Internet Engineering Task Force (IETF)
Request for Comments: 6467
Category: Informational
ISSN: 2070-1721

T. Kivinen
AuthenTec

December 2011

Secure Password Framework for Internet Key Exchange Version 2 (IKEv2)

Abstract

This document defines a generic way for Internet Key Exchange version 2 (IKEv2) to use any of the symmetric secure password authentication methods. Multiple methods are already specified in other documents, and this document does not add any new one. This document specifies a way to agree on which method is to be used in the current connection. This document also provides a common way to transmit, between peers, payloads that are specific to secure password authentication methods.

Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Not all documents approved by the IESG are a candidate for any level of Internet Standard; see Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc6467.

Table of Contents

1. Introduction

The IPsecME working group was chartered to provide for IKEv2 ([RFC5996]) a symmetric secure password authentication protocol that supports the use of low-entropy shared secrets, and to protect against off-line dictionary attacks without requiring the use of certificates or the Extensible Authentication Protocol (EAP). There are multiple such methods, and the working group was to pick one. Unfortunately, the working group failed to pick one protocol, and there are multiple candidates going forward as separate documents. As each of those older versions of those documents used a different technique to negotiate the use of the method and also used different payload formats, it is very hard to try to make an implementation where multiple such systems could co-exist.

Current document versions ([SPSK-AUTH], [PACE], and [PAKE]) use the method described in this document.

This document describes IKEv2 payload formats that can be used for multiple secure password methods to negotiate and transmit data so each different method can easily co-exist in the same implementation.

This document consists of two major parts:

- How to negotiate which secure password method negotiation is used.
- How to transmit data, between peers, that is specific to secure password methods.

The secure password methods are not usually meant to be used in the normal end user (remote access VPN) cases. In such cases, EAP-based authentication works fine, and the asymmetric nature of EAP does not matter. In such scenarios, the authentication is usually backed up with the back-end Authentication, Authorization, and Accounting (AAA) servers and other infrastructure. That is, in such scenarios, neither of the IKEv2 peers really knows the secret, as on one end it is typed in by the user when it is needed, and on the other end it is authenticated by the back-end AAA server.

The new secure password methods are meant to be used, for example, in the authentication between two servers or routers. These scenarios are usually symmetric: both peers know the shared secret, no back-end authentication servers are involved, and either end can initiate an IKEv2 connection. Note that such a model could also be supported by EAP when an EAP method that can run in symmetric fashion is in use, and the EAP method is directly implemented on both peers and no AAA is in use.

In many cases, each implementation will use only one of the proposed secure password authentication methods but can include support for multiple methods even when only one of them will be used. For example, a general-purpose operating system running IPsec and IKEv2 and supporting secure password authentication methods to protect services provided by the system might need to implement support for several methods. It is then up to the administrator which one is to be used. As the server might need to connect to multiple other servers, each implementing a different set of methods, it may not be possible to pick one method that would serve all cases.

The secure password methods mostly keep the existing IKEv2 IKE_SA_INIT exchange and modify the IKE_AUTH authentication step. As those methods do not want to add new round trips, negotiating which of the secure password methods to use needs to happen during the IKE_SA_INIT. As the identity of the other end is only provided inside IKE_AUTH, the responder needs to select the list of supported methods based only on the IP address of the initiator. This could lead to problems if only certain methods would be acceptable for certain identified peers. Fortunately, as the authentication is done based on the secret shared between both peers, the shared secret should be usable in all of the methods; thus, a remote peer usually

does not need to restrict selection of the method based on the initiator's identity but only based on the supported methods and the administrative policy.

Also, as the initiator already knows which peer it is connecting with, it can limit which methods it proposes to the other peer. And as secure password methods are meant to be used in symmetric cases, both ends should have similar configuration; i.e., they have the same shared secret and, most likely, also a list of acceptable authentication methods to be used. This could also be interpreted so that there is no need to support method negotiation, as both ends can already see this from the configuration. On the other hand, in most cases, either end does not really care which method is used but is willing to use any secure method that the other end supports. In such cases, the automatic negotiation provides a way to make the configuration easy, i.e., no need to pick one method to be used between the peers.

The reason for using the common IKEv2 payload to transmit, between peers, data that is specific to the secure password method is that the payload type field in the IKEv2 is only an 8-bit field, and 62.5% of the range is already reserved (50% to the private use numbers, and 12.5% to the IKEv1 payload numbers). This leaves 95 usable numbers, out of which 16 are already in use. Initially, it was proposed that five payload type numbers be consumed. Those five new payload types would already represent a 31% increase in the number of currently allocated payload types.

2. Method Negotiation

Because all of the methods modify the IKE_AUTH exchange, the negotiation of the secure password method to be used needs to happen during the IKE_SA_INIT exchange. The secure password negotiation exchange would be:

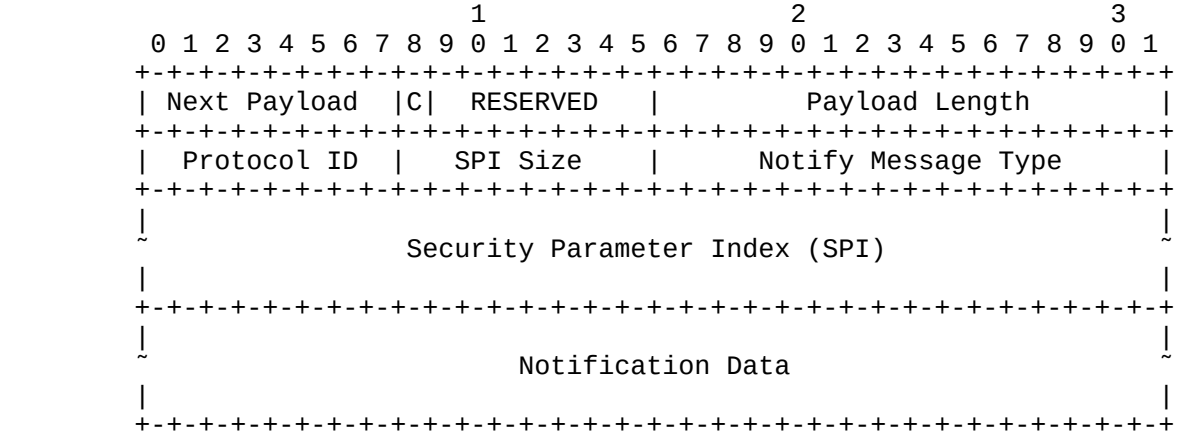
```
Initiator                          Responder
-------------------------------------------------------------------
HDR(SPIi=xxx, SPIr=0, IKE_SA_INIT,
    Flags: Initiator, Message ID=0),
    SAi1, KEi, Ni, [N(SECURE_PASSWORD_METHODS)]  -->

                    <--  HDR(SPIi=xxx, SPIr=yyy, IKE_SA_INIT,
                             Flags: Response, Message ID=0),
                             SAr1, KEr, Nr, [CERTREQ],
                             [N(SECURE_PASSWORD_METHODS)]
```

If the N(SECURE_PASSWORD_METHODS) Notify payload is missing, then normal IKEv2 authentication methods are used. If the Notify payloads are included, then the negotiation of the secure password methods happens inside those payloads.

As it might be possible that future secure password methods will modify the IKE_AUTH payload in a more substantial way, it is better that as an end result of the negotiation we have exactly one secure password method that will be used. The initiator will know which methods are usable when talking to that responder, so the initiator will send a list of acceptable methods in its IKE_SA_INIT request. The responder will pick exactly one method and put that in its response.

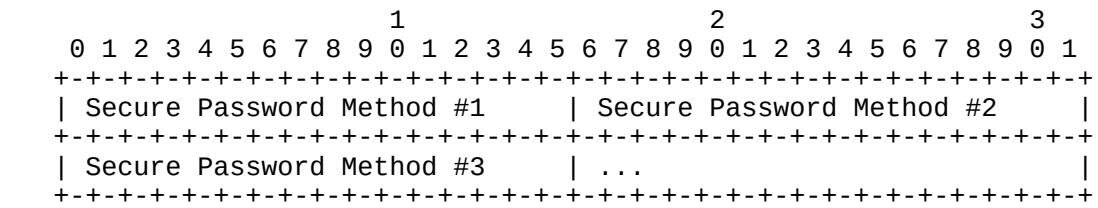
The secure password methods are identified by the 16-bit IANA-allocated numbers stored in the Notify payload notification data field. If a method supports multiple different password preprocessing methods, each of those may be allocated a separate number from this space, or the method might do its own negotiation of the preprocessing method later. As the initiator has already selected the shared secret it will be using, it will also know which preprocessing it might need, so it should propose only those preprocessing methods suitable for the selected shared secret. This means that it is recommended that separate IANA numbers be allocated for different preprocessing methods.

The actual Notify payload will look like this:



The Protocol ID will be zero, and the SPI Size will also be zero, meaning that the SPI field will be empty. The Notify Message Type will be 16424.

The Notification Data contains the list of the 16-bit secure password method numbers:

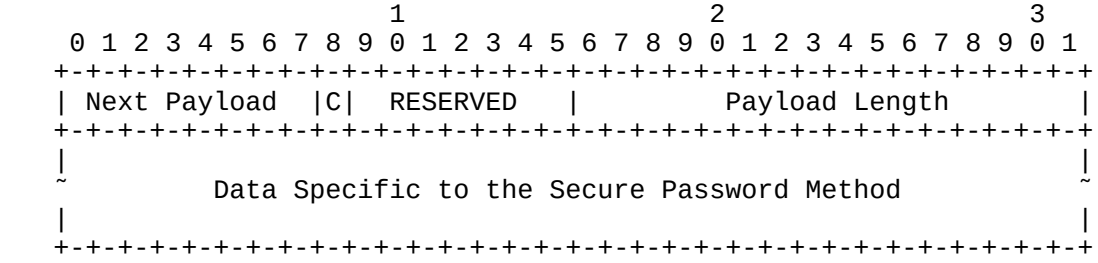


The response Notify payload contains exactly one 16-bit secure password method number inside the Notification Data field.

3. Generic Secure Password Method Payload

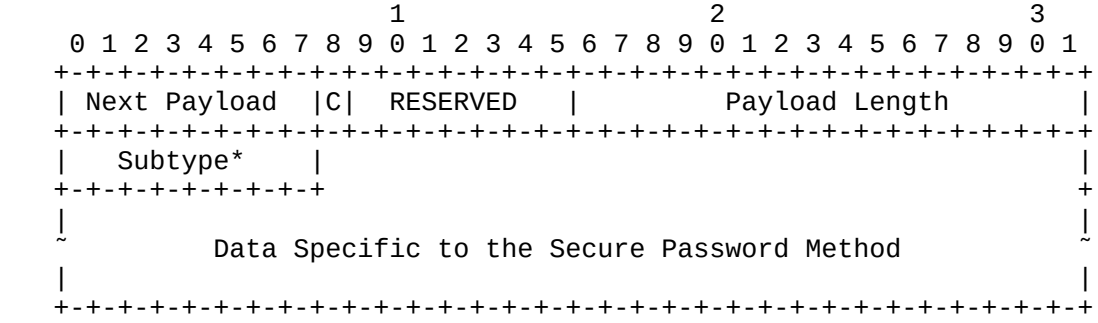
This payload will contain the data that is specific to the secure password payload. The IKE_AUTH exchanges might have a number of these inside, depending on what is required and specified by the secure password method. As the secure password method is already selected during IKE_SA_INIT, there is no need to repeat the information of the selected secure password method; thus, this payload only contains the method-specific data. As some secure password methods require multiple different payloads, they are assumed to include their method-specific payload type inside the payload -- for example, inside the first octet of the data. However, this is method-specific, and a method is free to format the payload data as it wants.

The Generic Secure Password Method (GSPM) payload will look like this:



The Payload Type for this payload is 49, and the name used in this document is "GSPM payload."

If the method uses payload subtypes (which are specific to the secure password method) inside the GSPM payload, the format will be like this:



* method-specific subtype field

This representation is here only for illustrative purposes; the secure password method will define the exact format of the payload contents.

4. IKE_AUTH Exchange

As the negotiation takes place during IKE_SA_INIT, the secure password methods may modify the IKE_AUTH exchange if needed. To easily enable implementing multiple methods, it is recommended that IKE_AUTH exchange not be modified unnecessarily. Adding zero, one, or multiple GSPM payloads to each exchange is needed, as is the modification to how the AUTH payload is calculated, but all other changes should be kept minimal.

The IKE_AUTH exchange should look similar to when EAP is used, meaning that the first request includes IDi, SAi2, TSi, TSr, and some number of GSPM payloads. The response should include IDr and, again, a number of GSPM payloads. There may be multiple exchanges, each consisting of some number of GSPM payloads; finally, when authentication is done, there should be one final exchange where the request includes the AUTH payload (along with some number of GSPM payloads) and the response contains AUTH, SAr2, TSi, TSr, and some number of GSPM payloads. The number of GSPM payloads is up to the secure password method but usually will be less than 3. However, depending on the method, it might be more.

The AUTH payload calculation should include all the data that would normally be included, in addition to the extra data needed by the secure password method. The secure password method needs to define how the AUTH payload is calculated.

As the AUTH payload calculation is changed, the secure payload method should not use any of the existing authentication method numbers in the AUTH Payload Auth Method field but instead should use the number allocated in this document. This number is meant to be used by all secure password authentication methods.

```
Initiator                          Responder
-----------------------------------------------------------------
HDR(SPIi=xxx, SPIr=yyy, IKE_AUTH,
    Flags: Initiator, Message ID=1),
    SK {IDi, [CERTREQ,]
        GSPM, [GSPM, ...,]
        [IDr,] SAi2,
        TSi, TSr}  -->

                   <--  HDR(SPIi=xxx, SPIr=yyy, IKE_AUTH, Flags:
                              Response, Message ID=1),
                              SK {IDr, [CERT,]
                                  GSPM, [GSPM, ...]}

HDR(SPIi=xxx, SPIr=yyy, IKE_AUTH,
    Flags: Initiator, Message ID=2),
    SK {GSPM, [GSPM, ...,]}  -->

                   <--  HDR(SPIi=xxx, SPIr=yyy, IKE_AUTH, Flags:
                              Response, Message ID=2),
                              SK {GSPM, [GSPM, ...]}
...

HDR(SPIi=xxx, SPIr=yyy, IKE_AUTH,
    Flags: Initiator, Message ID=x),
    SK {[GSPM, ...,], AUTH}  -->

                   <--  HDR(SPIi=xxx, SPIr=yyy, IKE_AUTH, Flags:
                              Response, Message ID=x),
                              SK {[GSPM, ...,] AUTH, SAr2,
                                  TSi, TSr}
```

Note that the number of the GSPM payloads and other payloads in each packet will be defined only by the secure password method documentation, and representations in this document are only for illustrative purposes.

5. Security Considerations

As this document does not describe an exact protocol, the security considerations are not relevant. Any secure password method documentation using payload types described here needs to also describe the security properties of the protocol it defines or discusses.

6. IANA Considerations

This document allocates one new IKEv2 message type in the "Notify Messages Types - Status Types" registry:

16424	SECURE_PASSWORD_METHODS

This document also allocates one new number in the "IKEv2 Authentication Method" registry:

12	Generic Secure Password Authentication Method

This document also adds one new payload type to the "IKEv2 Payload Types" registry:

49	Generic Secure Password Method	GSPM

This document creates a new IANA registry -- "IKEv2 Secure Password Methods":

0	Reserved

Values 1-1023 are unassigned. Values 1024-65535 are for private use among mutually consenting parties. Changes and additions to this registry are done by Expert Review.

7. References

7.1. Normative References

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

7.2. Informative References

[PACE] Kuegler, D. and Y. Sheffer, "Password Authenticated Connection Establishment with IKEv2", Work in Progress, September 2011.

[PAKE] Shin, S. and K. Kobara, "Most Efficient Augmented Password-Only Authentication and Key Exchange for IKEv2", Work in Progress, July 2011.

[SPSK-AUTH] Harkins, D., "Secure PSK Authentication for IKE", Work in Progress, July 2011.

Author’s Address

Tero Kivinen
AuthenTec
Eerikinkatu 28
HELSINKI FI-00180
Finland

EMail: kivinen@iki.fi