EAP IKEv2 Method (EAP-IKEv2)

Abstract

EAP-IKEv2 is an EAP method which reuses the cryptography and the payloads of IKEv2, creating a flexible EAP method that supports both symmetric and asymmetric authentication. Furthermore protection of legacy authentication mechanisms is supported. This EAP method offers the security benefits of IKEv2 without the goal of establishing IPsec security associations.

Table of Contents

1. Introduction......................................................2
2. Terminology......................................................2
3. Protocol overview..............................................3
4. Packet Format..................................................6
1. Introduction

IKEv2 [2] is a protocol which consists of two phases:

(1) an authentication and key exchange protocol which establishes an IKE-SA.

(2) messages and payloads which focus on the negotiation of parameters in order to establish IPsec security associations (i.e. Child-SAs). These payloads contain algorithm parameters and traffic selector fields.

In addition to the above-mentioned parts IKEv2 also includes some payloads and messages which allow configuration parameters to be exchanged primarily for remote access scenarios.

The EAP-IKEv2 method defined by this document primarily uses the IKEv2 payloads and messages used for phase (1).

IKEv2 provides an improvement over IKEv1 [5] as described in Appendix A of [2]. Important for this document are the reduced number of initial exchanges, support of legacy authentication, decreased latency of the initial exchange, optional DoS protection capability and some other fixes (e.g. hash problem). IKEv2 is a cryptographically sound protocol that has received a considerable amount of expert review and that benefits from a long practical experience with IKE.

The goal of EAP-IKEv2 is to inherit these properties within an efficient, secure EAP method.

In addition, IKEv2 provides authentication and key exchange capabilities which allow an entity to use symmetric as well as asymmetric authentication in addition to legacy authentication support within a single protocol. Such flexibility is considered important for an EAP method and is provided by EAP-IKEv2.

[6] provides a good tutorial for IKEv2 design decisions.

2. Terminology

This document does not introduce new terms other than those defined in [1] or in [2].
3. Protocol overview

The protocol for symmetric and asymmetric authentication within EAP-IKEv2 differs from IKEv2 only in the computation of the AUTH payload. For symmetric authentication no CERT and CERTREQ payloads are required. Figure 2 depicts such a protocol exchange. The following types of identities are used in this exchange:

a) The first part of the (outer) EAP message exchange provides information where the protocol messages described below terminate. This message exchange primarily is an identity request/response.

b) The identities used within EAP-IKEv2 for both the initiator and the responder. The initiator identity is often associated with a user identity such as a fully-qualified RFC 822 email address. The identity of the responder might be a FQDN. The identity is of importance for authorization.

For secure legacy authentication an EAP message exchange is protected with the established IKE-SA as shown in Figure 3. This exchange again adds EAP identities.

c) This inner EAP message exchange primarily serves the purpose of client authentication. The two identities used thereby are the EAP identity (i.e. a NAI) and possibly a separate identity for the selected EAP method.

The large number of identities is required due to nesting of authentication methods and due to overloaded function of the identity for routing (i.e. authentication end point indication).

Hence with this additional (nested) EAP exchange the end point of the EAP-IKEv2 exchange might not be the same as the end point of the inner EAP exchange which is protected by the IKE-SA (which in this case is not protected by the IKE-SA any more between the EAP-IKEv2 endpoint and the endpoint of the inner EAP exchange, but might be protected by other means that are not considered in this document).

This section illustrates some message exchanges for use in EAP-IKEv2. They are taken from [2], and are adapted. Since this document does not describe frameworks or particular architectures the message exchange takes place between two parties - between the Initiator (I) and the Responder (R). In context of EAP the Initiator is often called Authenticating Peer whereas the Responder is referred as Authenticator.
The first message flow shows EAP-IKEv2 without the optional DoS protection exchanges. The core EAP-IKEv2 exchange consists of four messages (two roundtrips) only:

I <-- R: EAP-Request/Identity

I -- R: EAP-Response/Identity(Id)

I <-- R: EAP-Request/EAP-Type=EAP-IKEv2(Start)

I -- R: EAP-Response/EAP-Type=EAP-IKEv2(HDR(A,0), SAi1, KEi, Ni)

I <-- R: EAP-Request/EAP-Type=EAP-IKEv2(HDR(A,B), SAr1, KEr, Nr, [CERTREQ])

I -- R: EAP-Response/EAP-Type=EAP-IKEv2(HDR(A,B), SK {IDi, [CERT,] [CERTREQ,] [IDr,], AUTH})

I <-- R: EAP-Request/EAP-Type=EAP-IKEv2(HDR(A,B), SK {IDr, [CERT,] AUTH})

I -- R: EAP-Response/EAP-Type=EAP-IKEv2(Finish)

I <-- R: EAP-Success

Figure 1: EAP-IKEv2 message flow

The subsequent message flow shows EAP-IKEv2 with the optional DoS protection. The IKEv2 DoS protection mechanism uses cookies and keeps the responder stateless when it receives the first IKEv2 message. As a consequence of DoS protection an additional roundtrip is required.

I <-- R: EAP-Request/Identity

I -- R: EAP-Response/Identity(Id)

I <-- R: EAP-Request/EAP-Type=EAP-IKEv2(Start)

I -- R: EAP-Response/EAP-Type=EAP-IKEv2(HDR(A,0), SAi1, KEi, Ni)

I <-- R: EAP-Request/EAP-Type=EAP-IKEv2(HDR(A,0), NCOOKIE-REQUIRED), NCOOKIE)

I -- R: EAP-Response/EAP-Type=EAP-IKEv2(HDR(A,0), NCOOKIE), SAi1, KEi, Ni)

I <-- R: EAP-Request/EAP-Type=EAP-IKEv2(HDR(A,B), SAr1, KEr, Nr, [CERTREQ])

The following EAP message exchange is taken from Section 2.16 of [2] and adapted (TSi, TSr, SAi2 and SAr2 payloads are omitted). It provides an example of a successful inner EAP exchange using the EAP-SIM Authentication method [9], which is secured by the IKE-SA.

I <-- R: EAP-Request/Identity

I --> R: EAP-Response/Identity(Id)

I <-- R: EAP-Request/EAP-Type=EAP-IKEv2(Start)

I --> R: EAP-Response/EAP-Type=EAP-IKEv2(HDR, SAi1, KEi, Ni)

I <-- R: EAP-Request/EAP-Type=EAP-IKEv2(HDR, SAr1, KEr, Nr, [CERTREQ])

I --> R: EAP-Response/EAP-Type=EAP-IKEv2(HDR, SK {IDi, [CERTREQ], [IDr],})

I <-- R: EAP-Request/EAP-Type=EAP-IKEv2(HDR, SK {IDr, [CERT], AUTH, EAP(EAP-Request/Identity))

I --> R: EAP-Response/EAP-Type=EAP-IKEv2(HDR, SK {EAP(EAP-Response/Identity), [AUTH]})

I <-- R: EAP-Request/EAP-Type=EAP-IKEv2(HDR, SK {EAP(EAP-Request/SIM/Start(Version_list)),[AUTH]})

I --> R: EAP-Response/EAP-Type=EAP-IKEv2(HDR, SK {EAP(EAP-Response/SIM/Start(Nonce_MT, Selected_VERSION)), [AUTH]})

I <-- R: EAP-Request/EAP-Type=EAP-IKEv2(HDR, SK {EAP(EAP-Request/SIM/Challenge(RAND, MAC)), [AUTH]})

I --> R: EAP-Response/EAP-Type=EAP-IKEv2(Finish)
I <-- R: EAP-Success

Figure 3: EAP-IKEv2 SLA with EAP-SIM Authentication

Since the goal of this EAP method is not to establish an IPsec SA some payloads used in IKEv2 are omitted. In particularly the following messages and payloads are not required:

- Traffic Selectors
- IPsec SA negotiation payloads
  (e.g. CREATE_CHILD_SA exchange or SAx2 payloads)
- ECN Notification
- Requesting an internal address on a remote network
  (aka ModeCfg or DHCP-based approaches)
- Port handling
- NAT traversal
- Rekeying of IKE-SAs

Some of these messages and payloads are optional in IKEv2. In general it does not make sense to directly negotiate IPsec SAs with EAP-IKEv2, as such SAs were unlikely to be used between the EAP endpoints.

4. Packet Format

The IKEv2 payloads, which are defined in [2], are embedded into the Data field of the standard EAP Request/Response packets. The Code, Identifier, Length and Type field is described in [1]. The Type-Data field carries a one byte Flags field following the IKEv2 payloads. Each IKEv2 payload starts with a header field HDR (see [2]).

The packet format is shown in Figure 4:

```
  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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Code      |   Identifier  |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |   Flags       |  Data...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
Figure 4: Packet Format
```

No additional packet formats other than those defined in [2] are required for this EAP method.
The Flags field is required to indicate Start and Finish messages which are required due to the asymmetric nature of IKEv2 and the Request/Response message handling of EAP.

Currently only two bits in the eight bit Flags field are required. The remaining bits are set to zero.

```
0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
|S F 0 0 0 0 0 0|
+-+-+-+-+-+-+-+-+
```

S = EAP-IKEv2 start message
F = EAP-IKEv2 finish message

EAP-IKEv2 messages which have neither an S or an F flag set contain regular IKEv2 message payloads inside the Data field.

The EAP Type for this EAP method is <TBD>.

5. Key derivation

The EAP-IKEv2 method described in this document generates session keys. These session keys are used to establish an IKE-SA which provides protection of other payloads. To export a session key as part of the EAP keying framework [7] it is required to derive another session key for usage with EAP (sometimes referred as Pre-Master-Secret). It is good cryptographic security practice to use different keys for different "applications". Hence we suggest to reuse the key derivation function suggested in Section 2.17 of [2] to export the KEYMAT (as a Pre-Master-Secret) for further key derivation.

The key derivation function defined is KEYMAT = prf+(SK_d, Ni | Nr), where Ni and Nr are the Nonces from the IKE_SA_INIT exchange.

6. Security Considerations

The security of the proposed EAP method is intentionally based on IKEv2 [2]. Man-in-the-middle attacks discovered in the context of tunneled authentication protocols (see [3] and [4]) are applicable to IKEv2 if legacy authentication with EAP [1] is used. To counter this threat IKEv2 provides a compound authentication by including the EAP provided session key inside the AUTH payload.

Further security considerations will be provided with future versions of this document.
7. Open Issues

The following open issues have been identified:

- Fragmentation

Currently it is not clear whether fragmentation support has to be provided although IKEv2 itself does not worry about fragmentation itself.

- Certificate revocation

IKEv2 allows CRLs to be exchanged. However, it does not provide the capability to exchange OCSP [8] payloads.

- Session resumption

TLS provides the capability of resuming a session. This offers primarily performance improvement for a new authentication and key exchange protocol run. It is for further study whether the concept of session resumption (i.e. a fast re-authentication procedure) is a useful context for EAP methods and for the AAA environment in particular. If it turns out to be useful then one possible approach is to reuse the dead peer detection informational exchange with is able to provide fast re-authentication based on the established IKE-SA. This exchange is cheap in terms of processing complexity and provides both end points the capability to perform authentication based on an available IKE-SA.

- Reducing the number of messages

The message flows given in this document finish with an EAP-Success message. Depending on the outcome of the current mailing list discussions it is possible that the EAP-Success message can be omitted if other success indications are available; e.g. mutual authentication is provided by the IKEv2 method. Furthermore it is possible to omit the first exchange if the identity can be learned by other means.

8. References


Acknowledgments

Add your name here.

Author’s Addresses

Hannes Tschofenig
Siemens AG
Otto-Hahn-Ring 6
81739 Munich
Germany
EMail: Hannes.Tschofenig@siemens.com

Dirk Kroeselberg
Siemens AG
Otto-Hahn-Ring 6
81739 Munich
Germany
EMail: Dirk.Kroeselberg@siemens.com