Internet Draft
draft-ietf-ipsec-revised-identity-00.txt
April 30, 2002
Expires in six months

Editor: Paul Hoffman
VPN Consortium

Revised Use of Identity in Successors to IKE

Status of this memo

Abstract

There is an opportunity in successor-to-IKE to fix two major problems that have plagued IKEv1: a misunderstanding about what is identity, and having to send certificates every time because you don't know if the other party already has your certificate. This proposal covers both topics at once because it turns out that they are related.

1. Introduction

The discussion in this document is for the successor-to-IKE proposals. It does not use the certificate, certificate request, or ID payloads from IKEv1 [IKEV1].

This proposal could work in either IKEv2 [IKEV2] or JFK [JFK]. In this document, the message numbers refer to either the four-message version of IKEv2 or JFK.

In this document, the key words "MAY", "MUST, "MUST NOT", "SHOULD", and "SHOULD NOT", are to be interpreted as described in [RFC2119].

2. New payloads for messages 1 and 2

Messages 1 and 2 MAY include a TrustedRoot payload. The TrustedRoot payload includes a series of one or more PKIX [PKIX] keyIdentifiers of roots trusted by the sender. This payload is completely optional and is used only to inform the recipient of what capabilities the sender has.

Messages 1 and 2 MUST include exactly one IDAccepted payload. The payload holds a series of one or more fields indicating the FullID types that the sender will accept. The receiver MUST NOT send any FullID payloads in messages 3 or 4 that are not listed in the sender's IDAccepted payload.

3. New payloads for messages 3 and 4

Messages 3 and 4 MUST include exactly one FullID payload. The payload's format is an ID type followed by the content. The ID types are:

1 PKIX certificate
A standalone PKIX certificate.

2 Certificate bundle
A simple ASN.1 sequence of PKIX certificates. A bundle can have end-entity certificates or certificate chains. The first certificate in the bundle is the sender's preferred identity certificate, but beyond that there is no meaning to the ordering.

3 Hash-and-URL of PKIX certificate
The first 20 octets are the SHA-1 hash of a certificate; the rest is a URL that resolves to the certificate. This is described in more detail below.

4 URL to a PKIX certificate bundle
This is described in more detail below.

5 PKIX keyIdentifier
Identifies a self-signed certificate that the receiver has already pre-loaded. Note that this is only useful when using self-signed certificates.

6 IDForSharedSecret
This is only for use with shared secrets. It is an ASCII string (all octets are 31 < x < 127) of any length.

3.1 Using URLs and caching certificates

For FullID types 3 and 4, the URL scheme must be http, although it can be on any port number. Future versions of this document may add requirements for how the named part looks. The URL can be to a persistent repository, or it might be to the initiating machine (such as in a remote access client).

If a recipient uses FullID type 3, it might cache the certificate with the hash as an index, or the certificate can be retrieved from the URL. Of course, retrieving a certificate from a URL means many more round trips before the key exchange protocol can finish. On the other hand, if the certificate has been cached, no additional processing is needed and the certificate does not need to be sent in the UDP-based protocol.

If a system that is using certificates knows that it cannot resolve URLs (for example, because it is not yet on the Internet), it SHOULD use FullID types 1 and 2 in its IDAccepted payload. If a system can resolve URLs, it SHOULD use type 3 and 4 unless it is sure that it does not have the certificate of the other side, such as if it has just recovered from a crash and its cache is empty. All systems should be able to handle certificate bundles because the other party might have multiple identities which have different certificates.

3.2 Using legacy authentication

FullID type 6 (IDForSharedSecret) indirectly supports non-certificate, non-shared-secret authentication (commonly called "legacy authentication") with IKEv2. The authentication system must create two temporary tokens, one of which is used as the identity and the other is used as the secret. The IKEv2 system must have some way of securely querying the authentication system with the identity token and receiving back the secret token.

As simple scenario is username-and-password. The IDForSharedSecret is the username, and the key added to the HMAC is the password. When receiving this message, the system looks up the password based on the username, possibly using something like RADIUS.

A more complex scenario involves a hardware token that doesn't do public key cryptography. Typically, the user does some challenge-response exchange with the authenticating server and is then authenticated. At that point, as long as the user has two pieces of authentication information, they can use them in an HMAC. For systems that only return one item (such as a long hash), there must be some agreement on how to split that into two items, such as "80 bits for the IDForSharedSecret and 80 bits for the secret". That can be done by the legacy authentication vendor, possibly instantiated in an informational RFC.

4. New error codes

The use of this proposal causes the need for additional error codes:

- Could not get your certificate through the URL you gave

- Could not get your certificate bundle through the URL you gave

- The certificate bundle was malformed

- Could not find the certificate that matches the keyIdentifier you gave

- Could not use your IDForSharedSecret

5. Security Considerations

Sending a TrustedRoot payload exposes information about the sender of the payload to a passive attack. The attacker can determine information about the sender, such as which roots the sender trusts. For users in a corporate environment, the TrustedRoot payload may reveal who the sender's employer.

An attacker snooping on a receiver can learn the identity of the senders who use certificates that are not cached by the receiver by watching the HTTP traffic generated from the receiver.

6. References

[IKEV1] "Internet Key Exchange (IKE)", RFC 2409

[IKEV2] "Proposal for the IKEv2 Protocol", draft-ietf-ipsec-ikev2

[JFK] "Just Fast Keying (JFK)", draft-ietf-ipsec-jfk

[PKIX] "Internet X.509 Public Key Infrastructure Certificate and CRL Profile", RFC 2459

[RFC2119] "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119

7. Editor's Address

Paul Hoffman
VPN Consortium
127 Segre Place
Santa Cruz, CA 95060 USA
paul.hoffman@vpnc.org