IPsec

From Wikipedia, the free encyclopedia
Jump to: navigation, search

Internet Protocol Security (**IPsec**) is a protocol suite for securing Internet Protocol (IP) communications by authenticating and encrypting each IP packet of a communication session. IPsec includes protocols for establishing mutual authentication between agents at the beginning of the session and negotiation of cryptographic keys to be used during the session. IPsec can be used in protecting 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]

Internet Protocol security (IPsec) uses cryptographic security services to protect communications over Internet Protocol (IP) networks. IPsec supports network-level peer authentication, data origin authentication, data integrity, data confidentiality (encryption), and replay protection.

IPsec is an end-to-end security scheme operating in the Internet Layer of the Internet Protocol Suite, while some other Internet security systems in widespread use, such as Transport Layer Security (TLS) and Secure Shell (SSH), operate in the upper layers at Application layer. Hence, only IPsec protects any application traffic over an IP network. Applications can be automatically secured by IPsec at the IP layer.

History[edit]

In December 1993, the Software IP Encryption protocol swIPe (protocol) was researched at Columbia University and AT&T Bell Labs by John Ioannidis and others.

In July 1994, Wei Xu at Trusted Information Systems continued this research, enhanced the IP protocols, and completed successfully on the BSDI platform. Wei quickly extended his development on to Sun OS, HP UX, and other UNIX systems. One of the challenges was slow performance of DES and Triple DES. The software encryption was unable to support a T1 speed under the Intel 80386 architecture. By exploring the Crypto cards from Germany, Wei Xu further developed an automated device driver, known as plug-and-play today. By achieving the throughput for more than a T1s, this work made the commercial product practically feasible, that was released as a part of the well-known Gauntlet firewall. In December 1994, it was used for the first time in production for securing some remote sites between east and west coastal states of the United States.[2]

Another IP Encapsulating Security Payload (ESP)[3] was researched at the Naval Research Laboratory as part of a DARPA-sponsored research project, with openly published by IETF SIPP[4] 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, but designed by the Secure Data Network System project of the US Department of Defense. The Security Authentication Header (AH) is derived partially from previous IETF standards work for authentication of the Simple Network Management Protocol (SNMP) version 2.

In 1995, The IPsec working group in the IETF was started to create an open freely available and vetted version of protocols that had been developed under NSA contract in the Secure Data Network System (SDNS) project. The SDNS project had defined a Security Protocol Layer 3 (SP3) that had been published by NIST and was also the basis of the ISO Network Layer Security Protocol (NLSP).[5] Key management for SP3 was provided by the Key Management Protocol (KMP) that provided a baseline of ideas for subsequent work in the IPsec committee.

IPsec is officially standardised by the Internet Engineering Task Force (IETF) in a series of Request for Comments documents addressing various components and extensions. It specifies the spelling of the protocol name to be *IPsec*.[6]

Security architecture[edit]

The IPsec suite is an open standard. IPsec uses the following protocols to perform various functions:[7][8]

- Authentication Headers (AH) provide connectionless integrity and data origin authentication for IP datagrams and provides protection against replay attacks.[9][10]

- Encapsulating Security Payloads (ESP) provide confidentiality, data-origin authentication, connectionless integrity, an anti-replay service (a form of partial sequence integrity), and limited traffic-flow confidentiality.[1]

- Security Associations (SA) provide the bundle of algorithms and data that provide the parameters necessary for AH and/or ESP operations. The Internet Security Association and Key Management Protocol (ISAKMP) provides a framework for authentication and key exchange,[11] 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.[12][13][14][15]

Authentication Header[edit]

Authentication Header (AH) is a member of the IPsec protocol suite. AH guarantees connectionless integrity and data origin authentication of IP packets. Further, it can optionally protect against replay attacks by using the sliding window technique and discarding old packets (see below).

- 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.[10]

- 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.[10]

AH operates directly on top of IP, using IP protocol number 51.[16]

The following AH packet diagram shows how an AH packet is constructed and interpreted:[9][10]

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.[10]

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[edit]

Encapsulating Security Payload (ESP) is a member of the IPsec protocol suite. In IPsec it provides origin authenticity, integrity and confidentiality protection of packets. ESP also supports encryption-only and authentication-only configurations, but using encryption without authentication is strongly discouraged because it is insecure.[17][18][19] 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.[16]

The following ESP packet diagram shows how an ESP packet is constructed and interpreted:[1][20]

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[edit]

Main article: Security association

The IP security architecture uses the concept of a security association as the basis for building security functions into IP. A security association is simply the bundle of algorithms and parameters (such as keys) that is being used to encrypt and authenticate a particular flow in one direction. Therefore, in normal bi-directional traffic, the flows are secured by a pair of security associations.

Security associations 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.[15][21][22] RFC 5386 defines Better-Than-Nothing Security (BTNS) as an unauthenticated mode of IPsec using an extended IKE protocol.

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 identify 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 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[edit]

IPsec can be implemented in a host-to-host transport mode, as well as in a network tunneling mode.

Transport mode[edit]

In transport mode, only the payload of the IP packet is usually encrypted and/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 translated, as this will invalidate the hash value. The transport and application layers are always secured by 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[edit]

Main article: Tunneling protocol

In tunnel mode, the entire IP packet is encrypted and/or 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).

Tunnel mode supports NAT traversal.

Cryptographic algorithms[edit]

Cryptographic algorithms defined for use with IPsec include:

- HMAC-SHA1 for integrity protection and authenticity.
- TripleDES-CBC for confidentiality
- AES-CBC for confidentiality.
- AES-GCM providing confidentiality and authentication together efficiently.

Refer to RFC 7321 for details.

Software implementations[edit]

IPsec support is usually implemented in the kernel with key management and ISAKMP/IKE negotiation carried out from user space. The 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.[23]

Existing IPsec implementations usually include ESP, AH, and IKE version 2. Existing IPsec implementations on UNIX-like operating systems, for example, Sun Solaris or Linux, usually include PF_KEY version 2.

Standards status[edit]

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.[24] IPsec is also optional for IPv4 implementations but due to the slow deployment of IPv6, 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 working group is active at the IETF.[25][26]