

IPSECME
Internet-Draft
Intended status: Standards Track
Expires: August 21, 2015

D. Migault (Ed)
Ericsson
D. Palomares
Orange/LIP6
February 17, 2015

MOBIKEv2: MOBIKE extension for Transport mode
draft-mglt-ipsecme-mobikev2-01.txt

Abstract

MOBIKE, the IKEv2 Mobility and Multihoming Protocol is defined only for CHILD_SA using the tunnel mode. This document describes MOBIKEv2 that extends MOBIKE for CHILD_SA using also transport mode.

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 August 21, 2015.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Requirements notation	2
2. Terminology	2
3. Introduction	3
4. Problem Statement	4
5. IKE_AUTH Exchange with MOBIKEv2	5
6. Updating IP addresses with MOBIKEv2	6
7. IPsec Databases Impacts	7
7.1. Security Policy Database (SPD)	8
7.2. Security Association Database (SAD)	8
7.3. Peer Authentication Database (PAD)	8
8. Security Considerations	8
9. IANA Considerations	9
10. Normative References	9
Authors' Addresses	9

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. Terminology

This document uses the following terminology:

- Initiator: The Initiator is the peer that initiates an exchange. It starts by sending a message towards the Responder. Note that if two peers are connected, the Initiator of one exchange can be the Responder of another exchange.
- Responder: The Responder is the peer receiving an exchange. The message is sent from the Initiator.
- Security Policy (SP): is defined in section 4 of [RFC4301]. As mobility or multihoming concerns an already established session, SP mostly designate Security Policy in the SPD cache. The SP contains the processing information like the IPsec mode, the protocol to use as well as encryption and authorization algorithms. SP also contains a binding to the appropriated CHILD_SA. Binding between SP and CHILD_SA is described in section 4.4.2.2 of [RFC4301] and in annex 1 of [RFC4555]. In most cases the binding is performed using addresses of implementation specific structures.
- Security Policy Database (SPD): is defined is defined in section 4.4.1.2 of [RFC4301]. In this document we are mostly focused

on the SPD cache. The SPD contains all SP. SP match for outbound packet is performed through Traffic Selectors usually composed of the IP addresses and ports.

- Security Association (SA): is defined in section 4 of [RFC4301]. SA are stored in the Security Association Database. The SA carries the processing information (cryptographic keys, counters, tunnel IP addresses when the tunnel mode is used), as well as the SPD Traffic Selectors used to check the processed inbound packet matches the SP the SA is derived from. SA are also designated by CHILD_SA in this document
- Security Associations Database (SAD): is defined in section 4.4.1.2 of [RFC4301]. The SAD contains all CHILD_SA. The CHILD_SA is indexed by Selectors (Security Parameters Index (SPI) as well as the IP addresses of the inbound packet).
- Peer Authorization Database (PAD): is defined in section 4.4.3 of [RFC4301].
- MOBIKE: designates MOBIKE as described in [RFC4555].
- MOBIKEv2: designates the protocol described in this document, that is MOBIKE version 2.

3. Introduction

Currently, MOBIKE [RFC4555] provides mobility and multihoming capabilities only for CHILD_SA using the tunnel mode. On the other hand, a large set of VPN solutions rely on GRE/IP tunnels and IPsec protection of these tunnels uses the transport mode. Similarly, for example when traffic is offloaded from Radio Access Network to public WLAN, IPsec may also be used to secure application. Some applications, like the DNS, may prefer the transport mode instead of the tunnel mode. In any case, the use of the transport mode prevents these connection to benefit from mobility and multihoming otherwise provided by MOBIKE.

This document specifies MOBIKEv2 that extends MOBIKE [RFC4555] to the transport mode. By doing so, communication protected with IPsec transport mode can also benefit from multihoming and mobility capabilities.

The remaining of the document is as follows. Section 5 specifies how to negotiate the support of MOBIKEv2 with the creation of an CHILD_SA using the transport mode. Section 6 describes how updates and additional IP addresses are handled with the transport mode. Section 7 details how IP updates on CHILD_SA impact the databases.

We assume the reader is familiar with IPsec [RFC4301], IKEv2 [RFC7296] and with MOBIKE [RFC4555].

4. Problem Statement

[RFC4555] section 3.2 states that the use of MOBIKE is indicated with the MOBIKE_SUPPORTED Notify Payload in the IKE_AUTH exchange. [RFC7296] section 1.3.1 states that the use of the transport mode is indicated by the USE_TRANSPORT_MODE Notify Payload in a Create Child exchange.

MOBIKE [RFC4555] section 1.2 considers outside its scope the use of mobility and multihoming with a CHILD_SA using the transport mode. As result, an Initiator is not supposed to send both an MOBIKE_SUPPORTED and a USE_TRANSPORT_MODE Notify Payload in its IKE_AUTH, and this case is left undefined. In case the Initiator sends these two payloads, possible Responder's behaviors may be:

- 1) The Responder responds with MOBIKE_SUPPORTED and USE_TRANSPORT_MODE. In this case, MOBIKE may only be provided for the IKE_SA, the CHILD_SA is using the transport mode and no mobility or multihoming facilities are provided for the CHILD_SA.
- 2) The Responder responds with MOBIKE_SUPPORTED only, in which case, it indicates it supports MOBIKE and refuses the transport mode for the CHILD_SA. One reason for refusing the transport mode may be that MOBIKE has only been defined for the tunnel mode. Such situation may results from prioritizing extensions.
- 3) The Responder responds with USE_TRANSPORT_MODE only, in which case it indicates it supports the transport mode for the CHILD_SA, but not MOBIKE. This case may be similar to case 2 with a different prioritization.
- 4) The Responder may ignore both Notify Payloads as this case has not been specified.

As a result, the use MOBIKEv2 with CHILD_SA using the transport mode requires to clarify the combination of the MOBIKE_SUPPORTED and the USE_TRANSPORT_MODE Notify Payload in an IKE_AUTH exchange. This is the purpose of Section 5

MOBIKE updates the IP addresses using an UPDATE_SA_ADDRESSES Notify Payload. At the reception of the UPDATE_SA_ADDRESSES Notify Payload, the Responder identifies the concerned IKE_SA and associated CHILD_SA(s). The IP addresses of the Initiator is replaced in both the IKE_SA and the CHILD_SA(s) with the IP address of the IP header

used to carry UPDATE_SA_ADDRESSES Notify Payload. The IKE_SA is actually stored in the IKEv2 application, whereas CHILD_SAs are in the SAD.

When MOBIKE is activated, the CHILD_SAs are using the tunnel mode of IPsec. Thus, updating the IP address requires the tunnel to be updated within the CHILD_SA as well as the Selectors (SPI, IP addresses) of the CHILD_SA in the SAD. MOBIKEv2 supports CHILD_SA with transport mode. In this case, updating the IP address requires updating the SPD Traffic Selectors within the CHILD_SA as well as the Selectors of the SAD. In addition, the Traffic Selectors of the SPD cache also need to be updated. This is the major change of MOBIKEv2 versus MOBIKE. Section 6 specifies the protocol details of MOBIKEv2 and Section 7 clarifies the impact on the various IPsec databases.

5. IKE_AUTH Exchange with MOBIKEv2

With MOBIKEv2 support of mobility and multihoming for a CHILD_SA using the transport mode results from the combination of the USE_TRANSPORT_MODE and MOBIKE_SUPPORTED Notify Payload within the IKE_AUTH exchange in the message containing the SA Payload. Outside of this scope MOBIKE_SUPPORTED Notify Payload is not expected as defined in [RFC4555] section 3.2.

With MOBIKEv2 the Initiator may initiate an IKE_AUTH exchange with the following combinations of the USE_TRANSPORT_MODE and MOBIKE_SUPPORTED Notify Payload.

- 1) The presence of the USE_TRANSPORT_MODE and MOBIKE_SUPPORTED Notify Payload indicates a request for both a CHILD_SA with the transport mode and the support for MOBIKEv2 for this CHILD_SA. This support is only provided by MOBIKEv2 and is not specified by MOBIKE [RFC4555].
- 2) The presence of the USE_TRANSPORT_MODE and the absence of the MOBIKE_SUPPORTED Notify Payload indicates a request for a CHILD_SA with the transport mode and no support for MOBIKE. This case is specified in [RFC7296] and MOBIKEv2 leave this specification unchanged.
- 3) The absence of the USE_TRANSPORT_MODE and the presence of the MOBIKE_SUPPORTED Notify Payload indicates a request for a CHILD_SA with the tunnel mode and the support for MOBIKEv2. This case is specified in [RFC4555] and MOBIKEv2 leave the specification unchanged.
- 4) The absence of both the USE_TRANSPORT_MODE and the MOBIKE_SUPPORTED Notify Payload indicates a request for a

CHILD_SA with the tunnel mode and no support for MOBIKE. This case is specified in [RFC4555] and MOBIKEv2 leave the specification unchanged.

As specified by IKEv2 [RFC7296] in section 1.3.1 and in MOBIKE [RFC4555] section 3.2, the Responder can respond with a USE_TRANSPORT_MODE or MOBIKE_SUPPORTED Notify Payload only if such payload has been previously provided by the Initiator while initiating a CHILD_SA negotiation during the IKE_AUTH exchange. Given these restrictions, with MOBIKEv2 the Responder may respond with the following combination of the USE_TRANSPORT_MODE and MOBIKE_SUPPORTED Notify Payload.

- 1) The presence of the USE_TRANSPORT_MODE and MOBIKE_SUPPORTED Notify Payload indicates the CHILD_SA uses the transport mode and the support for MOBIKEv2 for this CHILD_SA. This support is only provided by MOBIKEv2 and is not specified by MOBIKE [RFC4555].
- 2) The presence of the USE_TRANSPORT_MODE and the absence of the MOBIKE_SUPPORTED Notify Payload indicates the CHILD_SA uses the transport mode and no support for MOBIKE is provided. This case is specified in [RFC7296] and MOBIKEv2 leave this specification unchanged.
- 3) The absence of the USE_TRANSPORT_MODE and the presence of the MOBIKE_SUPPORTED Notify Payload indicates the CHILD_SA uses the tunnel mode and support for MOBIKEv2 is provided. This case is specified in [RFC4555] and MOBIKEv2 leave the specification unchanged.
- 4) The absence of both the USE_TRANSPORT_MODE and the MOBIKE_SUPPORTED Notify Payload indicates the CHILD_SA uses the tunnel mode and no support for MOBIKE is provided. This case is specified in [RFC4555] and MOBIKEv2 leave the specification unchanged.

In case the response does not satisfy the Initiator, it MUST delete the CHILD_SA as specified in [RFC7296] section 1.3.1.

6. Updating IP addresses with MOBIKEv2

CHILD_SAs may be updated when a UPDATE_SA_ADDRESSES Notify Payload is received or when the other peer become unreachable, in which case, the newly assigned IP address has been provided by an ADDITIONAL_*_ADDRESS Notify Payload. This section details how CHILD_SA MUST be updated when the CHILD_SA uses the transport mode.

Updating the IP address of the CHILD_SA using the transport mode impacts the SPD cache. As a result, IP address MUST be checked against the SPD and the PAD before performing any update of the CHILD_SA, or before communicating the IP address as an alternate IP address. More specifically:

- 1) The Initiator MUST NOT send an UPDATE_SA_ADDRESSES if the newly acquired IP does not match the SPD and the PAD.
- 2) The Initiator MUST NOT send an IP address in an ADDITIONAL_*_ADDRESS if the IP address does not match the SPD and the PAD.
- 3) The Initiator and Responder MUST check an IP address match the SPD and the PAD before updating the CHILD_SA.
- 4) The Responder MUST send an UNACCEPTABLE_ADDRESS Notify Payload described in section 4.1.1 of [RFC4555] if the IP address does not match the SPD and the PAD.

Similarly to MOBIKE, the appropriated IP address is the newly acquired IP address considered by the Initiator (either when a mobility occurs or when an additional IP address is used). This IP address is provided by the Initiator to the Responder via the IP header of the UPDATE_SA_ADDRESSES Notify Payload.

Updating a CHILD_SA using the transport mode with a new IP address involves updating:

- 1) SPD Traffic Selectors in the CHILD_SA. Note that with the tunnel mode these selectors remain unchanged so this update is specific to the transport mode.
- 2) SA Selectors in the SAD. This update is similar as with MOBIKE except that the IP address is not the one of the tunnel.
- 3) Traffic Selectors of the SPD cache MUST also be updated with the appropriated IP address. Note that with the tunnel mode these selectors remain unchanged so this update is specific to the transport mode.

7. IPsec Databases Impacts

This section discusses the impact of MOBIKEv2 on the IPsec databases. Since implementations vary widely, we do not discuss how these updates MUST be performed.

7.1. Security Policy Database (SPD)

The SPD MUST NOT be modified. Only the SPD cache needs to be modified. MOBIKE did not necessarily require update on the SPD cache, mostly because the Traffic Selectors are left unchanged with the tunnel mode. In fact, SPD Cache also have the outer IP addresses in its processing information (cf. section 4.1.2 of [RFC4301]). This information MAY be also defined in conjunction of the PAD, and eventually MAY be derived from the IP header of the IKE_INIT. However, this information is mostly used to negotiate the corresponding CHILD_SA, and for this reason, does not necessarily require to be updated. On the other hand as discussed in Appendix A.1 of [RFC4555], if this information is used to link the SPD cache entry to the CHILD_SA, then this information MUST be updated properly.

With MOBIKEv2 for CHILD_SA using the transport mode, the SPD Traffic Selectors MUST be updated, and as such, the SPD MUST be updated. For this reason the IP address MUST match the SPD and PAD before performing the update.

7.2. Security Association Database (SAD)

MOBIKE requires to update the Selector of the CHILD_SA as well as the content of the CHILD_SA (the Tunnel outer IP addresses). With MOBIKEv2 for CHILD_SA using the transport mode, there is no tunnel outer IP addresses to update. Instead the SPD Selectors in the CHILD_SA as well as the Selector of the CHILD_SA MUST be updated.

7.3. Peer Authentication Database (PAD)

The PAD MUST NOT be updated.

8. Security Considerations

Security Considerations regarding mobility and multihoming have already been expressed in [RFC4555].

The use of the transport mode makes visible and unprotected the IP header of the carried IP packet. This. This discloses privacy related information as the IP header indicates the end points communicating. This could be avoided with the tunnel mode as the end point was the Security Gateway.

9. IANA Considerations

There is no IANA consideration. The signaling provided by MOBIKE is sufficient.

10. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.
- [RFC4555] Eronen, P., "IKEv2 Mobility and Multihoming Protocol (MOBIKE)", RFC 4555, June 2006.
- [RFC7296] Kaufman, C., Hoffman, P., Nir, Y., Eronen, P., and T. Kivinen, "Internet Key Exchange Protocol Version 2 (IKEv2)", STD 79, RFC 7296, October 2014.

Authors' Addresses

Daniel Migault
Ericsson
8400 boulevard Decarie
Montreal, QC H4P 2N2
Canada

Email: mglt.ietf@gmail.com

Daniel Palomares
Orange/LIP6
38 rue du General Leclerc
92794 Issy-les-Moulineaux Cedex 9
France

Phone: +33 1 45 29 51 16
Email: daniel.palomares@orange.com

IPSecME Working Group
Internet-Draft
Intended status: Standards Track
Expires: October 31, 2014

Y. Nir
Check Point
April 29, 2014

Protecting Internet Key Exchange (IKE) Implementations from Denial of
Service Attacks through Client Puzzles
draft-nir-ipsecme-puzzles-00

Abstract

This document describes an enhancement to the Stateless Cookie mechanism described in RFC 5996. Whereas the original mechanism prevents denial-of-service (DoS) attacks that use multiple spoofed source addresses, the mechanism here is effective against a distributed denial of service attack (DDoS), where the attackers use their own source address. This is accomplished by requiring proof of work by the Initiator before allocating resources at the Responder.

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 31, 2014.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must

include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
1.1. Conventions Used in This Document	3
2. Protocol Overview	3
3. Puzzle Notification Format	4
4. Operational Considerations	5
5. Security Considerations	6
6. IANA Considerations	6
7. Normative References	6
Author's Address	6

1. Introduction

The Initial Exchange described in section 1.2 of [RFC5996] involves the Initiator sending a single message. The Responder also sends a single message, but also allocates state for a structure called a half-open IKE SA (Security Association). This half-open SA is later authenticated in the Authentication Exchange, but if that exchange doesn't come, the half-open SA is kept for an unspecified amount of time.

This creates an easy attack vector against an Internet Key Exchange (IKE) Responder. Generating the Initial request is cheap, and sending multiple such requests can either cause the Responder to allocate too much resources and fail, or else if resource allocation is limited, legitimate Initiators would also be prevented from setting up IKE SAs.

An obvious defense is limiting the number of half-open SAs opened by a single peer. However, since all that is required is a single packet, an attacker can use multiple spoofed source IP addresses.

Section 2.6 of RFC 5996 offers a mechanism to mitigate this DoS attack: the stateless cookie. When the server is under load, the Responder responds to the Initial request with a calculated "stateless cookie" - a value that can be re-calculated based on values in the Initial request without storing Responder-side state. The Initiator is expected to repeat the Initial request, this time including the stateless cookie.

This mechanism is not effective against attackers that have multiple source IP addresses with return routability, such as bot-nets.

The mechanism described here adds a proof of work for the Initiator, by partially breaking a hash function. This sets an upper bound, determined by the attacker's CPU to the number of negotiations it can force the Responder to participate in.

1.1. Conventions 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].

2. Protocol Overview

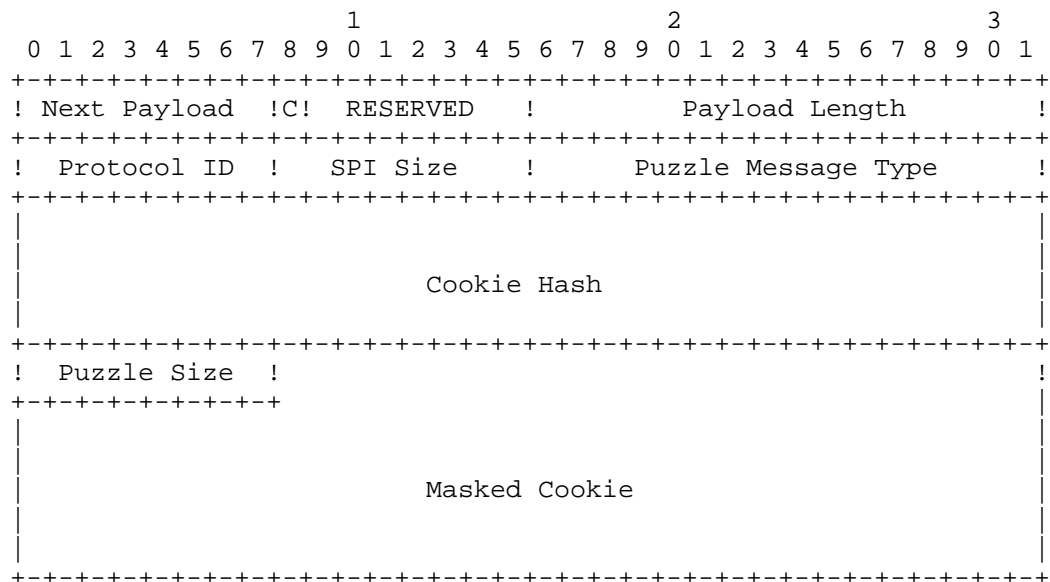
As described in section 2.6 of RFC 5996, when a responder detects a large number of half-open IKE SAs, it SHOULD reply to IKE_SA_INIT

requests with a response containing the COOKIE notification. When the number of half-open SAs gets even higher, so that there is a danger of degraded ability to reply to legitimate initiations, the responder SHOULD switch to sending puzzles instead of cookies. The puzzle is described in Section 3. The answer to the puzzle is the value to be returned in the Cookie notification.

The Initiator solves the puzzle, figures out what the stateless cookie is, and re-initiates as described in RFC 5996 with the Cookie notification carrying the answer to the puzzle.

3. Puzzle Notification Format

This section details the notification format for the puzzle. This notification is sent from Responder to the Initiator. See Section 4 for Operational Considerations in enabling this feature.



- o Protocol ID is set to zero.
- o SPI Size is set to zero.
- o Puzzle Message Type is set to xxxxx, the value assigned by IANA for the PUZZLE notification.
- o Cookie Hash (32 bytes) is the SHA2-256 hash of the stateless cookie.
- o Puzzle len (1 byte) is the number of unknown bits in the Masked Cookie field.

- o Masked Cookie (arbitrary length) is the stateless cookie that the Initiator is expected to return. The length is determined by the Payload Length field. The final n bits MUST be set to zero, where n is the number in the Puzzle Size field.

The Responder sets a difficulty level to a number of bits. See Section 4 for considerations in setting this difficulty level.

To construct this payload, the Responder first calculates the stateless cookie using the same procedure from RFC 5996. The responder then calculates the SHA2-256 hash of this stateless cookie to create the Cookie Hash field. The difficulty level is then placed in the Puzzle Size field. Finally, the last n bits (where n is the difficulty level) are zero'd out in the stateless cookie, and it is copied into the Masked Cookie field.

To solve the puzzle, the Initiator repeatedly attempts to complete the final n bits of the Masked Cookie, hashing each attempt until one is found where the hash matches the Cookie Hash field.

The Initiator then begins the Initial exchange again, this time including the Cookie Notification.

The entire exchange is below:

Initiator	Responder

HDR(A,0), SAi1, KEi, Ni -->	
	<-- HDR(A,0), N(PUZZLE)
HDR(A,0), N(COOKIE), SAi1, KEi, Ni -->	
	<-- HDR(A,B), SAR1, KEr, Nr, [CERTREQ]
HDR(A,B), SK {IDi, [CERT,] [CERTREQ,] [IDr,] AUTH, SAi2, TSi, TSr} -->	
	<-- HDR(A,B), SK {IDr, [CERT,] AUTH, SAR2, TSi, TSr}

4. Operational Considerations

[This section needs a lot of expanding]

Not all Initiators support this extension, but all initiators are supposed to support stateless cookies. If this notification is sent to a non-supporting but legitimate initiator, the exchange will fail. Responders are advised to first try to mitigate the DoS using

stateless cookies, and only if the number of half-open SAs keeps increasing, switch to using this mechanism.

The difficulty level should be set by balancing the requirement to minimize the latency for legitimate initiators and making things difficult for attackers. A good rule of thumb is for taking about 1 second to solve the puzzle. A typical initiator or bot-net member in 2014 can perform slightly less than a million hashes per second per core, so setting the difficulty level to $n=20$ is a good compromise. It should be noted that mobile initiators, especially phones are considerably weaker than that. Implementations should allow administrators to set the difficulty level, and/or be able to set the difficulty level dynamically in response to load.

Initiators should set a maximum difficulty level beyond which they won't try to solve the puzzle and log or display a failure message to the administrator or user.

5. Security Considerations

To be added.

6. IANA Considerations

IANA is requested to assign a notify message type from the status types range (16430-40959) of the "IKEv2 Notify Message Types - Status Types" registry with name "PUZZLE".

7. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC5996] Kaufman, C., Hoffman, P., Nir, Y., and P. Eronen, "Internet Key Exchange Protocol Version 2 (IKEv2)", RFC 5996, September 2010.

Author's Address

Yoav Nir
Check Point Software Technologies Ltd.
5 Hasolelim st.
Tel Aviv 6789735
Israel

Email: ynir.ietf@gmail.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: March 6, 2015

V. Smyslov
ELVIS-PLUS
September 2, 2014

The NULL Authentication Method in IKEv2 Protocol
draft-smyslov-ipsecme-ikev2-null-auth-03

Abstract

This document introduces the NULL Authentication Method for the IKEv2 Protocol. This method provides a way to omit peer authentication in the IKEv2. It may be used to preserve anonymity of or in the situations, where no trust relationship exists between the parties.

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 March 6, 2015.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
1.1. Conventions Used in This Document	3
2. Using the NULL Authentication Method	4
2.1. Authentication Payload	4
2.2. Identity Payload	4
3. Security Considerations	5
4. Acknowledgments	6
5. IANA Considerations	7
6. Normative References	8
Author's Address	9

1. Introduction

The Internet Key Exchange Protocol version 2 (IKEv2), specified in [IKEv2], provides a way for two parties to perform authenticated key exchange. Mutual authentication is mandatory in the IKEv2, so that each party must be authenticated by the other. However the authentication methods, used by the peers, need not be the same.

In some situations mutual authentication is undesirable, superfluous or impossible. For example:

- o User wants to get anonymous access to some server. In this situation he/she should be able to authenticate the server, but to leave out his/her own authentication to preserve anonymity. In this case one-way authentication of the responder is desirable.
- o Sensor, that sleeps most of the time, but periodically wakes up, makes some measurement (e.g. temperature) and sends the results to some server. The sensor must be authenticated by the server to ensure authenticity of the measurement, but the server need not be authenticated by the sensor. In this case one-way authentication of the initiator is sufficient.
- o Two peers without any trust relationship want to get some level of security in their communications. Without trust relationship they cannot prevent active Man-in-the-Middle attacks, but it is still possible to prevent passive eavesdropping with opportunistic encryption. In this case they can use unauthenticated key exchange.

To meet these needs the document introduces the NULL Authentication Method, which is a "dummy" method, that provides no authentication. This allows peer to explicitly indicate to the other side that it is unwilling or unable to certify its identity.

1.1. Conventions 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].

2. Using the NULL Authentication Method

In IKEv2 each peer independently selects the method to authenticate itself to the other side. It means that any of the peers may choose to omit its authentication by using the NULL Authentication Method. If it is not acceptable for the other peer, it MUST return AUTHENTICATION_FAILED Notification. Note, that when the Initiator uses EAP, the Responder MUST NOT use the NULL Authentication Method (in conformance with the section 2.16 of [IKEv2]).

The NULL Authentication Method affects how the Authentication and the Identity payloads are formed in the IKE_AUTH Exchange.

2.1. Authentication Payload

Despite the fact that the NULL Authentication Method provides no authentication, the AUTH Payload must still be present in the IKE_AUTH Exchange messages and must be properly formed, as it cryptographically links the IKE_SA_INIT Exchange messages with the other messages sent over the IKE SA.

With the NULL Authentication Method the content of the AUTH Payload MUST be computed using the syntax for pre-shared secret authentication, described in Section 2.15 of [IKEv2]. The values SK_pi and SK_pr MUST be used as shared secrets for the content of the AUTH Payloads generated by Initiator and Responder respectively. Note, that this is exactly how the content of the two last AUTH Payloads is calculated for non-key generating EAP Method (see Section 2.16 of [IKEv2] for details). The value for the the NULL Authentication Method is <TBA by IANA>.

2.2. Identity Payload

The NULL Authentication Method provides no authentication of the party using it. For that reason the Identity Payload content cannot be verified by the peer and MUST be ignored by the IKE.

This specification defines new ID Type - ID_NULL, which is intended to be used with the NULL Authentication Method to explicitly indicate anonymity of the peer. This ID Type SHOULD NOT be used with other authentication methods. The Identification Data in Identity Payload for the ID_NULL type MUST be absent and the ID Type is set to <TBA by IANA>.

3. Security Considerations

IKEv2 protocol provides mutual authentication of the peers. If one peer uses the NULL Authentication Method, then this peer cannot be authenticated by the other side, and it makes authentication in IKEv2 to be one-way. If both peers use the NULL Authentication method, key exchange becomes unauthenticated, that makes it subject to the Man-in-the-Middle attack.

The identity of the peer using the NULL Authenticated Method cannot be verified by the other side and, therefore, MUST NOT be used neither for authorization purposes, nor for policy decisions. All peers who use the NULL Authenticated Method should be considered by the other party as "guests" and get the least possible privileges.

If endpoint receives a request to create an unauthenticated IKE SA from the IP address, which is configured on the endpoint to be authenticated, the request SHOULD be rejected.

If the peer uses the NULL Authenticated Method, then the content of its Traffic Selector Payloads must be treated with care. In particular, implementations are advised not to trust blindly that the public IP addresses the peer put into TS Payload are really belong to it. It is RECOMMENDED for security gateways to always assign internal IP addresses to unauthenticated clients as described in Section 2.19 of [IKEv2].

4. Acknowledgments

The author would like to thank Paul Wouters, Yaron Sheffer and Tero Kivinen for their reviews and valuable comments.

5. IANA Considerations

This document defines new value in the "IKEv2 Authentication Method" registry:

<TBA> NULL Authentication Method

It also defines new value in the "IKEv2 Identification Payload ID Types" registry:

<TBA> ID_NULL

6. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [IKEv2] Kaufman, C., Hoffman, P., Nir, Y., Eronen, P., and T. Kivinen, "Internet Key Exchange Protocol Version 2 (IKEv2)", draft-kivinen-ipsecme-ikev2-rfc5996bis-04 (work in progress), June 2014.

Author's Address

Valery Smyslov
ELVIS-PLUS
PO Box 81
Moscow (Zelenograd) 124460
Russian Federation

Phone: +7 495 276 0211
Email: svan@elvis.ru

