IPPM
Internet-Draft
Intended status: Standards Track
Expires: August 29, 2013
Y. Cui
E. Bi
K. Pentikousis, Ed.
Huawei Technologies
February 25, 2013

Network Performance Measurement for IPsec
draft-bi-ippm-ipsec-01.txt

Abstract

IPsec is a mature technology with several interoperable implementations. Indeed, the use of IPsec tunnels is increasingly gaining popularity in several deployment scenarios, not the least in what used to be solely areas of traditional telecommunication protocols. Wider deployment calls for mechanisms and methods that enable tunnel end-users, as well as operators, to measure one-way and two-way network performance. Unfortunately, however, standard IP performance measurement security mechanisms cannot be readily used with IPsec. This document makes the case for employing IPsec to protect O/TWAMP and proposes a method which combines IKEv2 and O/TWAMP as defined in RFC 4656 and RFC 5357, respectively. This specification aims, on the one hand, to ensure that O/TWAMP can be secured, while on the other hand, it extends the applicability of O/TWAMP to networks that have already deployed IPsec.

Table of Contents

1. Introduction

The active measurement protocols OWAMP [RFC4656] and TWAMP [RFC5357] can be used to measure network performance parameters, such as latency, bandwidth, and packet loss by sending probe packets and monitoring their experience in the network. In order to guarantee the accuracy of network measurement results, security aspects must be considered, otherwise, attacks may occur and authenticity may be violated. For example, if no protection is provided, an adversary in the middle may modify packet timestamps, thus altering the measurement results.

Cryptographic security mechanisms, such as IPsec, have been considered during the early stage of working towards the definition of the two protocols mentioned above. However, due to several reasons, it was preferred to avoid tying the development and deployment of O/TWAMP protocols to such security mechanisms. In practice, for many networks, the issues listed in [RFC4656], Sec. 6.6 with respect to IPsec are still valid. However, we expect that in the near future IPsec will be deployed in many more hosts and networks than today. For example, IPsec tunnels may be used to secure wireless channels. In this case, what we are interested in is measuring network performance specifically for the traffic carried by the tunnel, not in general over of the wireless channel. Therefore, in this document we attempt to make the case that for networks where wide deployment of IPsec and other security mechanisms is mandatory for a variety of reasons, there are increasingly more use cases in which IPsec and O/TWAMP protocols are needed simultaneously. In other words, we argue that it is now time to specify how O/TWAMP can be used in a network environment where IPsec is already deployed. In such an environment, measuring IP performance over IPsec tunnels with O/TWAMP is an important tool for operators.

Another advantage of IPsec key exchange protocol may be that it is not necessary to use distinct keys in OWAMP-Control and OWAMP-Test layers. One key for encryption and another for authentication is sufficient for both the Control and Test layers. This obviates the need to generate two keys and could reduce the complexity of O/TWAMP protocols in this environment. This observation comes from the fact that separate session keys in Control and Test layers are designed for preventing reflection attacks when employing the current mechanism. Once IPsec is employed, such a potential threat is alleviated. Note that this will be very useful in the environments where IPsec capability has been supported.

2. Terminology used in this document

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. Motivation

Let us first consider why the reasons originally listed in [RFC4656] Sec. 6.6 may not apply today in many cases. First, the argument made is that partial authentication in O/TWAMP authentication mode is not possible with IPsec. IPsec indeed cannot authenticate only a part of a packet. However, in an environment where IPsec is already deployed and actively used, partial authentication of O/TWAMP contradicts the operational reasons dictating the use of IPsec. At the same time, this limits the applicability and use of O/TWAMP in networks using IPsec.

The second argument made is the need to keep separate deployment paths between O/TWAMP and IPsec. In several currently deployed types of networks, IPsec is widely used to protect the data and signaling planes. For example, in mobile telecommunication networks, the deployment rate of IPsec exceeds 95% with respect to the LTE serving network. In older technology cellular networks, such as UMTS and GSM, IPsec use penetration is lower, but still quite significant. Additionally, there is a great number of IPSec-based VPN applications which are widely used in business applications to provide end-to-end, or host-to-host security over IEEE 802.11 wireless LANs. At the same time, lots of standardized protocols make use of IPsec/IKE, including MIPv4/v6, HIP, SCTP, BGP, NAT and SIP, just to name a few.

Third, with respect to the support of IPsec in lightweight embedded devices, nowadays, a large number of limited-resource and low-cost devices, such as Ethernet switches, DSL modems, and other such devices come with support for IPsec "out of the box". Therefore concerns about implementation, although likely valid a decade ago, are not well founded today.

Fourth, everyday use of IPsec applications by field technicians, on the one hand, and good understanding of the IPsec API by many programmers, on the other, should not be anymore a reason for concern. On the contrary: By now, IPsec open source code is available for anyone who wants to use it. Therefore, although IPsec does need a certain level of expertise to deal with it, in practice, most competent technical personnel and programmers have no problems using it on a daily basis.

O/TWAMP actually consists of two inter-related protocols: O/TWAMP-Control and O/TWAMP-Test. O/TWAMP-Control is used to initiate, start, and stop test sessions and to fetch their results, whereas O/TWAMP-Test is used to exchange test packets between two measurement nodes. In the following subsections we consider security for each one separately and then make the case for using them over IPsec.

3.1. O/TWAMP-Control Security

O/TWAMP uses a simple cryptographic protocol which relies on AES-CBC for confidentiality and on HMAC-SHA1 truncated to 128 bits for message authentication. Three modes of operation are supported: unauthenticated, authenticated, and encrypted. The authenticated and encrypted modes require that endpoints possess a shared secret, typically a passphrase. The secret key is derived from the passphrase using a password-based key derivation function PBKDF2 (PKCS#5) [RFC2898].

In the unauthenticated mode, the security parameters are left unused. In the authenticated and encrypted modes, security parameters are negotiated during the control connection establishment. Before the client can send commands to a server, it has to establish a connection to the server. Then, the client opens a TCP connection to the server on the well-known port number 861. The server responds with a server greeting, which contains the Challenge, Mode, Salt and Count. If the client wants to establish the connection, it responds with a Set-Up-Response message, wherein the KeyID, Token and Client IV are included. The Token is the concatenation of a 16-octet challenge, a 16-octet AES Session-key used for encryption, and a 32-octet HMAC-SHA1 Session-key used for authentication. The token itself is encrypted using AES in Cipher Block Chaining (AES-CBC).

Encryption is performed using a key derived from the shared secret associated with KeyID. In the authenticated and encrypted modes, all further communications are encrypted using the AES Session-key and authenticated with the HMAC Session-key. The client encrypts everything it transmits through the just-established O/TWAMP-Control connection using stream encryption with Client-IV as the IV. Correspondingly, the server encrypts its side of the connection using Server-IV as the IV. The IVs themselves are transmitted in cleartext. Encryption starts with the block immediately following the block containing the IV.

The AES Session-key and HMAC Session-key are generated randomly by the client. The HMAC Session-key is communicated along with the AES Session-key during O/TWAMP-Control connection setup. The HMAC Session-key is derived independently of the AES Session-key.

3.2. O/TWAMP-Test Security

The O/TWAMP-Test protocol runs over UDP, using sender and receiver IP and port numbers negotiated during the Request-Session exchange. As with O/TWAMP-Control, O/TWAMP-Test has three modes: unauthenticated, authenticated, and encrypted. All O/TWAMP-Test sessions that are spawned by an O/TWAMP-Control session inherit its mode.

The O/TWAMP-Test packet format is the same in authenticated and encrypted modes. The encryption and authentication operations are, however, different. Similarly with the respective O/TWAMP-Control session, each O/TWAMP-Test session has two keys: an AES Session-key and an HMAC Session-key. However, there is a difference in how the keys are obtained. In the case of O/TWAMP-Control, the keys are generated by the client and communicated (as part of the Token) during connection setup through the Set-Up-Response message. In the case of O/TWAMP-Test, the keys are derived from the O/TWAMP-Control keys and the session identifier (SID), as inputs of the key derivation function (KDF). The O/TWAMP-Test AES Session-key is generated by using the O/TWAMP-Control AES Session-key, with the 16-octet session identifier (SID), for encrypting and decrypting the packets of the particular O/TWAMP-Test session. The O/TWAMP-Test HMAC Session-key is generated by using the O/TWAMP-Control HMAC Session-key, with the 16-octet session identifier (SID), for authenticating the packets of the particular O/TWAMP-Test session.

3.3. O/TWAMP Security Root

As discussed above, the AES Session-key and HMAC Session-key used in the O/TWAMP-Test protocol are derived from the AES Session-key and HMAC Session-key which are used in O/TWAMP-Control protocol. The AES Session-key and HMAC Session-key used in the O/TWAMP-Control protocol are generated randomly by the client, and encrypted with the shared secret associated with KeyID. Therefore, the security root is the shared secret key. Thus, key provision and management are complicated and need to be taken care of appropriately. Comparatively, a certificate-based approach in IKEv2/IPsec can automatically manage the security root and solve this problem.

3.4. Co-existence of O/TWAMP and IPsec

According to [RFC4656] "[t]he deployment paths of IPsec and OWAMP could be separate if OWAMP does not depend on IPsec." The problem may occur in practice is that the security mechanism of O/TWAMP and IPsec cannot co-exist at the same time. IPsec provides confidentiality and data integrity to IP datagrams. Distinct protocols are provided: Authentication Header (AH), Encapsulating Security Payload (ESP) and Internet Key Exchange (IKE v1/v2). Only

integrity protection can be provided with AH. Both integrity and encryption can be provided with ESP. The IKE Protocol is used for dynamical key negotiation and automatic key management.

When the sender and receiver implement O/TWAMP over IPsec, they need to agree on a shared key during the establishment of the IPsec tunnel; subsequently all IP packets sent by the sender are protected. If the AH protocol is used, IP packets are transmitted in plaintext. The authentication part covers the entire packet. So all test information, such as UDP port number, and the test results will be visible to any attacker, which can intercept these test packets, and introduce errors or forge packets that may be injected during the transmission. In order to avoid this attack, the receiver must validate the integrity of these packets with the negotiated secret key. If ESP is used, IP packets are encrypted, and hence no other than the receiver can use the IPsec secret key and decrypt the IP packet, and then it can obtain the test data to assess the IP network performance based on the measurements. So both the sender and receiver must support IPsec to generate the security secret key of IPsec.

In the current implementation of O/TWAMP, after the test packets are received by the receiver, it cannot execute active measurement over IPsec. That is because the receiver knows only the shared secret key but not the IPsec key, while the test packets are protected by the IPsec key ultimately. Therefore, it needs to be considered how to measure IP network performance in an IPsec tunnel with O/TWAMP. Without this functionality, the use of OWAMP and TWAMP over IPsec is hindered.

Of course, backward compatibility should be considered, as well. That is, the intrinsic security method based on shared key as specified in the O/TWAMP standards can also fit the other platforms. There should be no impact on the current security mechanisms defined in O/TWAMP for other use cases. This document describes a possible solution to this problem which takes advantage of the secret key derived by IPsec, to provision the key needed in RFC 4656 and RFC 5357.

4. O/TWAMP over IPsec

A security method based on a shared secret key has been defined in O/TWAMP [RFC4656][RFC5357]. In this section, in order to employ O/TWAMP over IPsec, a method of binding O/TWAMP and IKEv2 is described, for those both the sender and receiver supporting the IPsec protocols. The shared key used in the security of O/TWAMP is derived from IPsec [RFC5996]. If the AH protocol is used, the IP

packets are transmitted in plaintext. All of O/TWAMP is integrity-protected by IPsec. Even if the peers choose to opt for the unauthenticated mode, IPsec integrity protection is extended to O/TWAMP. In the authenticated and encrypted modes, the shared secret can be derived from IKE SA or IPsec SA. If the shared secret key is derived from IKE SA, SKEYSEED must be generated firstly. SKEYSEED and its derivatives are computed as per [RFC5996], where prf is a pseudorandom function:

SKEYSEED = prf(Ni | Nr, g^ir)

Ni and Nr are nonces, negotiated during initial exchange. g^ir is the shared secret from the ephemeral Diffie-Hellman exchange and is represented as a string of octets. SKEYSEED can be used as the shared secret key directly, then the shared key is equal to SKEYSEED. Alternatively, the shared secret key can be generated as follows:

Shared secret key=PRF{ SKEYSEED, Session ID}

wherein the session ID is the SID agreed during the O/TWAMP-Test protocol.

If the shared secret key is derived from IPsec SA, the shared secret key can be equal to KEYMAT, wherein

KEYMAT = prf+(SK_d, Ni | Nr)

The term "prf+" describes a function that outputs a pseudorandom stream based on the inputs to a prf [RFC5996]; or the shared secret key can be generated as follows:

Shared secret key=PRF{ KEYMAT, Session ID}

wherein the session ID is the SID agreed during the O/TWAMP-Test protocol.

There are some cases for rekeying IKE SA and IPsec SA, after which the corresponding key of SA is updated. Generally ESP and AH SAs always exist in pairs, with one SA in each direction. If the SA is deleted, the key generated from the IKE SA or IPsec SA should also be updated.

As discussed above, a binding association between the key generated from IPsec and the shared secret key needs to be considered. SA can be identified by SPI and protocol uniquely for a given sender and a receiver. So these parameters should be agreed upon during the O/TWAMP protocol. When the sender and receiver execute O/TWAMP protocol to negotiate integrity key, the IPsec protocol and SPI

should be checked. Only if two parameters are matched with the information of IPsec, should the O/TWAMP connection be established. As illustrated in Fig. 1, the SPI and protocol type are included in the server greeting of the O/TWAMP-Control protocol. After the client receives the greeting, it closes the connection if it receives a greeting with an erroneous SPI and protocol value. Otherwise, the client responds with the following Set-Up-Response message and generates the shared secret key. This message exchange flow is illustrated as Fig. 1.

```
+--------+                  +--------+
| Client |                  | Server |
+--------+                  +--------+
    |                            |
    |     TCP  Connection        |
    |<-------------------------->|
    |                            |
    |    Greeting message        |
    |<---------------------------|
    |                            |
    |    Set-Up-Response         |
    |--------------------------->|
    |                            |
    |                            |
    |      Server-Start          |
    |<---------------------------|
    |                            |
```

Figure 1. The procedure of O/TWAMP-Control

The format of server greeting is illustrated in Fig. 2. The unused 12 octets are used to carry the new parameter: protocol and SPIs.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            protocol                           |
|+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-++-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            SPIi                               |
|+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-++-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            SPIr                               |
|+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-++-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             Modes                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                      Challenge (16 octets)                    |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                         Salt (16 octets)                      |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Count (4 octets)                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                         MBZ (12 octets)                       |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2. The format of server greeting

In ESP, when the IP packets are encrypted, no other than the receiver can use the IPsec key and decrypt the IP packets. It gains the test data to process measurement IP performance. In this case, the IPsec tunnel between the sender and receiver provides additional security. Even if the peers choose the unauthenticated mode, IPsec encryption and integrity protection is provided to O/TWAMP. If the sender and receiver also want to use authenticated or encrypted mode, the shared secret can be also derived from IKE SA or IPsec SA. The method of key generation and binding association is the same as AH protocol mode.

Besides, there is encryption-only configuration in ESP, though not recommended due to its limitations. Since it does not produce integrity key in this case, either encryption-only ESP should be prohibited for O/TWAMP, or a decryption failure should be distinguished due to possible integrity attack.

5. Others

The community may want to revisit the arguments listed in [RFC4656], Sec. 6.6. Other widely-used Internet security mechanisms, such as TLS and DTLS, may also be considered for future use over and above of what is already specified in O/TWAMP.

6. Security Considerations

As the shared secret key is derived from IPsec, the key derivation algorithm strength and limitations are as per [RFC5996]. The strength of a key derived from a Diffie-Hellman exchange using any of the groups defined here depends on the inherent strength of the group, the size of the exponent used, and the entropy provided by the random number generator employed. The strength of all keys and implementation vulnerabilities, particularly DoS attacks are as defined in [RFC5996].

7. IANA Considerations

There may be IANA considerations for allocating additional value for these options. The values of the protocol field needed to be assigned from the numbering space.

8. Acknowledgments

We would like to thank Eric Chen and Yakov Stein for their comments, and Al Morton for pointing to previous work discussed in IPPM WG.

9. References

9.1. Normative References

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

[RFC4656] Shalunov, S., Teitelbaum, B., Karp, A., Boote, J., and M. Zekauskas, "A One-way Active Measurement Protocol (OWAMP)", RFC 4656, September 2006.

[RFC5357] Hedayat, K., Krzanowski, R., Morton, A., Yum, K., and J. Babiarz, "A Two-Way Active Measurement Protocol (TWAMP)", RFC 5357, October 2008.

[RFC5996] Kaufman, C., Hoffman, P., Nir, Y., and P. Eronen, "Internet Key Exchange Protocol Version 2 (IKEv2)", RFC 5996, September 2010.

9.2. Informative References

[RFC2898] Kaliski, B., "PKCS #5: Password-Based Cryptography Specification Version 2.0", RFC 2898, September 2000.

Authors' Addresses

Yang Cui
Huawei Technologies
Huawei Building, Q20, No.156, Rd. BeiQing
Haidian District, Beijing 100095
P. R. China

Email: cuiyang@huawei.com

Emily Bi
Huawei Technologies
Huawei Building, Q20, No.156, Rd. BeiQing
Haidian District, Beijing 100095
P. R. China

Phone: +86-10-82881907
Email: bixiaoyu@huawei.com

Kostas Pentikousis (editor)
Huawei Technologies
Carnotstrasse 4
10587 Berlin
Germany

Email: k.pentikousis@huawei.com