security validation.

- o Hlen: length of optional fields (byte)/4. Note that Payload Transform function does not require a private field.

- o Proto/ctype: Contain the protocol of the encapsulated payload packet, i.e. next header when the C bit is not set. Contains a control message type when the C bit is set. The next header begins at the offset provided by Hlen.

'E' Extension flag. It is the last bit in the GUE header. For usage of Extension field see [GUE]. The security transport does not require use of any extension field.

UDP header field must be set per [GUE]. The checksum and length implementation MUST be compliant with GUE implementation [GUE].

4. GUE Payload Encryption

The payload of a GUE packet can be secured using Datagram Transport Layer Security [RFC6347]. An encapsulator would apply DTLS to the GUE payload so that the payload packets are encrypted and the GUE header remains in plaintext.

To differ encrypted payload from plaintext payload, the document proposes allocating one flag from GUE optional flag field for payload transformation indication and adding a 32 bit field when Payload Transform flag is set. Following format shows GUE header with Payload Transform flag, i.e. 'T' is set.

```
  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
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |        Source port            |      Destination port         |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |           Length              |          Checksum             |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |Ver|C|  Hlen   |  Proto/Ctype  |V|SEC|K|F|S|T|     Flags     |E|
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                                                               |
 |     VNID, security, GUE checksum, fragmentation,              ~
 ~               Session (optional)                              |
 |                                                               |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                    Payload Transform Field                    |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2 GUE Header with Payload Transform Flag

A 32 bits field in GUE header is for the payload transform function and MUST be presented when Payload Transform flag T is set and MUST NOT be presented when clear. The format of Payload Transform Field is in Figure 3.

```
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |    Type       | Payload Type  |            Reserved           |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3 Payload Encryption Field Format

Type: Payload Transform Type or Code point. Each payload transform mechanism must have one code point registered in IANA. This document specifies:

0x01: for DTLS [RFC6347]

0x80~0xFF: for private payload transform types

A private payload transform type can be used for experimental purpose or vendor proprietary mechanism.

Payload Type: used to indicate the encrypted payload type. When encryption flag is set, next protocol in the base header should set to 59 ("No next header") for a data message and zero for a control message. The payload type (IP protocol or control message type) of the unencrypted paytload must be encoded in this field.

The benefit of this rule is to prevent a middle box from inspecting the encrypted payload according to GUE next protocol. Assumption here is that a middle box may understand GUE base header but does not understand GUE option flag definitions.

Reserved field for DTLS type MUST set to Zero. For other transformation type, the reserved field may be specified for a purpose.

The usage of the key fields in the GUE header [GUE] for the payload encryption mechanism is described as below:

- Ver: Set to 0x0. Payload transform option applies to version 0x0.

- Control flag: When it set, control message presents SHOULD be encrypted except GUE base header.

- Hlen: length of optional fields (byte)/4

- Proto/CType: Set to 59. The payload type is encoded in the payload encryption option field.

- 'E' Extension flag. It is the last bit in the GUE header. For usage of Extension field see [GUE]. The payload encryption does not require use of any extension field.

UDP header usage for payload encryption mechanism: UDP dst port SHOULD be filled with GUE port [GUE]; UDP src port MAY be filled with entropy or a random value. The checksum and length implementation MUST be compliant with GUE implementation [GUE].

5. Encapsulation/Decapsulation Operations

GUE secure transport mechanism applies to both IPv4 and IPv6 underlay networks. The outer IP address MUST be tunnel egress IP address (dst) and tunnel ingress IP address (src). GUE security option provides origin authentication and integrity to GUE based tunnel; GUE payload encryption provides payload privacy over an IP delivery network or Internet. Two functions are processed separately at tunnel end points. A GUE tunnel can use both functions or use one of them.

When both encryption and security are required, an encapsulator must perform payload encryption first and then encapsulate the encrypted packet with security flag and encryption flag set in GUE header; the security field must be filled according Section 3 above; the encryption field must be filled according to Section 4 above; the decapsulator must decapsulate the packet first, then perform the authentication validation; if the validation is successful, it performs the payload decryption according the encryption information in the payload encryption field in the header; if the validation fails, the decapsulator must discard the packet and may generate an alert to the management system. These processing rules also apply when only one function, either encryption or security, is enabled, except another function is not performed as stated above.

If GUE fragmentation is used in concert with the GUE security option and/or payload transform option, the security and playload transformation are performed after fragmentation at the encapsulator and before reassembly at the decapsulator.

In order to get flow entropy from the payload, the encapsulator needs to get the flow entropy first before performing the payload encryption; then the flow entropy is inserted into UDP src port in the GUE header.

DTLS [RFC6347] provides packet fragmentation capability. To avoid packet fragmentation performed multiple times at GUE encapsulator, GUE encapsulator SHOULD only perform the packet fragmentation at

packet encapsulation process, i.e., not in payload encryption process. The encryption process should apply to GUE control packets.

DTLS usage [RFC6347] is limited to a single DTLS session for any specific tunnel encapsulator/decapsulator pair (identified by source and destination IP addresses). Both IP addresses MUST be unicast addresses - multicast traffic is not supported when DTLS is used. A GUE tunnel decapsulator implementation that supports DTLS can establish DTLS session(s) with one or multiple tunnel encapsulators, and likewise a GUE tunnel encapsulator implementation can establish DTLS session(s) with one or multiple decapsulators.

6. Considerations of Using Other Security Tunnel Mechanisms

GUE may rely on other secure tunnel mechanisms such as DTLS [RFC6347] for securing the whole GUE packet or IPsec [RFC4301] to achieve the secure transport over an IP network or Internet.

IPsec [RFC4301] was designed as a network security mechanism, and therefore it resides at the network layer. As such, if the tunnel is secured with IPsec, the UDP header would not be visible to intermediate routers anymore in either IPsec tunnel or transport mode. The big drawback here prohibits intermediate routers to perform load balance based on the flow entropy in UDP header. In addition, this method prohibits any middle box function on the path.

By comparison, DTLS [RFC6347] was designed with application security and can better preserve network and transport layer protocol information than IPsec [RFC4301]. Using DTLS to secure the GUE tunnel, both GUE header and payload will be encrypted. In order to differentiate plaintext GUE header from encrypted GUE header, the destination port of the UDP header between two must be different, which essentially requires another standard UDP port for GUE with DTLS. The drawback on this method is to prevent a middle box operation to GUE tunnel on the path.

Use of two independent tunnel mechanisms such as GUE and DTLS to carry a network protocol over an IP network adds some overlap and process complex. For example, fragmentation will be done twice.

As the result, a GUE tunnel SHOULD use the security mechanisms specified in this document to provide secure transport over an IP network or Internet when it is needed. GUE tunnel can be used as secure transport mechanism over an IP network and Internet.

7. Security Considerations

Encapsulation of network protocol in GUE should not increase security risk, nor provide additional security in itself. GUE requires that the source port for UDP packets should be randomly seeded to mitigate some possible denial service attacks.

If the integrity and privacy of data packets being transported through GUE is a concern, GUE security and payload encryption SHOULD be used to remove the concern. If the integrity is the only concern, the tunnel may consider use of GUE security only for optimization.

Likewise, if the privacy is the only concern, the tunnel may use GUE encryption function only.

If GUE payload already provides secure mechanism, e.g., the payload is IPsec packets, it is still valuable to consider use of GUE secure mechanisms for the payload header privacy and the tunnel integrity.

7.1. GUE Security Field Usage

The GUE security field should be used to provide integrity and authentication of the GUE header. Security negotiation (interpretation of security field, key management, etc.) is expected to be negotiated out of band between two communicating hosts. Two possible uses for this field are outlined below, a more precise specification is deferred to other documents.

7.1.1. Cookies

The security field may be used as a cookie. This would be similar to cookie mechanism described in L2TP [RFC3931], and the general properties should be the same. The cookie may be used to validate the encapsulation. The cookie is a shared value between an encapsulator and decapsulator which should be chosen randomly and may be changed periodically. Different cookies may used for logical flows between the encapsulator and decapsulator, for instance packets sent with different VNIs in network virtualization [GUE4NVO] might have different cookies.

7.1.2. Secure hash

Strong authentication of the GUE header can be provided using a secure hash. This may follow the model of the TCP authentication option [RFC5925]. In this case the security field holds a message digest for the GUE header (e.g. 16 bytes from MD5). The digest might be done over static fields in IP and UDP headers per negotiation (addresses, ports, and protocols). In order to provide enough

entropy, a random salt value in each packet might be added, for instance the security field could be a 256 bit value which contains 128 bits of salt value, and a 128 bit digest value. The use of secure hashes requires shared keys which are presumably negotiated and rotated as needed out of band.

8. IANA Considerations

This document requires IANA to allocate:

Two bits in GUE option flag field for GUE Security.
One bit in GUE option flag field for GUE Payload Transform.

This document requires IANA to have a new registry for Encryption Type and require two code points for:

0x01: for DTLS [RFC6347]

0x80~0xFF: for private payload transform

9. References

9.1. Normative References

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

[RFC3931] Lau, J., Townsley, W., et al, "Layer Tow Tunneling Protocol - Version 3 (L2TPv3)", RFC3931, 1999

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

[RFC5925] Touch, J., et al, "The TCP Authentication Option", RFC5925, June 2010

[RFC6347] Rescoria, E., Modadugu, N., "Datagram Transport Layer Security Version 1.2", RFC6347, 2012.

[GUE] Herbert, T., Yong, L., et al "Generic UNP Encapsulation", draft-ietf-nvo3-gue-00, work in progress.

9.2. Informative References

[GUE4NVO] Yong, L., and Herbert T., "Generic UNP Encapsulation for

NVO", draft-hy-nvo3-gue-4-nvo, work in progress.

10. Authors' Addresses

Lucy Yong
Huawei USA

Email: lucy.yong@huawei.com

Tom Herbert
Facebook
1 Hacker Way
Menlo Park, CA 94052
US

Email: tom@herbertland.com