IPsec

From Wikipedia, the free encyclopedia

IPsec (**IP security**) is a suite of protocols for securing Internet Protocol (IP) communications by authenticating and/or encrypting each IP packet in a data stream. IPsec also includes protocols for cryptographic key establishment.

Contents

The five-layer TCP/IP model

5. Application layer

DHCP • DNS • FTP • Gopher • HTTP • IMAP4 • IRC • NNTP • XMPP • POP3 • SIP • SMTP • SNMP • SSH • TELNET • RPC • RTP • RTCP • RTSP • TLS/SSL • SDP • SOAP • BGP • PPTP • L2TP • GTP • STUN • NTP • …

4. Transport layer

TCP • UDP • DCCP • SCTP • …

3. Internet Layer

IP (IPv4 • IPv6) • IGMP • ICMP • RSVP • OSPF • ISIS • **IPsec** • ARP • RARP • RIP • …

2. Data link layer

802.11 • ATM • DTM • Token Ring • Ethernet • FDDI • Frame Relay • GPRS • EVDO • HSPA • HDLC • PPP • …

1. Physical layer

Ethernet physical layer • ISDN • Modems • PLC • SONET/SDH • G.709 • Optical Fiber • WiFi • WiMAX • Coaxial Cable . Twisted Pair . …

Summary

IPsec protocols operate at the network layer, layer 3 of the OSI model. Other Internet security protocols in widespread use, such as SSL and TLS, operate from the transport layer up (OSI layers 4 - 7). This makes IPsec more flexible, as it can be used for protecting both TCP- and UDP-based protocols (as well as other layer 4 protocols), but increases its complexity and processing overhead, as it cannot rely on TCP (OSI layer 4) to manage reliability and fragmentation.

Security architecture

IPsec is implemented by a set of cryptographic protocols for (1) securing packet flows and (2) internet key exchange. There are two families of key exchange protocols.

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. The actual choice of encryption and authentication algorithms from (a defined list) is left to the IPsec adminsitrator.

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, 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 gather 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 make the choice.

Current status as a standard

IPsec is a mandatory part of IPv6, and is optional for use with IPv4. While the standard is designed to be indifferent to IP versions, current widespread deployment and experience concerns IPv4 implementations.

IPsec protocols were originally defined by RFCs 1825–1829, published in 1995. In 1998, these documents were obsoleted by RFCs 2401–2412. 2401–2412 are not compatible with 1825–1829, although they are conceptually identical. In December 2005, third-generation documents, RFCs 4301–4309, were produced. They are largely a superset of 2401–2412, but provide a second Internet Key Exchange standard. These third-generation documents standardized the abbreviation of IPsec to uppercase “IP” and lowercase “sec”.

It is unusual to see any product that offers RFC1825–1829 support. “ESP” generally refers to 2406, while ESPbis refers to 4303.

Design intent

IPsec was intended to provide either **transport mode** (end-to-end) security of packet traffic in which the end-point computers do the security processing, or **tunnel mode** (portal-to-portal) communications security in which security of packet traffic is provided to several machines (even to whole LANs) by a single node.

IPsec can be used to create Virtual Private Networks (VPN) in either mode, and this is the dominant use. Note, however, that the security implications are quite different between the two operational modes.

End-to-end communication security on an Internet-wide scale has been slower to develop than many had expected. Part of the reason is that no universal, or universally trusted, Public Key Infrastructure (PKI) has emerged (DNSSEC was originally envisioned for this); another part is that many users understand neither their needs nor the available options well enough to promote inclusion in vendors' products.

Since the Internet Protocol does not inherently provide any security capabilities, IPsec was introduced to provide security services such as the following:

1. Encrypting traffic (so it cannot be read by parties other than those for whom it is intended)
2. Integrity validation (ensuring traffic has not been modified along its path)
3. Authenticating the peers (ensuring that traffic is from a trusted party)
4. Anti-replay (protecting against replay of the secure session).

Modes

There are two modes of **IPsec** operation: **transport mode** and **tunnel mode**.

Transport mode

In **transport mode**, only the payload (message) of the IP packet is encrypted. 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). **Transport mode** is used for host-to-host communications.

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. It must then be encapsulated into a new IP packet for routing to work. **Tunnel mode** is used for network-to-network communications (secure tunnels between routers) or host-to-network and host-to-host communications over the Internet.

Technical details

Two protocols have been developed to provide packet-level security for both IPv4 and IPv6:

- The IP **Authentication Header** provides integrity and authentication and non-repudiation, if the appropriate choice of cryptographic algorithms is made.
- The IP **Encapsulating Security Payload** provides confidentiality, along with optional (but strongly recommended) authentication and integrity protection.

Cryptographic algorithms defined for use with IPsec include HMAC-SHA1 for integrity protection, and TripleDES-CBC and AES-CBC for confidentiality. Refer to RFC 4305 for details.

Authentication header (AH)

The AH is intended to guarantee connectionless integrity and data origin authentication of IP datagrams. Further, it can optionally protect against replay attacks by using the sliding window technique and discarding old packets. 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. In IPv4, mutable (and therefore unauthenticated) IP header fields include TOS, Flags, Fragment Offset, TTL and Header Checksum. AH operates directly on top of IP, using IP protocol number 51. The IP protocol doesn't need IPSec by itself.

An AH packet diagram:

0 - 7 bit	**8 - 15 bit**	**16 - 23 bit**	**24 - 31 bit**
Next header	Payload length	RESERVED	
Security parameters index (SPI)			
Sequence number			
Authentication data (variable)			

Field meanings:

Next header
: Identifies the protocol of the transferred data.

Payload length
: Size of AH packet.

RESERVED

Reserved for future use (all zero until then).

Security parameters index (SPI)
: Identifies the security parameters, which, in combination with the IP address, then identify the security association implemented with this packet.

Sequence number
: A monotonically increasing number, used to prevent replay attacks.

Authentication data
: Contains the integrity check value (ICV) necessary to authenticate the packet; it may contain padding.

Encapsulating Security Payload (ESP)

The ESP protocol provides origin authenticity, integrity, and confidentiality protection of a packet. ESP also supports encryption-only and authentication-only configurations, but using encryption without authentication is strongly discouraged because it is insecure.[1]. Unlike AH , the IP packet header is not protected by ESP. (Although in tunnel mode ESP, protection is afforded to the whole inner IP packet, including the inner header; the outer header remains unprotected.) ESP operates directly on top of IP, using IP protocol number 50.

An ESP packet diagram:

<table>
<tr><th>0 - 7 bit</th><th>8 - 15 bit</th><th>16 - 23 bit</th><th>24 - 31 bit</th></tr>
<tr><td colspan="4">Security parameters index (SPI)</td></tr>
<tr><td colspan="4">Sequence number</td></tr>
<tr><td colspan="4">Payload data (variable)</td></tr>
<tr><td></td><td colspan="3">Padding (0-255 bytes)</td></tr>
<tr><td colspan="2"></td><td>Pad Length</td><td>Next Header</td></tr>
<tr><td colspan="4">Authentication Data (variable)</td></tr>
</table>

Field meanings:

Security parameters index (SPI)
: Identifies the security parameters in combination with IP address.

Sequence number
: A monotonically increasing number, used to prevent replay attacks.

Payload data
: The data to be transferred.

Padding
: Used with some block ciphers to pad the data to the full length of a block.

Pad length
: Size of padding in bytes.

Next header
: Identifies the protocol of the transferred data.

Authentication data
: Contains the data used to authenticate the packet.

Implementations

IPsec support is usually implemented in the kernel with key management and ISAKMP/IKE negotiation carried out from user-space. Existing IPsec implementations tend to include both of these functionalities. However, as there is a standard interface for key management, it is possible to control one kernel IPsec stack using key management tools from a different implementation.

Because of this, there is confusion as to the origins of the IPsec implementation that is in the Linux kernel. The FreeS/WAN project made the first complete and open source implementation of IPsec for Linux. It consists of a kernel IPsec stack (KLIPS), as well as a key management daemon (pluto) and many shell scripts. The FreeS/WAN project was disbanded in March 2004. Openswan and strongSwan are continuations of FreeS/WAN. The KAME project also implemented complete IPsec support for NetBSD, FreeBSD. Its key management daemon is called racoon. OpenBSD made its own ISAKMP/IKE daemon, simply named isakmpd (which was also ported to other systems, including Linux).

However, none of these kernel IPsec stacks were integrated into the Linux kernel. Alexey Kuznetsov and David S. Miller wrote a kernel IPsec implementation from scratch for the Linux kernel around the end of 2002. This stack was subsequently released as part of Linux 2.6, and is referred to variously as "native" or "NETKEY".

Therefore, contrary to popular belief, the Linux IPsec stack did not originate from the KAME project. As it supports the standard PF_KEY protocol (RFC 2367) and the native XFRM interface for key management, the Linux IPsec stack can be used in conjunction with either *pluto* from Openswan/strongSwan, *isakmpd* from OpenBSD project, *racoon* from the KAME project or without any ISAKMP/IKE daemon (using manual keying).

The new architectures of network processors, including multi-core processors with integrated encryption engines, change the way the IPsec stacks are designed. A dedicated Fast Path is used in order to offload the processing of the IPsec processing (SA, SP lookups, encryption, etc.). These Fast Path stacks must be co-integrated on dedicated cores with Linux or RTOS running on other cores. These OS are the control plane that runs ISAKMP/IKE of the Fast Path IPsec stack.

There are a number of implementations of **IPsec** and ISAKMP/IKE protocols. These include:

- 6WINDGate (http://www.6wind.com/6WINDGate-software.html) , Network processor MPU Fast Path IPsec stack
- NRL [1] (http://www.itd.nrl.navy.mil/) IPsec, one of the original sources of IPsec code [2] (http://ezine.daemonnews.org/199812/security.html)
- OpenBSD, with its own code derived from NRL IPsec
- the KAME stack, that is included in Mac OS X, NetBSD and FreeBSD
- "IPsec" in Cisco IOS Software [3] (http://www.cisco.com/en/US/tech/tk583/tk372/technologies_tech_note09186a0080094203.shtml)
- "IPsec" in Microsoft Windows, including Windows XP[4] (http://support.microsoft.com/?kbid=884909) [5] (http://support.microsoft.com/kb/818043/en-us) , Windows 2000[6] (http://www.microsoft.com/windows2000/technologies/communications/ipsec/default.mspx) , and Windows 2003[7] (http://www.microsoft.com/windowsserver2003/technologies/networking/ipsec/default.mspx)
- SafeNet QuickSec toolkits [8] (http://www.safenet-inc.com/products/swTK/index.asp)
- IPsec in Solaris [9] (http://docs.sun.com/app/docs/doc/817-2694?a=expand)
- IBM AIX operating system
- IBM z/OS
- IPsec and IKE in HP-UX (HP-UX IPSec)
- "IPsec and IKE" in VxWorks [10] (http://www.windriver.com/portal/server.pt?space=CommunityPage&cached=true&control=SetCommunity&CommunityID=766)

See also

- Information security
- Year 2005 saw the standardization of a UDP encapsulation mechanism for NAT traversal: NAT-T
- Layer 2 Tunneling Protocol (L2TP)
- Security association (SA)
- Opportunistic encryption

Overview of IPsec-related RFCs

RFC 2367
: PF_KEY Interface

RFC 2401 (obsoleted by RFC 4301)
: Security Architecture for the Internet Protocol

RFC 2402 (obsoleted by RFC 4302 and RFC 4305)
: Authentication Header

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 2406 (obsoleted by RFC 4303 and RFC 4305)
: Encapsulating Security Payload

RFC 2407 (obsoleted by RFC 4306)
: IPsec Domain of Interpretation for ISAKMP (IPsec DoI)

RFC 2408 (obsoleted by RFC 4306)
: Internet Security Association and Key Management Protocol (ISAKMP)

RFC 2409 (obsoleted by RFC 4306)
: Internet Key Exchange (IKE)

RFC 2410
: The NULL Encryption Algorithm and Its Use With IPsec

RFC 2411
: IP Security Document Roadmap

RFC 2412
: The OAKLEY Key Determination Protocol

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 3706
: A Traffic-Based Method of Detecting Dead Internet Key Exchange (IKE) Peers

RFC 3715
: IPsec-Network Address Translation (NAT) Compatibility Requirements

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 (obsoletes RFC 2401)
: Security Architecture for the Internet Protocol

RFC 4302 (obsoletes RFC 2402)

IP Authentication Header
RFC 4303 (obsoletes RFC 2406)
IP Encapsulating Security Payload (ESP)
RFC 4304
Extended Sequence Number (ESN) Addendum to IPsec Domain of Interpretation (DOI) for Internet Security Association and Key Management Protocol (ISAKMP)
RFC 4305
Cryptographic Algorithm Implementation Requirements for Encapsulating Security Payload (ESP) and Authentication Header (AH)
RFC 4306 (obsoletes RFC 2407, RFC 2408, and RFC 2409)
Internet Key Exchange (IKEv2) Protocol
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 4478
Repeated Authentication in Internet Key Exchange (IKEv2) Protocol
RFC 4543
The Use of Galois Message Authentication Code (GMAC) in IPsec ESP and AH
RFC 4555
IKEv2 Mobility and Multihoming Protocol (MOBIKE)
RFC 4621
Design of the IKEv2 Mobility and Multihoming (MOBIKE) Protocol
RFC 4806
Online Certificate Status Protocol (OCSP) Extensions to IKEv2
RFC 4809
Requirements for an IPsec Certificate Management Profile
RFC 4835 (obsoletes RFC 4305)
Cryptographic Algorithm Implementation Requirements for Encapsulating Security Payload (ESP) and Authentication Header (AH)

References

1. ^ Bellovin, Steven M. (1996). "Problem Areas for the IP Security Protocols (http://www.cs.columbia.edu/~smb/papers/badesp.ps) ". *Proceedings of the Sixth Usenix Unix Security Symposium*: 1-16. Retrieved on 2007-07-09.

External links

- Kame Project (http://www.kame.net/)
- racoon and ipsec-tools (http://ipsec-tools.sourceforge.net/)
- IETF IPsec WG has "concluded", archive of the page is here (http://www.ietf.org/html.charters/OLD/ipsec-charter.html)
- IPsec WG still has important active drafts (https://datatracker.ietf.org/public/idindex.cgi?command=show_wg_id&id=1091)
- All IETF active security WGs (http://www.ietf.org/html.charters/wg-dir.html#Security%20Area)
- IETF BTNS WG (chartered to work on unauthenticated IPsec, IPsec APIs, connection latching) (http://www.ietf.org/html.charters/btns-charter.html)
- Securing Data in Transit with IPsec (http://www.windowsecurity.com/articles/Securing_Data_in_Transit_with_IPSec.html)
- Free S/WAN project homepage (http://www.freeswan.org/) .
- Openswan project homepage (http://www.openswan.org/) .
- strongSwan project homepage (http://www.strongswan.org/) .
- Free IPSec VPN Client for Windows (http://vpn.ebootis.de/) .
- Free IPSec VPN Client for MacOS X (http://www.ipsecuritas.com/) .
- The VPN Consortium (http://www.vpnc.org/) .
- A long thread on the ipsec@lists.tislabs.com (http://www.netsys.com/ipsec/2000/msg00777.html) about the capitalization of the letter S in IPsec. The RFCs indicate that it is spelled "IPsec".
- An Illustrated Guide to IPsec (http://www.unixwiz.net/techtips/iguide-ipsec.html)
- A minimal IPsec implementation for embedded systems (http://www.embeddedipsec.org)
- ipsec(4) man page (http://www.openbsd.org/cgi-bin/man.cgi?query=ipsec) via OpenBSD
- IPsec in VoIP Networks (http://www.newport-networks.com/whitepapers/IPsec-1.html) TISPAN has recently selected UDP encapsulation of IPsec to secure signalling in IMS based networks. This paper looks at the issues of IPsec and NAT traversal that lead TISPAN to its choice.
- Windows XP IPSec HowTo (http://slashdot.org/~CronScript/journal/131319)
- A Cryptographic Tour of the IPsec Standards (http://eprint.iacr.org/2006/097)
- Lost in Translation: Theory and Practice in Cryptography - Lessons from the discovery of an IPsec flaw (http://www.computer.org/portal/site/security/index.jsp?pageID=security_level1_article&TheCat=1001&path=security/2006/v4n3&file=crypto.xml)
- ike-scan wiki (http://www.nta-monitor.com/wiki/index.php/Ike-scan_Documentation)
- IPsec Mind Map from MindCert.com (http://www.mindcert.com/2007/03/03/cisco-ipsec-mind-map-cisco-ccie-security-and-ccsp/)

Retrieved from "http://en.wikipedia.org/wiki/IPsec"

Categories: Internet protocols | Cryptographic protocols | Tunneling protocols | Network layer protocols

---