IPSEC
Internet-Draft
Expires: November 4, 2004

P. Eronen
Nokia
H. Tschofenig
Siemens

May 6, 2004

Extension for EAP Authentication in IKEv2
draft-eronen-ipsec-ikev2-eap-auth-01.txt

Status of this Memo

This Internet-Draft will expire on November 4, 2004.

Copyright Notice

Abstract

IKEv2 specifies that EAP authentication must be used together with public key signature based responder authentication. This is necessary with old EAP methods that provide only unilateral authentication using e.g. one-time passwords or token cards.

This document specifies how EAP methods that provide mutual authentication and key agreement can be used to provide extensible responder authentication for IKEv2 based on other methods than public key signatures.

1. Introduction

The Extensible Authentication Protocol (EAP), defined in [7], is an authentication framework which supports multiple authentication mechanisms. Today, EAP has been implemented at end hosts and routers that connect via switched circuits or dial-up lines using PPP [16], IEEE 802 wired switches [10], and IEEE 802.11 wireless access points [12].

One of the advantages of the EAP architecture is its flexibility. EAP is used to select a specific authentication mechanism, typically after the authenticator requests more information in order to determine the specific authentication method to be used. Rather than requiring the authenticator (e.g., wireless LAN access point) to be updated to support each new authentication method, EAP permits the use of a backend authentication server which may implement some or all authentication methods.

IKEv2 [3] is a component of IPsec used for performing mutual authentication and establishing and maintaining security associations for IPsec ESP and AH. In addition to supporting authentication using public key signatures and shared secrets, IKEv2 also supports EAP authentication.

IKEv2 provides EAP authentication since it was recognized that public key signatures and shared secrets are not flexible enough to meet the requirements of many deployment scenarios. By using EAP, IKEv2 can leverage existing authentication infrastructure and credential databases, since EAP allows users to choose a method suitable for existing credentials, and also makes separation of the IKEv2 responder (VPN gateway) from the EAP authentication endpoint (backend AAA server) easier.

Some older EAP methods are designed for unilateral authentication only (that is, EAP peer to EAP server). These methods are used in conjunction with IKEv2 public key based authentication of the responder to the initiator. It is expected that this approach is especially useful for "road warrior" VPN gateways that use, for instance, one-time passwords or token cards to authenticate the clients.

However, most newer EAP methods, such as those typically used with

IEEE 802.11i wireless LANs, provide mutual authentication and key agreement. Currently, IKEv2 specifies that also these EAP methods must be used together with public key signature based responder authentication.

In some environments, requiring the deployment of PKI for just this

purpose can be counterproductive. Deploying new infrastructure can be expensive, and it may weaken security by creating new vulnerabilities. Mutually authenticating EAP methods alone can provide a sufficient level of security in many circumstances, and indeed, IEEE 802.11i uses EAP without any PKI for authenticating the WLAN access points.

This document specifies how EAP methods that offer mutual authentication and key agreement can be used to provide responder authentication in IKEv2 completely based on EAP.

1.1 Terminology

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 [2].

2. Scenarios

In this section we describe two scenarios for extensible authentication within IKEv2. These scenarios are intended to be illustrative examples rather than specifying how things should be done.

Figure 1 shows a configuration where the EAP and the IKEv2 endpoints are co-located. Authenticating the IKEv2 responder using both EAP and public key signatures is redundant. Offering EAP based authentication has the advantage that multiple different authentication and key exchange protocols are available with EAP with different security properties (such as strong password based protocols, protocols offering user identity confidentiality and many more). As an example it is possible to use GSS-API support within EAP [5] to support Kerberos based authentication which effectively replaces the need for KINK [17].

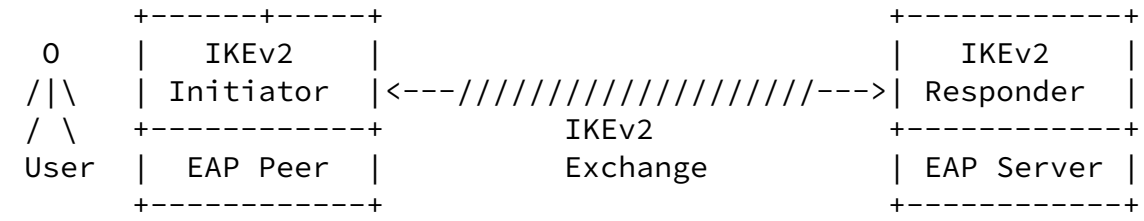


Figure 1: EAP and IKEv2 endpoints are co-located

Figure 2 shows a typical corporate network access scenario. The initiator (client) interacts with the responder (VPN gateway) in the corporate network. The EAP exchange within IKE runs between the client and the home AAA server. As a result of a successful EAP

authentication protocol run, session keys are established and sent from the AAA server to the VPN gateway, and then used to authenticate the IKEv2 SA with AUTH payloads.

The protocol used between the VPN gateway and AAA server could be, for instance, Diameter [7] or RADIUS [4]. See Section 5 for related security considerations.

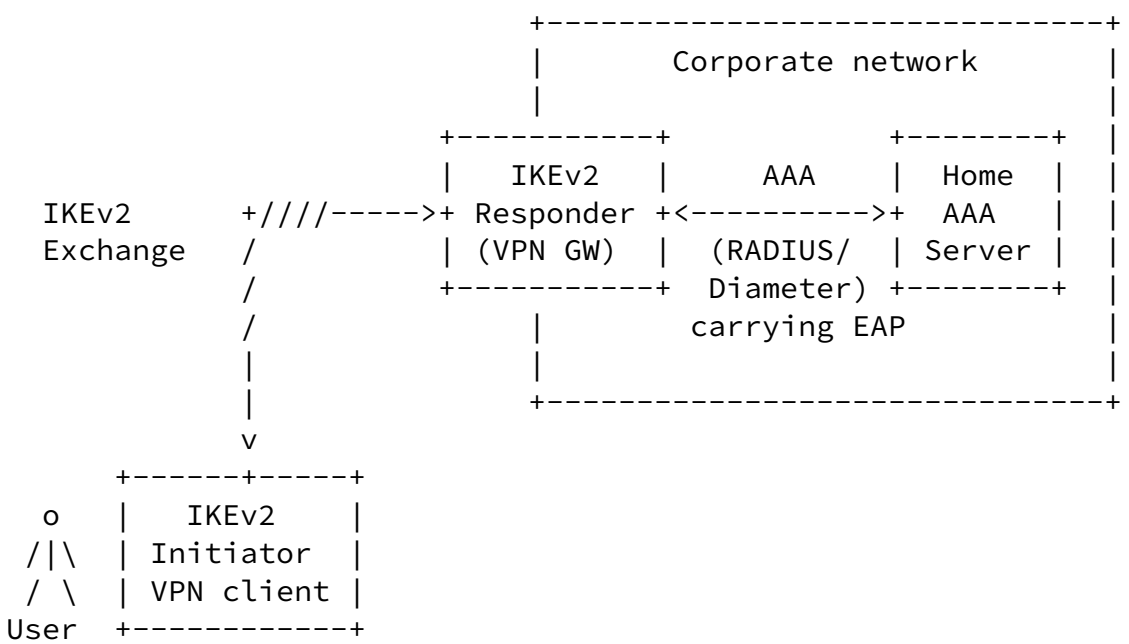


Figure 2: Corporate Network Access

3. Solution

IKEv2 specifies that when the EAP method establishes a shared secret key, that key is used by both the initiator and responder to generate an AUTH payload (thus authenticating the IKEv2 SA set up by messages 1 and 2).

When used together with public key responder authentication, the responder is in effect authenticated using two different methods: the public key signature AUTH payload in message 4, and the EAP-based AUTH payload later.

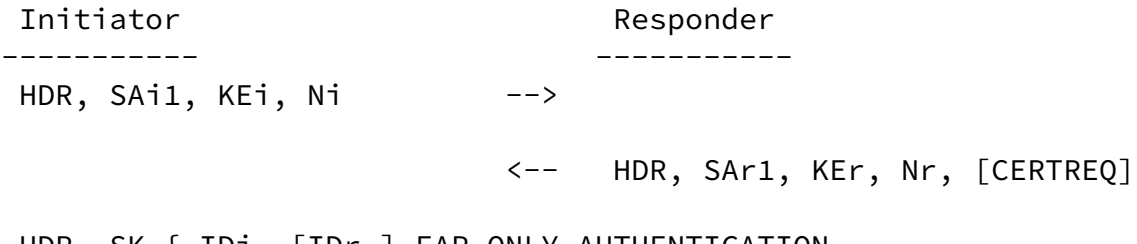
If the initiator does not wish to use public key based responder authentication, it includes an EAP_ONLY_AUTHENTICATION notification payload (type TBD-BY-IANA) in message 3. There is no additional data associated with this notification.

If the responder supports this notification, it omits the public key based AUTH payload and CERT payloads from message 4.

If the responder does not support the EAP_ONLY_AUTHENTICATION notification, it ignores the notification payload, and includes the AUTH payload in message 4. In this case the initiator can, based on its local policy, choose to either ignore the AUTH payload, or verify it and any associated certificates as usual.

Both the initiator and responder MUST verify that the EAP method actually used provided mutual authentication and established a shared secret key. The AUTH payloads sent after EAP Success MUST use the EAP-generated key, and MUST NOT use SK_pi or SK_pr.

An IKEv2 message exchange with this modification is shown below:



```
                    SAi2, TSi, TSr}  -->

                                     <--   HDR, SK { IDr, EAP(Request) }

          HDR, SK { EAP(Response) }  -->

                                     <--   HDR, SK { EAP(Request) }

          HDR, SK { EAP(Response) }  -->

                                     <--   HDR, SK { EAP(Success) }

          HDR, SK { AUTH }  -->

                                     <--   HDR, SK { AUTH, SAr2, TSi, TSr }
```

4. IANA considerations

This document defines a new IKEv2 Notification Payload type, EAP_ONLY_AUTHENTICATION, described in Section 3. This payload must be assigned a new type number from the "status types" range.

This document does not define any new namespaces to be managed by IANA.

5. Security Considerations

Security considerations applicable to all EAP methods are discussed in [1]. The EAP Key Management Framework [6] deals with issues that arise when EAP is used as a part of a larger system.

5.1 Authentication of IKEv2 SA

It is important to note that the IKEv2 SA is not authenticated by just running an EAP conversation: the crucial step is the AUTH payload based on the EAP-generated key. Thus, EAP methods that do not provide mutual authentication or establish a shared secret key MUST NOT be used with the modifications presented in this document.

5.2 Authentication with separated IKEv2 responder/EAP server

As described in Section 2, the EAP conversation can terminate either at the IKEv2 responder or at a backend AAA server.

If the EAP method terminates at the IKEv2 responder then no key transport via the AAA infrastructure is required. Pre-shared secret and public key based authentication offered by IKEv2 is then replaced by a wider range of authentication and key exchange methods.

However, typically EAP will be used with a backend AAA server. See [6] for a more complete discussion of the related security issues; here we provide only a short summary.

When a backend server is used, there are actually two authentication exchanges: the EAP method between the client and the AAA server, and another authentication between the AAA server and IKEv2 gateway. The AAA server authenticates the client using the selected EAP method, and they establish a session key. The AAA server then sends this key to the IKEv2 gateway over a connection authenticated using e.g. IPsec or TLS.

Some EAP methods do not have any concept of pass-through authenticator (e.g. NAS or IKEv2 gateway) identity, and these two authentications remain quite independent of each other. That is, after the client has verified the AUTH payload sent by the IKEv2 gateway, it knows that it is talking to SOME gateway trusted by the home AAA server, but not which one. The situation is somewhat similar if a single cryptographic hardware accelerator, containing a single private key, would be shared between multiple IKEv2 gateways (perhaps in some kind of cluster configuration). In particular, if one of the gateways is compromised, it can impersonate any of the other gateways towards the user (until the compromise is discovered and access rights revoked).

In some environments it is not desirable to trust the IKEv2 gateways this much (also known as the "Lying NAS Problem"). EAP methods that provide what is called "connection binding" or "channel binding" transport some identity or identities of the gateway (or WLAN access point/NAS) inside the EAP method. Then the AAA server can check that it is indeed sending the key to the gateway expected by the client.

In some deployment configurations, AAA proxies may be present between the IKEv2 gateway and the backend AAA server. These AAA proxies MUST be trusted for secure operation, and therefore SHOULD be avoided when possible; see [7] and [6] for more discussion.

5.3 Protection of EAP payloads

Although the EAP payloads are encrypted and integrity protected with SK_e/SK_a, this does not provide any protection against active attackers. Until the AUTH payload has been received and verified, a man-in-the-middle can change the KEi/KEr payloads and eavesdrop or modify the EAP payloads.

In IEEE 802.11i WLANs, the EAP payloads are neither encrypted nor integrity protected (by the link layer), so EAP methods are typically designed to take that into account.

In particular, EAP methods that are vulnerable to dictionary attacks when used in WLANs are still vulnerable (to active attackers) when run inside IKEv2.

5.4 User identity confidentiality

IKEv2 provides confidentiality for the initiator identity against passive eavesdroppers, but not against active attackers. The initiator announces its identity first (in message #3), before the responder has been authenticated. The usage of EAP in IKEv2 does not change this situation, since the ID payload in message #3 is used instead of the EAP Identity Request/Response exchange. This is somewhat unfortunate since when EAP is used with public key authentication of the responder, it would be possible to provide active user identity confidentiality for the initiator.

IKEv2 protects the responder identity even against active attacks. This property cannot be provided when using EAP. If public key responder authentication is used in addition to EAP, the responder reveals its identity before authenticating the initiator. If only EAP is used (as proposed in this document), the situation depends on the EAP method used (in some EAP methods, the server reveals its identity first).

Hence, if active user identity confidentiality for the initiator is required then EAP methods that offer this functionality have to be used (see [1], Section 7.3).

6. Acknowledgments

This document borrows some text from [1], [3], and [7]. We would also like to thank Hugo Krawczyk for interesting discussions about this topic.

7. References

7.1 Normative References

[1] Blunk, L., Vollbrecht, J., Aboba, B., Carlson, J. and H. Levkowetz, "Extensible Authentication Protocol (EAP)", draft-ietf-eap-rfc2284bis-09 (work in progress), February 2004.

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

[3] Kaufman, C., "Internet Key Exchange (IKEv2) Protocol", draft-ietf-ipsec-ikev2-13 (work in progress), March 2004.

7.2 Informative References

[4] Aboba, B. and P. Calhoun, "RADIUS (Remote Authentication Dial In User Service) Support For Extensible Authentication Protocol (EAP)", RFC 3579, September 2003.

[5] Aboba, B. and D. Simon, "EAP GSS Authentication Protocol", draft-aboba-pppext-eapgss-12 (work in progress), April 2002.

[6] Aboba, B., Simon, D., Arkko, J. and H. Levkowetz, "EAP Key Management Framework", draft-ietf-eap-keying-01 (work in progress), October 2003.

[7] Eronen, P., Hiller, T. and G. Zorn, "Diameter Extensible Authentication Protocol (EAP) Application", draft-ietf-aaa-eap-05 (work in progress), April 2004.

[8] Forsberg, D., Ohba, Y., Patil, B., Tschofenig, H. and A. Yegin, "Protocol for Carrying Authentication for Network Access (PANA)", draft-ietf-pana-pana-03 (work in progress), February 2004.

[9] Funk, P. and S. Blake-Wilson, "EAP Tunneled TLS Authentication Protocol (EAP-TTLS)", draft-ietf-pppext-eap-ttls-04 (work in

progress), April 2004.

[10] Institute of Electrical and Electronics Engineers, "Local and Metropolitan Area Networks: Port-Based Network Access Control", IEEE Standard 802.1X-2001, 2001.

[11] Institute of Electrical and Electronics Engineers, "Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications", IEEE Standard 802.11-1999, 1999.

[12] Institute of Electrical and Electronics Engineers, "IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements - Part 11: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications: Amendment 6: Medium Access Control (MAC) Security Enhancements", IEEE Draft 802.11i/D10.0, April 2004.

[13] Josefsson, S., Palekar, A., Simon, D. and G. Zorn, "Protected EAP Protocol (PEAP)", draft-josefsson-pppext-eap-tls-eap-07 (work in progress), October 2003.

[14] Puthenkulam, J., "The Compound Authentication Binding Problem", draft-puthenkulam-eap-binding-04 (work in progress), October 2003.

[15] Rigney, C., Willens, S., Rubens, A. and W. Simpson, "Remote Authentication Dial In User Service (RADIUS)", RFC 2865, June 2000.

[16] Simpson, W., "The Point-to-Point Protocol (PPP)", STD 51, RFC 1661, July 1994.

[17] Thomas, M. and J. Vilhuber, "Kerberized Internet Negotiation of Keys (KINK)", draft-ietf-kink-kink-05 (work in progress), January 2003.

[18] Tschofenig, H., Kroeselberg, D. and Y. Ohba, "EAP IKEv2 Method (EAP-IKEv2)", draft-tschofenig-eap-ikev2-03 (work in progress), February 2004.

Authors' Addresses

Pasi Eronen
Nokia Research Center
P.O. Box 407
FIN-00045 Nokia Group
Finland

EMail: pasi.eronen@nokia.com

Hannes Tschofenig
Siemens
Otto-Hahn-Ring 6
Munich, Bayern 81739
Germany

EMail: Hannes.Tschofenig@siemens.com

Appendix A. Alternative Approaches

In this section we list alternatives which have been considered during the work on this document. Finally, the solution presented in Section 3 seems to fit better into IKEv2.

A.1 Ignore AUTH payload at the initiator

With this approach, the initiator simply ignores the AUTH payload in message #4 (but obviously must check the second AUTH payload later!). The main advantage of this approach is that no protocol modifications are required and no signature verification is required.

The initiator could signal the responder (using a NOTIFY payload) that it did not verify the first AUTH payload.

A.2 Unauthenticated PKs in AUTH payload (message 4)

The first solution approach suggests the use of unauthenticated public keys in the public key signature AUTH payload (for message 4).

That is, the initiator verifies the signature in the AUTH payload, but does not verify that the public key indeed belongs to the intended party (using certificates)--since it doesn't have a PKI that would allow this. This could be used with X.509 certificates (the initiator ignores all other fields of the certificate except the public key), or "Raw RSA Key" CERT payloads.

This approach has the advantage that initiators that wish to perform

certificate-based responder authentication (in addition to EAP) may do so, without requiring the responder to handle these cases separately.

If using RSA, the overhead of signature verification is quite small (compared to g^xy calculation).

A.3 Use EAP derived session keys for IKEv2

It has been proposed that when using an EAP methods that provides mutual authentication and key agreement, the IKEv2 Diffie-Hellman exchange could also be omitted. This would mean that the sessions keys for IPsec SAs established later would rely only on EAP-provided keys.

It seems the only benefit of this approach is saving some computation time (g^xy calculation). This approach requires designing a completely new protocol (which would not resemble IKEv2 anymore) we do not believe that it should be considered. Nevertheless, we include it for completeness.

Intellectual Property Statement

ietf-ipr@ietf.org.

Disclaimer of Validity

Copyright Statement

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.