Network Working Group
Internet Draft
Intended status: Standards Track
Expires: October 30, 2009

K. Grewal
Intel Corporation
G. Montenegro
Microsoft Corporation
M. Bhatia
Alcatel-Lucent

April 30, 2009

Wrapped ESP for Traffic Visibility
draft-ietf-ipsecme-traffic-visibility-02.txt

Status of this Memo

Copyright

Abstract

This document describes the Wrapped Encapsulating Security Payload (WESP) protocol, which builds on top of ESP [RFC4303] and is designed to allow intermediate devices to ascertain if ESP-NULL is being employed and hence inspect the IPsec packets for network monitoring and access control functions. Currently in the IPsec standard, there is no way to differentiate between ESP encryption and ESP NULL encryption by simply examining a packet. This poses certain challenges to the intermediate devices that need to deep inspect the packet before making a decision on what should be done with that packet (Inspect and/or Allow/Drop). The mechanism described in this document can be used to easily disambiguate ESP-NULL from ESP encrypted packets, without compromising on the security provided by ESP.

Table of Contents

1. Introduction

Use of ESP within IPsec [RFC4303] specifies how ESP packet encapsulation is performed. It also specifies that ESP can use NULL encryption [RFC2410] while preserving data integrity and authenticity. The exact encapsulation and algorithms employed are negotiated out-of-band using, for example, IKEv2 [RFC4306] and based on policy.

Enterprise environments typically employ numerous security policies (and tools for enforcing them), as related to access control, content screening, firewalls, network monitoring functions, deep packet inspection, Intrusion Detection and Prevention Systems (IDS and IPS), scanning and detection of viruses and worms, etc. In order to enforce these policies,

network tools and intermediate devices require visibility into

packets, ranging from simple packet header inspection to deeper payload examination. Network security protocols which encrypt the data in transit prevent these network tools from performing the aforementioned functions.

When employing IPsec within an enterprise environment, it is desirable to employ ESP instead of AH [RFC4302], as AH does not work in NAT environments. Furthermore, in order to preserve the above network monitoring functions, it is desirable to use ESP-NULL. In a mixed mode environment some packets containing sensitive data employ a given encryption cipher suite, while other packets employ ESP-NULL. For an intermediate device to unambiguously distinguish which packets are leveraging ESP-NULL, they would require knowledge of all the policies being employed for each protected session. This is clearly not practical. Heuristic-based methods can be employed to parse the packets, but these can be very expensive, containing numerous rules based on each different protocol and payload. Even then, the parsing may not be robust in cases where fields within a given encrypted packet happen to resemble the fields for a given protocol or heuristic rule. This is even more problematic when different length Initialization Vectors (IVs), Integrity Check Values (ICVs) and padding are used for different security associations, making it difficult to determine the start and end of the payload data, let alone attempting any further parsing. Furthermore, storage, lookup and cross-checking a set of comprehensive rules against every packet adds cost to hardware implementations and degrades performance. In cases where the packets may be encrypted, it is also wasteful to check against heuristics-based rules, when a simple exception policy (e.g., allow, drop or redirect) can be employed to handle the encrypted packets. Because of the non-deterministic nature of heuristics-based rules for disambiguating between encrypted and non-encrypted data, an alternative method for enabling intermediate devices to function in encrypted data environments needs to be defined. Additionally there are many types and classes of network devices employed within a given network and a deterministic approach would provide a simple solution for all these devices. Enterprise environments typically use both stateful and stateless packet inspection mechanisms. The previous considerations weigh particularly heavy on stateless mechanisms such as router ACLs and NetFlow exporters. Nevertheless, a deterministic approach provides a simple solution for the myriad types of devices employed within a network, regardless of their stateful or stateless nature.

This document defines a mechanism to provide additional information in relevant IPsec packets so intermediate devices

can efficiently differentiate between encrypted ESP packets and ESP packets with NULL encryption.

The document is consistent with the operation of ESP in NAT environments [RFC3947].

The design principles for this protocol are the following:

o Allow easy identification and parsing of integrity-only IPsec traffic

o Leverage the existing hardware IPsec parsing engines as much as possible to minimize additional hardware design costs

o Minimize the packet overhead in the common case

1.1. 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 RFC 2119 [RFC2119].

1.2. Applicability Statement

The document is applicable only to the wrapped ESP header defined below, and does not describe any changes to either ESP [RFC4303] nor AH [RFC4302].

2. Wrapped ESP (WESP) Header format

The proposal is to define a protocol number for Wrapped ESP encapsulation (WESP), which provides additional attributes in each packet to assist in differentiating between encrypted and non-encrypted data, as well as aid parsing of the packet. WESP follows RFC 4303 for all IPv6 and IPv4 considerations (e.g., alignment considerations).

This extension essentially acts as a wrapper to the existing ESP protocol and provides an additional 4 octets at the front of the existing ESP packet.

This may be depicted as follows:

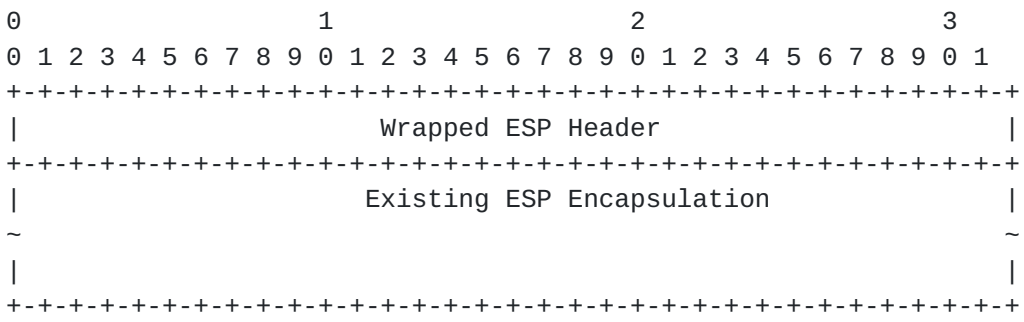


Figure 1 WESP Packet Format

By preserving the body of the existing ESP packet format, a compliant implementation can simply add in the new header, without needing to change the body of the packet. The value of the new protocol used to identify this new header is TBD via IANA. Further details are shown below:

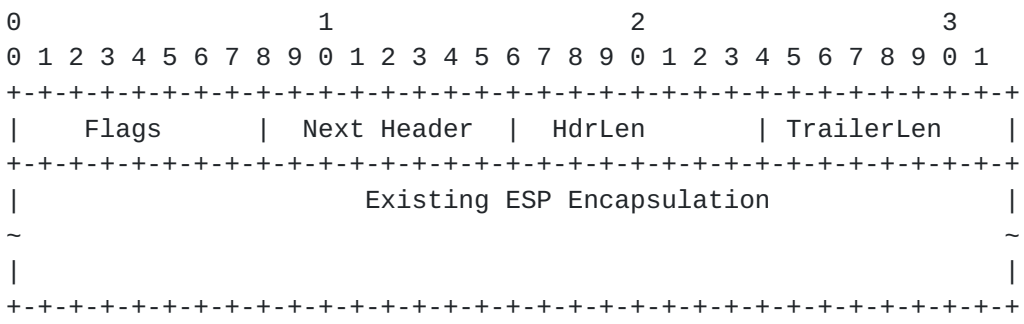


Figure 2 Detailed WESP Packet Format

Where:

Flags, 8 bits

2 bits: Version. Version is set to 0 by the transmitter and validated by the receiver. Any modifications to the WESP header in the future will require an update in the version number.

6 bits: reserved for future use. These MUST be set to zero per this specification, but usage may be defined by other specifications.

Note: To provide future compatibility, the version number is negotiated by the control channel handshake. An implementation compatible with this specification must set the version number and the reserved bits to the values specified above when

transmitting a packet. On receiving a packet, these values must be checked to ensure that they are as indicated above.

Next Header, 8 bits: If using ESP-NULL, this field MUST be equal to the Next Header field in the ESP trailer. If using ESP in encryption mode, this field MUST be set to zero..

HdrLen, 8 bits: Offset to the beginning of the Payload Data in octets.

TrailerLen, 8 bits: Offset from the end of the packet to the last byte of the payload data in octets.

As can be seen, this wrapped ESP format extends the standard ESP header by the first 4 octets. The WESP header is integrity protected, along with all the fields specified for ESP in RFC 4303.

2.1. UDP Encapsulation

This section describes a mechanism for running the new packet format over the existing UDP encapsulation of ESP as defined in RFC 3948. This allows leveraging the existing IKE negotiation of the UDP port for NAT-T discovery and usage [RFC3947], as well as preserving the existing UDP ports for ESP (port 4500). With UDP encapsulation, the packet format can be depicted as follows.

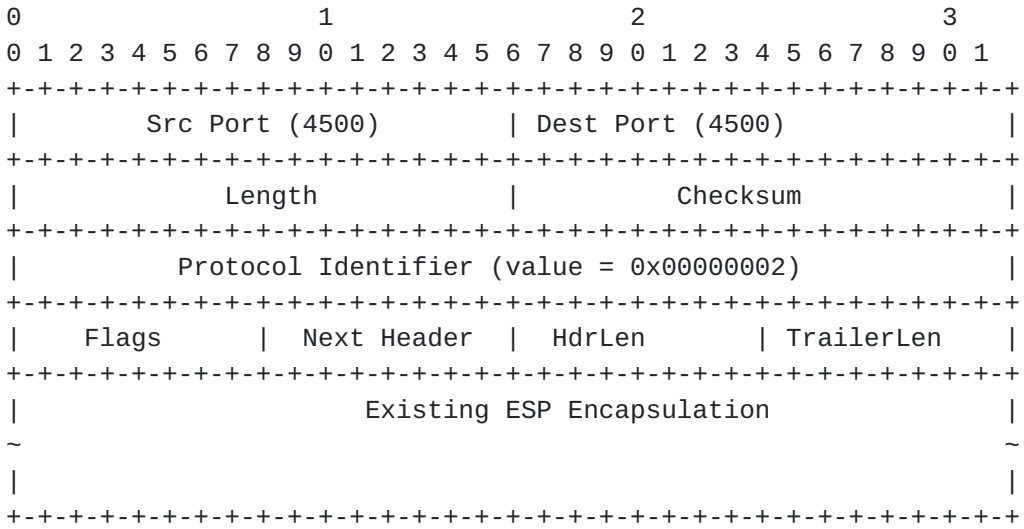


Figure 3 UDP-Encapsulated WESP Header

Where:

Source/Destination port (4500) and checksum: describes the UDP encapsulation header, per RFC3948.

Protocol Identifier: new field to demultiplex between UDP encapsulation of IKE, UDP encapsulation of ESP per RFC 3948, and the UDP encapsulation in this specification.

According to RFC 3948, clause 2.2, a 4 octet value of zero (0) immediately following the UDP header indicates a Non-ESP marker, which can be used to assume that the data following that value is an IKE packet. Similarly, a value greater then 255 indicates that the packet is an ESP packet and the 4-octet value can be treated as the ESP SPI. However, RFC 4303, clause 2.1 indicates that the values 1-255 are reserved and cannot be used as the SPI. We leverage that knowledge and use one of these reserved values to indicate that the UDP encapsulated ESP header contains this new packet format for ESP encapsulation.

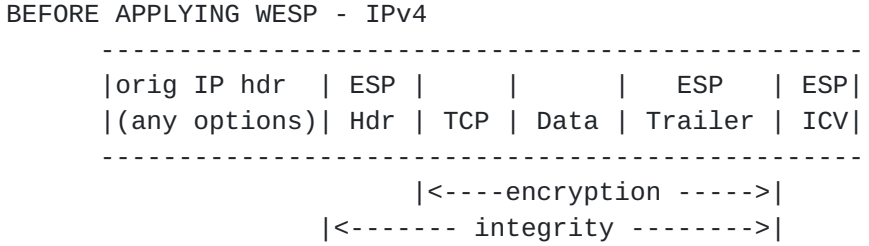
The remaining fields in the packet have the same meaning as per section 2 above.

2.2. Transport and Tunnel Mode Considerations

This extension is equally applicable for transport and tunnel mode where the ESP Next Header field is used to differentiate between these modes, as per the existing IPsec specifications.

2.2.1. Transport Mode Processing

In transport mode, ESP is inserted after the IP header and before a next layer protocol, e.g., TCP, UDP, ICMP, etc. The following diagrams illustrate how WESP is applied to the ESP transport mode for a typical packet, on a "before and after" basis.



* = if present, could be before WESP, after ESP, or both

All other considerations are as per RFC 4303.

2.2.2. Tunnel Mode Processing

In tunnel mode, ESP is inserted after the new IP header and before the original IP header, as per RFC 4303. The following diagram illustrates how WESP is applied to the ESP tunnel mode for a typical packet, on a "before and after" basis.

* = if present, construction of outer IP hdr/extensions and

modification of inner IP hdr/extensions is discussed in

the Security Architecture document.

All other considerations are as per RFC 4303.

2.3. IKE Considerations

This document assumes that WESP negotiation is performed using IKEv2. In order to negotiate the new format of ESP encapsulation

via IKEv2 [RFC4306], both parties need to agree to use the new packet format. This can be achieved using a notification method similar to USE_TRANSPORT_MODE defined in RFC 4306.

The notification, USE_WESP_MODE (value TBD) MAY be included in a request message that also includes an SA payload requesting a CHILD_SA using ESP. It requests that the CHILD_SA use WESP mode rather than ESP for the SA created. If the request is accepted, the response MUST also include a notification of type USE_WESP_MODE. If the responder declines the request, the CHILD_SA will be established using ESP, as per RFC 4303. If this is unacceptable to the initiator, the initiator MUST delete the SA. Note: Except when using this option to negotiate WESP mode, all CHILD_SAs will use standard ESP.

Negotiation of WESP in this manner preserves all other negotiation parameters, including NAT-T [RFC3948]. NAT-T is wholly compatible with this wrapped frame format and can be used as-is, without any modifications, in environments where NAT is present and needs to be taken into account.

3. Security Considerations

As this document augments the existing ESP encapsulation format, UDP encapsulation definitions specified in RFC 3948 and IKE negotiation of the new encapsulation, the security observations made in those documents also apply here. In addition, as this document allows intermediate device visibility into IPsec ESP encapsulated frames for the purposes of network monitoring functions, care should be taken not to send sensitive data over connections using definitions from this document, based on network domain/administrative policy. A strong key agreement protocol, such as IKE, together with a strong policy engine should be used to in determining appropriate security policy for the given traffic streams and data over which it is being employed.

ESP is end-to-end and it will be impossible for the intermediate devices to verify that all the fields in the WESP header are correct. It is thus possible to tweak the WESP header so that the packet sneaks past the firewall if the fields in the WESP header are set to something that the firewall will allow. The endpoint thus must verify the sanity of the WESP header before accepting the packet. In an extreme case, someone colluding with the attacker, could change the WESP fields back to the original values so that the attack goes unnoticed. However, this is not a new problem and it already exists IPSec.

4. IANA Considerations

Reserving an appropriate value for this encapsulation as well as a new value for the protocol in the IKE negotiation is TBD by IANA.

5. Acknowledgments

The authors would like to acknowledge the following people for their feedback on updating the definitions in this document.

David McGrew, Brian Weis, Philippe Joubert, Brian Swander, Yaron Sheffer, Men Long, David Durham, Prashant Dewan, Marc Millier among others.

This document was prepared using 2-Word-v2.0.template.doc.

6. References

6.1. Normative References

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

[RFC2410] Glenn, R. and S. Kent, "The NULL Encryption Algorithm and Its Use With IPsec", RFC 2410, November 1998.

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

6.2. Informative References

[RFC3947] Kivinen, T., Swander, B., Huttunen, A., and V. Volpe, "Negotiation of NAT-Traversal in the IKE", RFC 3947, January 2005.

[RFC3948] Huttunen, A., Swander, B., Volpe, V., DiBurro, L., and M. Stenberg, "UDP Encapsulation of IPsec ESP Packets", RFC 3948, January 2005.

[RFC4302] Kent, S., "IP Authentication Header", RFC 4302, December 2005.

[RFC4306] Kaufman, C., "Internet Key Exchange (IKEv2) Protocol", RFC 4306, December 2005.

Author's Addresses

Ken Grewal
Intel Corporation
2111 NE 25th Avenue, JF3-232
Hillsboro, OR 97124
USA

Phone:
Email: ken.grewal@intel.com

Gabriel Montenegro
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052
USA

Phone:
Email: gabriel.montenegro@microsoft.com

Manav Bhatia
Alcatel-Lucent
Bangalore
India

Phone:
Email: manav@alcatel-lucent.com