IPsec

In computing, **Internet Protocol Security** (**IPsec**) is a secure network protocol suite that authenticates and encrypts the packets of data to provide secure encrypted communication between two computers over an Internet Protocol network. It is used in virtual private networks (VPNs).

IPsec includes protocols for establishing mutual authentication between agents at the beginning of a session and negotiation of cryptographic keys to use during the session. IPsec can protect data flows between a pair of hosts (*host-to-host*), between a pair of security gateways (*network-to-network*), or between a security gateway and a host (*network-to-host*).[1] IPsec uses cryptographic security services to protect communications over Internet Protocol (IP) networks. It supports network-level peer authentication, data origin authentication, data integrity, data confidentiality (encryption), and replay protection.

The initial IPv4 suite was developed with few security provisions. As a part of the IPv4 enhancement, IPsec is a layer 3 OSI model or internet layer end-to-end security scheme. In contrast, while some other Internet security systems in widespread use operate above layer 3, such as Transport Layer Security (TLS) that operates at the Transport Layer and Secure Shell (SSH) that operates at the Application layer, IPsec can automatically secure applications at the IP layer.

Contents

History

Starting in the early 1970s, the Advanced Research Projects Agency sponsored a series of experimental ARPANET encryption devices, at first for native ARPANET packet encryption and subsequently for TCP/IP packet encryption; some of these were certified and fielded. From 1986 to 1991, the NSA sponsored the development of security protocols for the Internet under its Secure Data Network Systems (SDNS) program.[2] This brought together various vendors including Motorola who produced a network encryption device in 1988. The work was openly published from about 1988 by NIST and, of these, *Security Protocol at Layer 3* (SP3) would eventually morph into the ISO standard Network Layer Security Protocol (NLSP).[3]

From 1992 to 1995, various groups conducted research into IP-layer encryption.

- 1. In 1992, the US Naval Research Laboratory (NRL) began the Simple Internet Protocol Plus (SIPP) project to research and implement IP encryption.
- 2. In 1993, at Columbia University and AT&T Bell Labs, John Ioannidis and others researched the software experimental Software IP Encryption Protocol (swIPe) on SunOS.
- 3. In 1993, Sponsored by Whitehouse internet service project, Wei Xu at Trusted Information Systems (TIS) further researched the Software IP Security Protocols and developed the hardware support for the triple DES Data Encryption Standard,[4] which was coded in the BSD 4.1 kernel and supported both x86 and SUNOS architectures. By December 1994, TIS released their DARPA-sponsored open-source Gauntlet Firewall product with the integrated 3DES hardware encryption at over T1 speeds. It was the first-time using IPSec VPN connections between the east and west coast of the States, known as the first commercial IPSec VPN product.
- 4. Under NRL's DARPA-funded research effort, NRL developed the IETF standards-track specifications (RFC 1825 through RFC 1827) for IPsec, which was coded in the BSD 4.4 kernel and supported both x86 and SPARC CPU architectures.[5] NRL's IPsec implementation was described in their paper in the 1996 USENIX Conference Proceedings.[6] NRL's open-source IPsec implementation was made available online by MIT and became the basis for most initial commercial implementations.[7]

The Internet Engineering Task Force (IETF) formed the IP Security Working Group in 1992[8] to standardize openly specified security extensions to IP, called *IPsec*.[9] In 1995, the working group organized a few of the workshops with members from the five companies (TIS, CISCO, FTP, Checkpoint, etc.). During the IPSec workshops, the NRL's standards and Cisco and TIS' software are standardized as the public references, published as RFC-1825 through RFC-1827.[10]

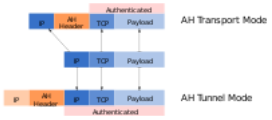
Security architecture

The IPsec is an open standard as a part of the IPv4 suite. IPsec uses the following protocols to perform various functions:[11][12]

- Authentication Headers (AH) provides connectionless data integrity and data origin authentication for IP datagrams and provides protection against replay attacks.[13][14]
- Encapsulating Security Payloads (ESP) provides confidentiality, connectionless data integrity, data origin authentication, an anti-replay service (a form of partial sequence integrity), and limited traffic-flow confidentiality.[1]
- Internet Security Association and Key Management Protocol (ISAKMP) provides a framework for authentication and key exchange,[15] with actual authenticated keying material provided either by manual configuration with pre-shared keys, Internet Key Exchange (IKE and IKEv2), Kerberized Internet Negotiation of Keys (KINK), or IPSECKEY DNS records.[16][17][18][19] The purpose is to generate the Security Associations (SA) with the bundle of algorithms and parameters necessary for AH and/or ESP operations.

Authentication Header

The Security Authentication Header (AH) was developed at the US Naval Research Laboratory in the early 1990s and is derived in part from previous IETF standards' work for authentication of the Simple Network Management Protocol (SNMP) version 2. Authentication Header (AH) is a member of the IPsec protocol suite. AH ensures connectionless integrity by using a hash function and a secret shared key in the AH algorithm. AH also guarantees the data origin by authenticating IP packets. Optionally a sequence number can protect the IPsec packet's contents against replay attacks,[20] using the sliding window technique and discarding old packets.



Usage of IPsec Authentication Header format in Tunnel and Transport modes

- In IPv4, AH prevents option-insertion attacks. In IPv6, AH protects both against header insertion attacks and option insertion attacks.
- In IPv4, the AH protects the IP payload and all header fields of an IP datagram except for mutable fields (i.e. those that might be altered in transit), and also IP options such as the IP Security Option (RFC 1108). Mutable (and therefore unauthenticated) IPv4 header fields are DSCP/ToS, ECN, Flags, Fragment Offset, TTL and Header Checksum.[14]
- In IPv6, the AH protects most of the IPv6 base header, AH itself, non-mutable extension headers after the AH, and the IP payload. Protection for the IPv6 header excludes the mutable fields: DSCP, ECN, Flow Label, and Hop Limit.[14]

AH operates directly on top of IP, using IP protocol number 51.[21]

The following AH packet diagram shows how an AH packet is constructed and interpreted:[13][14]

Authentication Header format

Offsets	Octet$_{16}$	0								1								2								3							
Octet$_{16}$	Bit$_{10}$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	*Next Header*								*Payload Len*								*Reserved*															
4	32	*Security Parameters Index (SPI)*																															
8	64	*Sequence Number*																															
C	96	*Integrity Check Value (ICV)* …																															
…	…																																

Next Header **(8 bits)**

Type of the next header, indicating what upper-layer protocol was protected. The value is taken from the list of IP protocol numbers.

Payload Len **(8 bits)**

The length of this *Authentication Header* in 4-octet units, minus 2. For example, an AH value of 4 equals 3×(32-bit fixed-length AH fields) + 3×(32-bit ICV fields) − 2 and thus an AH value of 4 means 24 octets. Although the size is measured in 4-octet units, the length of this header needs to be a multiple of 8 octets if carried in an IPv6 packet. This restriction does not apply to an *Authentication Header* carried in an IPv4 packet.

Reserved **(16 bits)**

Reserved for future use (all zeroes until then).

Security Parameters Index **(32 bits)**

Arbitrary value which is used (together with the destination IP address) to identify the security association of the receiving party.

Sequence Number **(32 bits)**

A monotonic strictly increasing sequence number (incremented by 1 for every packet sent) to prevent replay attacks. When replay detection is enabled, sequence numbers are never reused, because a new security association must be renegotiated before an attempt to increment the sequence number beyond its maximum value.[14]

Integrity Check Value **(multiple of 32 bits)**

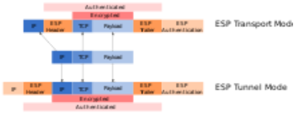
Variable length check value. It may contain padding to align the field to an 8-octet boundary for IPv6, or a 4-octet boundary for IPv4.

Encapsulating Security Payload

The IP Encapsulating Security Payload (ESP)[22] was developed at the Naval Research Laboratory starting in 1992 as part of a DARPA-sponsored research project, and was openly published by IETF SIPP[23] Working Group drafted in December 1993 as a security extension for SIPP. This ESP was originally derived from the US Department of Defense SP3D protocol, rather than being derived from the ISO Network-Layer Security Protocol (NLSP). The SP3D protocol specification was published by NIST in the late 1980s, but designed by the Secure Data Network System project of the US Department of Defense. Encapsulating Security Payload

(ESP) is a member of the IPsec protocol suite. It provides origin authenticity through source authentication, data integrity through hash functions and confidentiality through encryption protection for IP packets. ESP also supports encryption-only and authentication-only configurations, but using encryption without authentication is strongly discouraged because it is insecure.[24][25][26]

Unlike Authentication Header (AH), ESP in transport mode does not provide integrity and authentication for the entire IP packet. However, in Tunnel Mode, where the entire original IP packet is encapsulated with a new packet header added, ESP protection is afforded to the whole inner IP packet (including the inner header) while the outer header (including any outer IPv4 options or IPv6 extension headers) remains unprotected. ESP operates directly on top of IP, using IP protocol number 50.[21]



Usage of IPsec Encapsulating Security Payload (ESP) in Tunnel and Transport modes

The following ESP packet diagram shows how an ESP packet is constructed and interpreted:[1][27]

Encapsulating Security Payload format

Offsets	Octet$_{16}$	0								1								2								3							
Octet$_{16}$	Bit$_{10}$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	*Security Parameters Index (SPI)*																															
4	32	*Sequence Number*																															
8	64	*Payload data*																															
...	...																																
...	...									*Padding (0-255 octets)*																							
...	...																																
...	...																	*Pad Length*								*Next Header*							
...	...	*Integrity Check Value (ICV)* …																															
...	...																																

Security Parameters Index **(32 bits)**
Arbitrary value used (together with the destination IP address) to identify the security association of the receiving party.

Sequence Number **(32 bits)**
A monotonically increasing sequence number (incremented by 1 for every packet sent) to protect against replay attacks. There is a separate counter kept for every security association.

Payload data **(variable)**
The protected contents of the original IP packet, including any data used to protect the contents (e.g. an Initialisation Vector for the cryptographic algorithm). The type of content that was protected is indicated by the *Next Header* field.

Padding **(0-255 octets)**
Padding for encryption, to extend the payload data to a size that fits the encryption's cipher block size, and to align the next field.

Pad Length **(8 bits)**
Size of the padding (in octets).

Next Header **(8 bits)**
Type of the next header. The value is taken from the list of IP protocol numbers.

Integrity Check Value **(multiple of 32 bits)**
Variable length check value. It may contain padding to align the field to an 8-octet boundary for IPv6, or a 4-octet boundary for IPv4.

Security association

The IPsec protocols use a security association, where the communicating parties establish shared security attributes such as algorithms and keys. As such IPsec provides a range of options once it has been determined whether AH or ESP is used. Before exchanging data the two hosts agree on which algorithm is used to encrypt the IP packet, for example DES or IDEA, and which hash function is used to ensure the integrity of the data, such as MD5 or SHA. These parameters are agreed for the particular session, for which a lifetime must be agreed and a session key.[28]

The algorithm for authentication is also agreed before the data transfer takes place and IPsec supports a range of methods. Authentication is possible through pre-shared key, where a symmetric key is already in the possession of both hosts, and the hosts send each other hashes of the shared key to prove that they are in possession of the same key. IPsec also supports public key encryption, where each host has a public and a private key, they exchange their public keys and each host sends the other a nonce encrypted with the other host's public key. Alternatively if both hosts hold a public key certificate from a certificate authority, this can be used for IPsec authentication.[29]

The security associations of IPsec are established using the Internet Security Association and Key Management Protocol (ISAKMP). ISAKMP is implemented by manual configuration with pre-shared secrets, Internet Key Exchange (IKE and IKEv2), Kerberized Internet Negotiation of Keys (KINK), and the use of IPSECKEY DNS records.[19][30][31] RFC 5386 defines Better-Than-Nothing Security (BTNS) as an unauthenticated mode of IPsec using an extended IKE protocol. C. Meadows, C. Cremers, and others have used Formal Methods to identify various anomalies which exist in IKEv1 and also in IKEv2.[32]

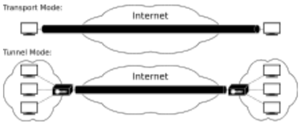
In order to decide what protection is to be provided for an outgoing packet, IPsec uses the Security Parameter Index (SPI), an index to the security association database (SADB), along with the destination address in a packet header, which together uniquely identifies a security association for that packet. A similar procedure is performed for an incoming packet, where IPsec gathers decryption and verification keys from the security association database.

For IP multicast a security association is provided for the group, and is duplicated across all authorized receivers of the group. There may be more than one security association for a group, using different SPIs, thereby allowing multiple levels and sets of security within a group. Indeed, each sender can have multiple security associations, allowing authentication, since a receiver can only know that someone knowing the keys sent the data. Note that the relevant standard does not describe how the association is chosen and duplicated across the group; it is assumed that a responsible party will have made the choice.

Modes of operation

The IPsec protocols AH and ESP can be implemented in a host-to-host transport mode, as well as in a network tunneling mode.

Transport mode



IPsec Modes

In transport mode, only the payload of the IP packet is usually encrypted or authenticated. The routing is intact, since the IP header is neither modified nor encrypted; however, when the authentication header is used, the IP addresses cannot be modified by network address translation, as this always invalidates the hash value. The transport and application layers are always secured by a hash, so they cannot be modified in any way, for example by translating the port numbers.

A means to encapsulate IPsec messages for NAT traversal has been defined by RFC documents describing the NAT-T mechanism.

Tunnel mode

In tunnel mode, the entire IP packet is encrypted and authenticated. It is then encapsulated into a new IP packet with a new IP header. Tunnel mode is used to create virtual private networks for network-to-network communications (e.g. between routers to link sites), host-to-network communications (e.g. remote user access) and host-to-host communications (e.g. private chat).[33]

Tunnel mode supports NAT traversal.

Algorithms

Symmetric encryption algorithms

Cryptographic algorithms defined for use with IPsec include:

- HMAC-SHA1/SHA2 for integrity protection and authenticity.
- TripleDES-CBC for confidentiality
- AES-CBC and AES-CTR for confidentiality.
- AES-GCM and ChaCha20-Poly1305 providing confidentiality and authentication together efficiently.

Refer to RFC 8221 for details.

Key exchange algorithms

- Diffie–Hellman (RFC 3526)
- ECDH (RFC 4753)

Authentication algorithms

- RSA
- ECDSA (RFC 4754)
- PSK (RFC 6617)

Implementations

The IPsec can be implemented in the IP stack of an operating system, which requires modification of the source code. This method of implementation is done for hosts and security gateways. Various IPsec capable IP stacks are available from companies, such as HP or IBM.[34] An alternative is so called bump-in-the-stack (BITS) implementation, where the operating system source code does not have to be modified. Here IPsec is installed between the IP stack and the network drivers. This way operating systems can be retrofitted with IPsec. This method of implementation is also used for both hosts and gateways. However, when retrofitting IPsec the encapsulation of IP packets may cause problems for the automatic path MTU discovery, where the maximum transmission unit (MTU) size on the network path between two IP hosts is established. If a host or gateway has a separate cryptoprocessor, which is common in the military and can also be found in commercial systems, a so-called bump-in-the-wire (BITW) implementation of IPsec is possible.[35]

When IPsec is implemented in the kernel, the key management and ISAKMP/IKE negotiation is carried out from user space. The NRL-developed and openly specified "PF_KEY Key Management API, Version 2" is often used to enable the application-space key management application to update the IPsec Security Associations stored within the kernel-space IPsec implementation.[36] Existing IPsec implementations usually include ESP, AH, and IKE version 2. Existing IPsec implementations on UNIX-like operating systems, for example, Solaris or Linux, usually include PF_KEY version 2.

Embedded IPsec can be used to ensure the secure communication among applications running over constrained resource systems with a small overhead.[37]

Standards status

IPsec was developed in conjunction with IPv6 and was originally required to be supported by all standards-compliant implementations of IPv6 before RFC 6434 made it only a recommendation.[38] IPsec is also optional for IPv4 implementations. IPsec is most commonly used to secure IPv4 traffic.

IPsec protocols were originally defined in RFC 1825 through RFC 1829, which were published in 1995. In 1998, these documents were superseded by RFC 2401 and RFC 2412 with a few incompatible engineering details, although they were conceptually identical. In addition, a mutual authentication and key exchange protocol Internet Key Exchange (IKE) was defined to create and manage security associations. In December 2005, new standards were defined in RFC 4301 and RFC 4309 which are largely a superset of the previous editions with a second version of the Internet Key Exchange standard IKEv2. These third-generation documents standardized the abbreviation of IPsec to uppercase “IP” and lowercase “sec”. “ESP” generally refers to RFC 4303, which is the most recent version of the specification.

Since mid-2008, an IPsec Maintenance and Extensions (ipsecme) working group is active at the IETF.[39][40]

Alleged NSA interference

In 2013, as part of Snowden leaks, it was revealed that the US National Security Agency had been actively working to "Insert vulnerabilities into commercial encryption systems, IT systems, networks, and endpoint communications devices used by targets" as part of the Bullrun program.[41] There are allegations that IPsec was a targeted encryption system.[42]

The OpenBSD IPsec stack came later on and also was widely copied. In a letter which OpenBSD lead developer Theo de Raadt received on 11 Dec 2010 from Gregory Perry, it is alleged that Jason Wright and others, working for the FBI, inserted "a number of backdoors and side channel key leaking mechanisms" into the OpenBSD crypto code. In the forwarded email from 2010, Theo de Raadt did not at first express an official position on the validity of the claims, apart from the implicit endorsement from forwarding the email.[43] Jason Wright's response to the allegations: "Every urban legend is made more real by the inclusion of real names, dates, and times. Gregory Perry's email falls into this category. … I will state clearly that I did not add backdoors to the OpenBSD operating system or the OpenBSD crypto framework (OCF)."[44] Some days later, de Raadt commented that "I believe that NETSEC was probably contracted to write backdoors as alleged. … If those were written, I don't believe they made it into our tree."[45] This was published before the Snowden leaks.

An alternative explanation put forward by the authors of the Logjam attack suggests that the NSA compromised IPsec VPNs by undermining the Diffie-Hellman algorithm used in the key exchange. In their paper[46] they allege the NSA specially built a computing cluster to precompute multiplicative subgroups for specific primes and generators, such as for the second Oakley group defined in RFC 2409. As of May 2015, 90% of addressable IPsec VPNs supported the second Oakley group as part of IKE. If an organization were to precompute this group, they could derive the keys being exchanged and decrypt traffic without inserting any software backdoors.

A second alternative explanation that was put forward was that the Equation Group used zero-day exploits against several manufacturers' VPN equipment which were validated by Kaspersky Lab as being tied to the Equation Group[47] and validated by those manufacturers as being real exploits, some of which were zero-day exploits at the time of their exposure.[48][49][50] The Cisco PIX and ASA firewalls had vulnerabilities that were used for wiretapping by the NSA.

Furthermore, IPsec VPNs using "Aggressive Mode" settings send a hash of the PSK in the clear. This can be and apparently is targeted by the NSA using offline dictionary attacks.[51][52][53]

IETF documentation

Standards track

- RFC 1829: The ESP DES-CBC Transform
- RFC 2403: The Use of HMAC-MD5-96 within ESP and AH
- RFC 2404: The Use of HMAC-SHA-1-96 within ESP and AH
- RFC 2405: The ESP DES-CBC Cipher Algorithm With Explicit IV
- RFC 2410: The NULL Encryption Algorithm and Its Use With IPsec
- RFC 2451: The ESP CBC-Mode Cipher Algorithms
- RFC 2857: The Use of HMAC-RIPEMD-160-96 within ESP and AH
- RFC 3526: More Modular Exponential (MODP) Diffie-Hellman groups for Internet Key Exchange (IKE)
- RFC 3602: The AES-CBC Cipher Algorithm and Its Use with IPsec
- RFC 3686: Using Advanced Encryption Standard (AES) Counter Mode With IPsec Encapsulating Security Payload (ESP)
- RFC 3947: Negotiation of NAT-Traversal in the IKE
- RFC 3948: UDP Encapsulation of IPsec ESP Packets
- RFC 4106: The Use of Galois/Counter Mode (GCM) in IPsec Encapsulating Security Payload (ESP)
- RFC 4301: Security Architecture for the Internet Protocol
- RFC 4302: IP Authentication Header
- RFC 4303: IP Encapsulating Security Payload
- RFC 4304: Extended Sequence Number (ESN) Addendum to IPsec Domain of Interpretation (DOI) for Internet Security Association and Key Management Protocol (ISAKMP)
- RFC 4307: Cryptographic Algorithms for Use in the Internet Key Exchange Version 2 (IKEv2)
- RFC 4308: Cryptographic Suites for IPsec
- RFC 4309: Using Advanced Encryption Standard (AES) CCM Mode with IPsec Encapsulating Security Payload (ESP)
- RFC 4543: The Use of Galois Message Authentication Code (GMAC) in IPsec ESP and AH
- RFC 4555: IKEv2 Mobility and Multihoming Protocol (MOBIKE)
- RFC 4806: Online Certificate Status Protocol (OCSP) Extensions to IKEv2
- RFC 4868: Using HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 with IPsec
- RFC 4945: The Internet IP Security PKI Profile of IKEv1/ISAKMP, IKEv2, and PKIX
- RFC 5280: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile
- RFC 5282: Using Authenticated Encryption Algorithms with the Encrypted Payload of the Internet Key Exchange version 2 (IKEv2) Protocol
- RFC 5386: Better-Than-Nothing Security: An Unauthenticated Mode of IPsec
- RFC 5529: Modes of Operation for Camellia for Use with IPsec
- RFC 5685: Redirect Mechanism for the Internet Key Exchange Protocol Version 2 (IKEv2)
- RFC 5723: Internet Key Exchange Protocol Version 2 (IKEv2) Session Resumption
- RFC 5857: IKEv2 Extensions to Support Robust Header Compression over IPsec
- RFC 5858: IPsec Extensions to Support Robust Header Compression over IPsec
- RFC 7296: Internet Key Exchange Protocol Version 2 (IKEv2)
- RFC 7321: Cryptographic Algorithm Implementation Requirements and Usage Guidance for Encapsulating Security Payload (ESP) and Authentication Header (AH)
- RFC 7383: Internet Key Exchange Protocol Version 2 (IKEv2) Message Fragmentation
- RFC 7427: Signature Authentication in the Internet Key Exchange Version 2 (IKEv2)
- RFC 7634: ChaCha20, Poly1305, and Their Use in the Internet Key Exchange Protocol (IKE) and IPsec

Experimental RFCs

- RFC 4478: Repeated Authentication in Internet Key Exchange (IKEv2) Protocol

Informational RFCs

- RFC 2367: PF_KEY Interface
- RFC 2412: The OAKLEY Key Determination Protocol
- RFC 3706: A Traffic-Based Method of Detecting Dead Internet Key Exchange (IKE) Peers
- RFC 3715: IPsec-Network Address Translation (NAT) Compatibility Requirements
- RFC 4621: Design of the IKEv2 Mobility and Multihoming (MOBIKE) Protocol
- RFC 4809: Requirements for an IPsec Certificate Management Profile
- RFC 5387: Problem and Applicability Statement for Better-Than-Nothing Security (BTNS)
- RFC 5856: Integration of Robust Header Compression over IPsec Security Associations
- RFC 5930: Using Advanced Encryption Standard Counter Mode (AES-CTR) with the Internet Key Exchange version 02 (IKEv2) Protocol
- RFC 6027: IPsec Cluster Problem Statement
- RFC 6071: IPsec and IKE Document Roadmap
- RFC 6379: Suite B Cryptographic Suites for IPsec
- RFC 6380: Suite B Profile for Internet Protocol Security (IPsec)
- RFC 6467: Secure Password Framework for Internet Key Exchange Version 2 (IKEv2)

Best current practice RFCs

- RFC 5406: Guidelines for Specifying the Use of IPsec Version 2

Obsolete/historic RFCs

- RFC 1825: Security Architecture for the Internet Protocol (obsoleted by RFC 2401)
- RFC 1826: IP Authentication Header (obsoleted by RFC 2402)
- RFC 1827: IP Encapsulating Security Payload (ESP) (obsoleted by RFC 2406)
- RFC 1828: IP Authentication using Keyed MD5 (historic)
- RFC 2401: Security Architecture for the Internet Protocol (IPsec overview) (obsoleted by RFC 4301)
- RFC 2406: IP Encapsulating Security Payload (ESP) (obsoleted by RFC 4303 and RFC 4305)
- RFC 2407: The Internet IP Security Domain of Interpretation for ISAKMP (obsoleted by RFC 4306)
- RFC 2409: The Internet Key Exchange (obsoleted by RFC 4306)
- RFC 4305: Cryptographic Algorithm Implementation Requirements for Encapsulating Security Payload (ESP) and Authentication Header (AH) (obsoleted by RFC 4835)
- RFC 4306: Internet Key Exchange (IKEv2) Protocol (obsoleted by RFC 5996)
- RFC 4718: IKEv2 Clarifications and Implementation Guidelines (obsoleted by RFC 7296)
- RFC 4835: Cryptographic Algorithm Implementation Requirements for Encapsulating Security Payload (ESP) and Authentication Header (AH) (obsoleted by RFC 7321)
- RFC 5996: Internet Key Exchange Protocol Version 2 (IKEv2) (obsoleted by RFC 7296)

See also

- Dynamic Multipoint Virtual Private Network
- Information security
- NAT traversal
- Opportunistic encryption
- tcpcrypt

References

1. Kent, S.; Atkinson, R. (November 1998). *IP Encapsulating Security Payload (ESP)* (https://tools.ietf.org/html/rfc2406). IETF. doi:10.17487/RFC2406 (https://doi.org/10.17487%2FRFC2406). RFC 2406 (https://tools.ietf.org/html/rfc2406).
2. "Implementation of IPSec Protocol - IEEE Conference Publication". doi:10.1109/ACCT.2012.64 (https://doi.org/10.1109%2FACCT.2012.64). S2CID 16526652 (https://api.semanticscholar.org/CorpusID:16526652).
3. "Archived copy" (https://web.archive.org/web/20140903145752/http://www.toad.com/gnu/netcrypt.html). Archived from the original (http://www.toad.com/gnu/netcrypt.html) on 2014-09-03. Retrieved 2014-02-18.
4. "The history of VPN creation" (https://www.le-vpn.com/history-of-vpn/).
5. "http://web.mit.edu/network/isakmp/"
6. "https://www.usenix.org/legacy/publications/library/proceedings/sd96/atkinson.html"
7. "http://web.mit.edu/network/isakmp/"
8. "IETF IP Security Protocol (ipsec) Working group History" (https://datatracker.ietf.org/wg/ipsec/history/).
9. "RFC4301: Security Architecture for the Internet Protocol" (http://tools.ietf.org/html/rfc4301#page-4). Network Working Group of the IETF. December 2005. p. 4. "The spelling "IPsec" is preferred and used throughout this and all related IPsec standards. All other capitalizations of IPsec [...] are deprecated."
10. "NRL ITD Accomplishments - IPSec and IPv6" (https://www.nrl.navy.mil/itd/sites/www.nrl.navy.mil.itd/files/files/itd_accomp_ipsec.pdf) (PDF). *US Naval Research Laboratories*.
11. Thayer, R.; Doraswamy, N.; Glenn, R. (November 1998). *IP Security Document Roadmap* (https://tools.ietf.org/html/rfc2411). IETF. doi:10.17487/RFC2411 (https://doi.org/10.17487%2FRFC2411). RFC 2411 (https://tools.ietf.org/html/rfc2411).
12. Hoffman, P. (December 2005). *Cryptographic Suites for IPsec* (https://tools.ietf.org/html/rfc4308). IETF. doi:10.17487/RFC4308 (https://doi.org/10.17487%2FRFC4308). RFC 4308 (https://tools.ietf.org/html/rfc4308).

13. Kent, S.; Atkinson, R. (November 1998). *IP Authentication Header* (https://tools.ietf.org/html/rfc2402). IETF. doi:10.17487/RFC2402 (https://doi.org/10.17487%2FRFC2402). RFC 2402 (https://tools.ietf.org/html/rfc2402).
14. Kent, S. (December 2005). *IP Authentication Header* (https://tools.ietf.org/html/rfc4302). IETF. doi:10.17487/RFC4302 (https://doi.org/10.17487%2FRFC4302). RFC 4302 (https://tools.ietf.org/html/rfc4302).
15. The Internet Key Exchange (IKE), RFC 2409, §1 Abstract
16. Harkins, D.; Carrel, D. (November 1998). *The Internet Key Exchange (IKE)* (https://tools.ietf.org/html/rfc2409). IETF. doi:10.17487/RFC2409 (https://doi.org/10.17487%2FRFC2409). RFC 2409 (https://tools.ietf.org/html/rfc2409).
17. Kaufman, C. (ed.). *IKE Version 2* (https://tools.ietf.org/html/rfc4306). IETF. doi:10.17487/RFC4306 (https://doi.org/10.17487%2FRFC4306). RFC 4306 (https://tools.ietf.org/html/rfc4306).
18. Sakane, S.; Kamada, K.; Thomas, M.; Vilhuber, J. (November 1998). *Kerberized Internet Negotiation of Keys (KINK)* (https://tools.ietf.org/html/rfc4430). IETF. doi:10.17487/RFC4430 (https://doi.org/10.17487%2FRFC4430). RFC 4430 (https://tools.ietf.org/html/rfc4430).
19. Richardson, M. (February 2005). *A Method for Storing IPsec Keying Material in DNS* (https://tools.ietf.org/html/rfc4025). IETF. doi:10.17487/RFC4025 (https://doi.org/10.17487%2FRFC4025). RFC 4025 (https://tools.ietf.org/html/rfc4025).
20. Peter Willis (2001). *Carrier-Scale IP Networks: Designing and Operating Internet Networks*. IET. p. 270. ISBN 9780852969823.
21. "Protocol Numbers" (https://web.archive.org/web/20100529122930/https://www.iana.org/assignments/protocol-numbers/protocol-numbers.xml). *IANA*. IANA. 2010-05-27. Archived from the original (https://www.iana.org/assignments/protocol-numbers/protocol-numbers.xml) on 2010-05-29.
22. "SIPP Encapsulating Security Payload" (https://web.archive.org/web/20160909031941/http://www.toad.com/gnu/draft-ietf-sip-esp-00.txt). IETF SIPP Working Group. 1993. Archived from the original (http://www.toad.com/gnu/draft-ietf-sip-esp-00.txt) on 2016-09-09. Retrieved 2013-08-07.
23. "Draft SIPP Specification" (http://tools.ietf.org/html/draft-ietf-sipp-spec-00). IETF. 1993. p. 21.
24. Bellovin, Steven M. (1996). "Problem Areas for the IP Security Protocols" (https://www.cs.columbia.edu/~smb/papers/badesp.ps) (PostScript). *Proceedings of the Sixth Usenix Unix Security Symposium*. San Jose, CA. pp. 1–16. Retrieved 2007-07-09.
25. Paterson, Kenneth G.; Yau, Arnold K.L. (2006-04-24). "Cryptography in theory and practice: The case of encryption in IPsec" (http://eprint.iacr.org/2005/416) (PDF). *Eurocrypt 2006, Lecture Notes in Computer Science Vol. 4004*. Berlin. pp. 12–29. Retrieved 2007-08-13.
26. Degabriele, Jean Paul; Paterson, Kenneth G. (2007-08-09). "Attacking the IPsec Standards in Encryption-only Configurations" (http://eprint.iacr.org/2007/125) (PDF). *IEEE Symposium on Security and Privacy, IEEE Computer Society*. Oakland, CA. pp. 335–349. Retrieved 2007-08-13.
27. Kent, S. (December 2005). *IP Encapsulating Security Payload (ESP)* (https://tools.ietf.org/html/rfc4303). IETF. doi:10.17487/RFC4303 (https://doi.org/10.17487%2FRFC4303). RFC 4303 (https://tools.ietf.org/html/rfc4303).
28. Peter Willis (2001). *Carrier-Scale IP Networks: Designing and Operating Internet Networks*. IET. p. 271. ISBN 9780852969823.
29. Peter Willis (2001). *Carrier-Scale IP Networks: Designing and Operating Internet Networks*. IET. pp. 272–3. ISBN 9780852969823.
30. RFC 2406, §1, page 2
31. Thomas, M. (June 2001). *Requirements for Kerberized Internet Negotiation of Keys* (https://tools.ietf.org/html/rfc3129). doi:10.17487/RFC3129 (https://doi.org/10.17487%2FRFC3129). RFC 3129 (https://tools.ietf.org/html/rfc3129).
32. C. Cremers, Key Exchange in IPsec Revisited: Formal Analysis of IKEv1 and IKEv2, ESORICS 2011, published by Springer: "https://link.springer.com/chapter/10.1007/978-3-642-23822-2_18"
33. William, S., & Stallings, W. (2006). Cryptography and Network Security, 4/E. Pearson Education India. p. 492-493
34. Peter Willis (2001). *Carrier-Scale IP Networks: Designing and Operating Internet Networks*. IET. p. 266. ISBN 9780852969823.
35. Peter Willis (2001). *Carrier-Scale IP Networks: Designing and Operating Internet Networks*. IET. p. 267. ISBN 9780852969823.
36. RFC 2367, *PF_KEYv2 Key Management API*, Dan McDonald, Bao Phan, & Craig Metz (July 1998)
37. Hamad, Mohammad; Prevelakis, Vassilis (2015). *Implementation and performance evaluation of embedded IPsec in microkernel OS* (https://publikationsserver.tu-braunschweig.de/receive/dbbs_mods_00065815). *2015 World Symposium on Computer Networks and Information Security (WSCNIS)*. IEEE. doi:10.1109/wscnis.2015.7368294 (https://doi.org/10.1109%2Fwscnis.2015.7368294). ISBN 9781479999064. S2CID 16935000 (https://api.semanticscholar.org/CorpusID:16935000).
38. RFC 6434, "IPv6 Node Requirements", E. Jankiewicz, J. Loughney, T. Narten (December 2011)
39. "ipsecme charter" (https://datatracker.ietf.org/wg/ipsecme/charter/). Retrieved 2015-10-26.
40. "ipsecme status" (https://tools.ietf.org/wg/ipsecme/). Retrieved 2015-10-26.
41. "Secret Documents Reveal N.S.A. Campaign Against Encryption" (https://www.nytimes.com/interactive/2013/09/05/us/documents-reveal-nsa-campaign-against-encryption.html). *New York Times*.
42. John Gilmore. "Re: [Cryptography] Opening Discussion: Speculation on "BULLRUN" " (http://www.mail-archive.com/cryptography@metzdowd.com/msg12325.html).
43. Theo de Raadt. "Allegations regarding OpenBSD IPSEC" (https://marc.info/?l=openbsd-tech&m=129236621626462&w=2).
44. Jason Wright. "Allegations regarding OpenBSD IPSEC" (https://marc.info/?l=openbsd-tech&m=129244045916861&w=2).
45. Theo de Raadt. "Update on the OpenBSD IPSEC backdoor allegation" (https://lwn.net/Articles/420858/).
46. David Adrian; Karthikeyan Bhargavan; Zakir Durumeric; Pierrick Gaudry; Matthew Green; J. Alex Halderman; Nadia Heninger; Drew Springall; Emmanuel Thomé; Luke Valenta; Benjamin VanderSloot; Eric Wustrow; Santiago Zanella-Béguelink; Paul Zimmermann. "Imperfect Forward Secrecy: How Diffie-Hellman Fails in Practice" (https://weakdh.org/imperfect-forward-secrecy.pdf) (PDF).
47. Goodin, Dan (August 16, 2016). "Confirmed: hacking tool leak came from "omnipotent" NSA-tied group" (https://arstechnica.com/security/2016/08/code-dumped-online-came-from-omnipotent-nsa-tied-hacking-group/). *Ars Technica*. Retrieved August 19, 2016.
48. Thomson, Iain (August 17, 2016). "Cisco confirms two of the Shadow Brokers' 'NSA' vulns are real" (https://www.theregister.co.uk/2016/08/17/cisco_two_shadow_brokers_vulnerabilities_real/). *The Register*. Retrieved September 16, 2016.
49. Pauli, Darren (August 24, 2016). "Equation Group exploit hits newer Cisco ASA, Juniper Netscreen" (https://www.theregister.co.uk/2016/08/24/equation_group_exploit_expanded_to_target_cisco_924_asa_boxes/). *The Register*. Retrieved September 16, 2016.
50. Chirgwin, Richard (August 18, 2016). "Fortinet follows Cisco in confirming Shadow Broker vuln" (https://www.theregister.co.uk/2016/08/18/fortinet_follows_cisco_in_confirming_shadow_broker_vuln/). *The Register*. Retrieved September 16, 2016.
51. https://weakdh.org/imperfect-forward-secrecy-ccs15.pdf
52. What are the problems of IKEv1 aggressive mode (compared to IKEv1 main mode or IKEv2)? (https://crypto.stackexchange.com/q/27404)
53. https://nohats.ca/wordpress/blog/2014/12/29/dont-stop-using-ipsec-just-yet/

External links

- Computer Security (https://curlie.org/Computers/Security/) at Curlie
- All IETF active security WGs (http://www.ietf.org/html.charters/wg-dir.html#Security%20Area)
  - IETF ipsecme WG (http://datatracker.ietf.org/wg/ipsecme/) ("IP Security Maintenance and Extensions" Working Group)
  - IETF btns WG (https://web.archive.org/web/20070416135452/http://www.ietf.org/html.charters/btns-charter.html) ("Better-Than-Nothing Security" Working Group) (chartered to work on unauthenticated IPsec, IPsec APIs, connection latching)]
- Securing Data in Transit with IPsec (http://www.windowsecurity.com/articles/Securing_Data_in_Transit_with_IPSec.html) WindowsSecurity.com article by Deb Shinder
- IPsec (http://www.microsoft.com/ipsec) on Microsoft TechNet

  - Microsoft IPsec Diagnostic Tool (http://www.microsoft.com/downloads/details.aspx?FamilyID=1d4c292c-7998-42e4-8786-789c7b457881&displaylang=en) on Microsoft Download Center
- An Illustrated Guide to IPsec (http://www.unixwiz.net/techtips/iguide-ipsec.html) by Steve Friedl
- Security Architecture for IP (IPsec) (https://www.ict.tuwien.ac.at/lva/384.081/infobase/P97-IPsec_v4-7.pdf) Data Communication Lectures by Manfred Lindner Part IPsec
- Creating VPNs with IPsec and SSL/TLS (http://www.linuxjournal.com/article/9916) Linux Journal article by Rami Rosen

Retrieved from "https://en.wikipedia.org/w/index.php?title=IPsec&oldid=1026913937"

This page was last edited on 4 June 2021, at 23:44 (UTC).