Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: 17 September 2026

V. Smyslov
ELVIS-PLUS
C. Patton
Cloudflare

16 March 2026

Downgrade Prevention for the Internet Key Exchange Protocol Version 2 (IKEv2)

draft-ietf-ipsecme-ikev2-downgrade-prevention-02

Abstract

This document describes an extension to the Internet Key Exchange protocol version 2 (IKEv2) that prevents particular downgrade attacks on this protocol by having the peers confirm they have participated in the same conversation.

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 https://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 17 September 2026.

Table of Contents

1. Introduction

The Internet Key Exchange version 2 protocol (IKEv2) defined in [RFC7296] provides authenticated key exchange in the IP Security (IPsec) architecture. The cryptographic design of IKEv2 is based on the SIGn-and-MAc (SIGMA) protocol defined in [SIGMA]. The protocol allows peers to mutually authenticate themselves and to derive session keys that are used to protect traffic.

(RFC EDITOR: Please remove this paragraph.) This document is being developed at https://github.com/smyslov/ikev2-downgrade-prevention.

2. Terminology and Notation

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

It is assumed that readers are familiar with the IKEv2 protocol [RFC7296].

3. Authentication in IKEv2

The details of how authentication is performed in IKEv2 are defined in Section 2.15 of [RFC7296]. Peers sign (or MAC) some blocks of data that consist of various parts of protocol data (see also [SIGMA] for the rationale). The definition of these blocks of data is provided below for convenience.

The initiator’s signed octets can be described as:

```
InitiatorSignedOctets = RealMessage1 | NonceRData | MACedIDForI
GenIKEHDR = [ four octets 0 if using port 4500 ] | RealIKEHDR
RealIKEHDR =  SPIi | SPIr |  . . . | Length
RealMessage1 = RealIKEHDR | RestOfMessage1
NonceRPayload = PayloadHeader | NonceRData
InitiatorIDPayload = PayloadHeader | RestOfInitIDPayload
RestOfInitIDPayload = IDType | RESERVED | InitIDData
MACedIDForI = prf(SK_pi, RestOfInitIDPayload)
```

The responder’s signed octets can be described as:

```
ResponderSignedOctets = RealMessage2 | NonceIData | MACedIDForR
GenIKEHDR = [ four octets 0 if using port 4500 ] | RealIKEHDR
RealIKEHDR =  SPIi | SPIr |  . . . | Length
RealMessage2 = RealIKEHDR | RestOfMessage2
NonceIPayload = PayloadHeader | NonceIData
ResponderIDPayload = PayloadHeader | RestOfRespIDPayload
RestOfRespIDPayload = IDType | RESERVED | RespIDData
MACedIDForR = prf(SK_pr, RestOfRespIDPayload)
```

In particular, the initiator, but not the responder, authenticates the IKE_SA_INIT request (RealMessage1) and the responder, but not the initiator, authenticates the IKE_SA_INIT response (RealMessage2). Thus, each side authenticates only the initial message it has sent and not the initial message it has received.

4. Downgrade Attacks Description

The way authentication is performed in IKEv2 allows at least two kinds of downgrade attacks. The first of these is a key-compromise impersonation (KCI) attack and requires a set of preconditions that are not common, but still plausible. In particular:

1. The attacker must be on the path with the ability to intercept communications between the peers and to modify their messages.

2. Security policies for both initiator and responder must include both "strong" and "weak" key exchange methods and the attacker must be able to break "weak" key exchange methods in real time.

3. The attacker must either have a long-term authentication key for one of the peers or must be able to break the authentication algorithm used by one of the peers in real time.

Having these preconditions the goal of the attacker is to eavesdrop on communication between the peers. While the attack requires impersonating one of these peers to the other, impersonation is not its primary goal.

In case the attacker knows the initiator’s long-term authentication key, the attack can be mounted as follows.

1. The initiator sends the IKE_SA_INIT request message with a list of proposed algorithms that includes both "weak" and "strong" key exchange methods.

2. The attacker intercepts this message and re-injects a modified message without "strong" key exchange methods. Note that this may require an additional step for the attack to succeed if the initiator includes a public key for a "strong" key exchange method in the request. In this case the attacker intercepts this message and responds with the INVALID_KE_PAYLOAD notification indicating that the initiator must include a public key for a "weak" key exchange method. Then this message is intercepted and re-injected without "strong" key exchange methods.

3. The responder receives this message and selects one of the "weak" key exchange methods (since the message does not include any "strong" ones), then it sends back a response message, which the attacker allows to pass through without modifications.

4. Since the attacker has seen both public keys and can break the selected "weak" key exchange method in real time, it calculates the SK_* session keys that allow the attacker to read and modify the content of the encrypted IKE messages.

5. The initiator receives the IKE_SA_INIT response message, accepts the responder’s selected algorithms, including the "weak" key exchange method (since it is allowed by its policy), and starts the IKE_AUTH exchange. It computes the AUTH payload, thus authenticating the IKE_SA_INIT request message it has sent.

6. The attacker intercepts this message, decrypts it and modifies the AUTH payload so that it allegedly authenticates the IKE_SA_INIT request message that was modified and injected by the attacker. The attacker is able to do this because it knows the session keys and the initiator's long-term authentication key.

7. The responder receives this message, verifies the AUTH payload and sends back the IKE_AUTH response message, which the attacker allows to pass through.

8. At this point the peers have established a connection using the "weak" key exchange method. Note, that this is allowed by their security policies, but without the attacker's intervention they would have used a more secure "strong" key exchange method. The attacker essentially forced the peers to use a "weak" method that it is able to break, thus downgrading the security properties of the connection so that it can read the peers' communication.

A variant of this attack can be mounted if the attacker has the responder' s long-term authentication key. In this case the attacker cannot change the set of algorithms from which the responder makes its choice, but still may be able to force peers not to use some protocol extensions, in particular those that are initially proposed by the responder.

The second type of attack is an identity misbinding attack described in [DOWNGRADE]. The attacker's goal is once again to eavesdrop on the communication between two peers, but unlike the KCI attack, it does not need to compromise one of the peers. Instead, the attacker only needs to know the long-term authentication key of some party with whom one of the peers is configured to communicate.

In particular, suppose the attacker wants to eavesdrop on communication between initiator I and responder R and has access to the long-term authentication key of a different initiator A. The attack works exactly the same way as the previous one, with one exception: after decrypting and modifying I's AUTH payload, it authenticates the modified AUTH payload with A's long-term authentication key instead of I's. At the end of the attack, initiator I will believe it has established a connection with responder R, but responder R will believe it has established a connection with initiator A (whose authentication key is known to the attacker). Nevertheless, the attacker will be able to read the encrypted messages sent between I and R.

This attack used to be less relevant when cryptographic algorithms were considered secure or insecure because peers would disable the insecure ones according to their security policy and not negotiate

them. On the other hand, the coexistence of old and new algorithms in the post-quantum (or any other) migration makes this attack more relevant. With migration to quantum-resistant algorithms the KCI or identity misbinding attacks could be mounted on a hybrid PQ/T ([RFC9370]) or pure post-quantum key exchange; where an attacker able to break a traditional key exchange method (e.g. by means of a quantum computer) prevents peers from executing quantum-resistant key exchange method(s).

5. Downgrade Attacks Prevention

This document defines an IKEv2 extension that detects attempts to mount the downgrade attacks described in Section 4. If both peers support this extension and if at least one non-compromised authentication key is used by the peers in the protocol run then:

* An attacker cannot fool any protocol participant that its peer does not support this extension without being detected.

* An attacker cannot modify the IKE_SA_INIT messages without being detected.

If this extension is not supported by both peers, then the IKEv2 negotiation runs as defined in [RFC7296].

The idea is that both the IKE_SA_INIT request and the IKE_SA_INIT response messages must be directly authenticated by both peers. Thus, if at least one non-compromised key is used in the IKE SA establishing, then any modification of the IKE_SA_INIT messages will be detected. In particular, for the attacks described in Section 4, it is necessary that the attacker forward the responder's signed octets. Since the attacker is presumed to be incapable of forging a valid signature on behalf of the responder, an attempt by the attacker to remove the responder's commitment to this extension would invalidate the signature in the AUTH payload. Consequently, while an attacker could strip support for this extension from the initiator, it could not do so from the responder.

In essence, the peers use this extension to confirm they have had the same conversation, a property enjoyed by many modern authenticated key exchange protocols that may have other benefits beyond downgrade protection, like TLS 1.3 [RFC8446].

6. Protocol Details

The initiator supporting this extension includes a new status type notification IKE_SA_INIT_FULL_TRANSCRIPT_AUTH in the IKE_SA_INIT request message. The Notify Message Type for this notification is <TBA1 by IANA>, Protocol ID and SPI Size are both set to 0 and the notification data is empty.

If the responder supports this extension then it also includes this notification in the response message regardless of whether it was received in the request or not.

```
Initiator                          Responder
-------------------------------------------------------------------
HDR, SAi1, KEi, Ni,
N(IKE_SA_INIT_FULL_TRANSCRIPT_AUTH)  --->
                                <---  HDR, SAr1, KEr, Nr, [CERTREQ,]
                                 N(IKE_SA_INIT_FULL_TRANSCRIPT_AUTH)
```

If a peer sent and received the IKE_SA_INIT_FULL_TRANSCRIPT_AUTH notification, then it uses the modified construction of the blocks of data to be signed (or MAC'ed) compared to the definition from Section 2.15 of [RFC7296]:

```
InitiatorSignedOctets = ZeroPrefix | RealMessage2
                        | RealMessage1 | NonceRData | MACedIDForI

ResponderSignedOctets = ZeroPrefix | RealMessage1
                        | RealMessage2 | NonceIData | MACedIDForR
```

where RealMessage1, RealMessage2, NonceIData, NonceRData, MACedIDForI and MACedIDForR are defined in Section 2.15 of [RFC7296], and ZeroPrefix is 8 octets of zero. ZeroPrefix serves a role of a domain separator making the new authentication blocks of data always different from authentication blocks of data defined in [RFC7296], because in both RealMessage1 and RealMessage2 the first 8 octets constitute IKE Initiator's SPI that can never be zero.

7. Interaction with other IKEv2 Extensions

7.1. Interaction with the IKE_INTERMEDIATE Exchange

The IKE_INTERMEDIATE exchange defined in [RFC9242] also modifies blocks of data to be signed (or MAC'ed). This modification is described in Section 3.3.2 of [RFC9242] and can be summarized as an addition of a new piece of data (IntAuth) to the end of the blocks of data from Section 2.15 of [RFC7296]. If peers support extension defined in this document, then they MUST treat modified blocks of

data to be signed (or MAC'ed) defined in Section 6 as replacements for blocks of data defined in Section 2.15 of [RFC7296], so that in case of IKE_INTERMEDIATE the IntAuth is added to these modified blocks.

> | Authentication of the IKE_INTERMEDIATE exchange includes
> | messages sent in both directions, thus the attacker cannot
> | change its messages without being detected.

7.2. Interaction with the IKE Session Resumption

IKE Session Resumption [RFC5723] allows peers to quickly restore IKE SA upon a failure. To be able to do it a security gateway provides a client with session ticket that allows the gateway to restore the IKE SA if this ticket is later presented by the client. [RFC5723] contains the list of IKE SA parameters marking each of parameter as either "restored from the ticket" or "re-negotiated at the time of resumption".

The information of whether an implementation used the new authentication logic for old SA MUST be stored in the ticket and the implementation MUST act the same way when doing resumption. This means that in the IKE_SESSION_RESUME exchange peers do not send the IKE_SA_INIT_AUTH notification and do not expect it in the received messages.

If there is a chance that the state of this feature can be changed during SA inactivity (e.g., a host is upgraded or its configuration was changed), it is RECOMMENDED that the host do not use IKE SA resumption and does a full handshake instead.

8. Security Considerations

The IKEv2 extension defined in this document protects against downgrade attacks on IKEv2 described in Section 4. It only provides this protection when both peers implement the extension.

The attacks described in this document can also be mitigated by disabling support for weak key exchange methods. Doing so is feasible when the peer is known out-of-band to support strong key exchange methods, but this information may not be available in all deployment scenarios for IKEv2.

The attacks can also be mitigated by mixing a pre-shared key into the session key calculation. An attacker that does not know this pre-shared key will be unable to decrypt even if it manages to downgrade the key exchange. However, the use of a pre-shared key is negotiated by an extension [RFC8784], [RFC9867], and this negotiation is itself

subject to downgrade attack. It is therefore necessary for each of the peers to mandate the use of a pre-shared key and abort the connection if negotiation fails.

9. IANA Considerations

This document defines new Notify Message Type in the "IKEv2 Notify Message Status Types" registry:

<TBA> IKE_SA_INIT_FULL_TRANSCRIPT_AUTH

10. Acknowledgements

TODO

11. References

11.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC7296] Kaufman, C., Hoffman, P., Nir, Y., Eronen, P., and T. Kivinen, "Internet Key Exchange Protocol Version 2 (IKEv2)", STD 79, RFC 7296, DOI 10.17487/RFC7296, October 2014, <https://www.rfc-editor.org/info/rfc7296>.

[RFC9242] Smyslov, V., "Intermediate Exchange in the Internet Key Exchange Protocol Version 2 (IKEv2)", RFC 9242, DOI 10.17487/RFC9242, May 2022, <https://www.rfc-editor.org/info/rfc9242>.

[RFC5723] Sheffer, Y. and H. Tschofenig, "Internet Key Exchange Protocol Version 2 (IKEv2) Session Resumption", RFC 5723, DOI 10.17487/RFC5723, January 2010, <https://www.rfc-editor.org/info/rfc5723>.

11.2. Informative References

[RFC9370] Tjhai, CJ., Tomlinson, M., Bartlett, G., Fluhrer, S., Van Geest, D., Garcia-Morchon, O., and V. Smyslov, "Multiple Key Exchanges in the Internet Key Exchange Protocol Version 2 (IKEv2)", RFC 9370, DOI 10.17487/RFC9370, May 2023, <https://www.rfc-editor.org/info/rfc9370>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

[RFC8784] Fluhrer, S., Kampanakis, P., McGrew, D., and V. Smyslov, "Mixing Preshared Keys in the Internet Key Exchange Protocol Version 2 (IKEv2) for Post-quantum Security", RFC 8784, DOI 10.17487/RFC8784, June 2020, <https://www.rfc-editor.org/info/rfc8784>.

[RFC9867] Smyslov, V., "Mixing Preshared Keys in the IKE_INTERMEDIATE and CREATE_CHILD_SA Exchanges of the Internet Key Exchange Protocol Version 2 (IKEv2) for Post-Quantum Security", RFC 9867, DOI 10.17487/RFC9867, November 2025, <https://www.rfc-editor.org/info/rfc9867>.

[SIGMA] Krawczyk, H., "SIGMA: The ‘SIGn-and-MAc’ Approach to Authenticated Diffie-Hellman and Its Use in the IKE Protocols", Springer Berlin Heidelberg, Lecture Notes in Computer Science pp. 400-425, DOI 10.1007/978-3-540-45146-4_24, ISBN ["9783540406747", "9783540451464"], 2003, <https://doi.org/10.1007/978-3-540-45146-4_24>.

[DOWNGRADE] Bhargavan, K., Brzuska, C., Fournet, C., Kohlweiss, M., Zanella-Bguelin, S., and M. Green, "Downgrade Resilience in Key-Exchange Protocols", Cryptology ePrint Archive Paper 2016/072, January 2016, <https://ia.cr/2016/072>.

Authors’ Addresses

Valery Smyslov
ELVIS-PLUS
Russian Federation
Email: svan@elvis.ru

Christopher Patton
Cloudflare
United States of America

Email: chrispatton+ietf@gmail.com