

IPsec Protocol Details for Implementing VPNs

Michael J. Martin

This article is first in a series on implementing VPN gateways using Cisco routers. The Cisco IOS implementation of the IPsec (Internet Protocol Security) suite is an open-standards based framework that provides network administrators with a variety of options to deliver secure IP network communications.

The IPsec framework is a suite of IETF standards that provides for secure transmission of data over unsecured networks, for example the Internet. IPsec provides protocols to secure communications at the Network Layer along with a mechanism for exchanging identity and security protocol management information. The IPsec suite was developed to address some of the fundamental security flaws of IPv4. IP version 4 was built to share information between computers running both like and dislike operating systems over a packet switched network. It could be said that in the 1970s, when TCP/IP was developed, security was not as much of an issue as actually getting the packets to where they belonged. However, when TCP/IP grew from supporting data exchange between a couple hundred systems to a global network of millions of computers, these vulnerabilities became of some concern.

IPsec Protects Against Security Vulnerabilities

There are three major IP security vulnerabilities. Two of these circle around the fact that IP data transmissions between hosts are dependent on each host having a universally unique IP address.

The first is IP spoofing. In order to trust received information, the origin of the information must also be trusted. IP packet delivery is handled on a hop-by-hop basis. An attacker with knowledge of the network topology can disable a system and assume or "spoof" its identity. Since most IP security paradigms revolve around a host's IP address, IP spoofing is problematic for network administrators who need to exchange data securely.

Session hijacking is IP spoofing taken to the next level. Here the attacker disables the spoofed host and assumes active network sessions. This is a far more sophisticated attack than assuming a host's IP identity, as it also depends on software vulnerabilities in order to be successfully executed.

The third vulnerability, traffic sniffing, is endemic to packet switched networks, where the packets are visible to all the network nodes that are connected to the transport medium. Routers, switches, and firewalls all have visibility into packets that pass through these gateways, regardless of the transmission medium. While this makes these devices great for passively filtering IP traffic, it also makes them great places to collect IP packets and extract the data contents.

IPsec Protocol Details

To address these vulnerabilities, the IETF has developed different protocol standard definitions. These standards provide four basic services:

- Data transmission encryption: The originating host can encrypt packets prior to transmission
- Data integrity validation: The receiving host can authenticate each packet sent to ensure the original data that was transmitted was received.
- Data source authentication: The originating host can mark packets, so the receiver can authenticate them.

IPsec Protocol Details for Implementing VPNs

Michael J. Martin

- Data state integrity: The originating and receiving hosts can mark packets, so any re-transmission of the data stream can be detected and rejected (also known as anti-replay).

IPsec implementations use a number of different security protocols to provide these services. From a not so high level, these protocols can be broken down into two different camps: packet protocols and service protocols. The packet protocols are used to provide data security services. There are two IPsec packet protocols: Authentication Header (AH) and Encapsulating Security Payload (ESP). There are a number of service protocols, but the primary one is the Internet Key Exchange protocol (IKE). Below is a basic overview of the protocols in the IOS's IPsec implementation.

Authentication Header: AH, defined in IETF RFC 2402, supports IPsec data validation, authentication and integrity services. It does not support data encryption. AH is typically implemented by itself, but can be implemented alongside ESP. AH is used when we only need to ensure with whom we are exchanging data. You might ask why one would use AH, because it does not support data encryption. Look at it this way: If data encryption is supported by the application, there is no need for additional data encryption. AH is "lighter" in terms of processing overhead, compared to ESP, so it is easy to employ on lower end routers.

Additionally, in comparison to ESP, AH provides greater IP layer security. AH secures the IP packet by generating hash signature data for all of the IP packet information that is not modified in transit. The AH security data is stored in the 32-bit long AH header, which is installed between the IP datagram header and the Layer 4 header. Because AH works by "securing" the IP Packet, AH cannot be employed in network environments where Network Address Translation (NAT) is used. AH operates in either transport or tunnel mode. In most cases tunnel mode is employed and the original IP packet is encapsulated in a new AH secured IP packet. This new packet contains a new IP header (with the destination address of the remote IPsec peer gateway) and the AH header, followed by the original IP packet and Layer 4 datagram. IANA (Internet Assigned Numbers Authority) has assigned ESP the protocol ID of 51.

Encapsulating Security Payload: ESP, defined in IETF RFC 2406, supports IPsec data encryption, validation, authentication and integrity services. ESP can be implemented alone or with AH. While the AH header is pre-pended to the data payload portion of the IP packet, ESP encapsulates the entire data portion of the IP packet with a header and trailer. The ESP header contains the security and sequencing information. The ESP trailer contains variable padding and (if desired) authentication data. ESP's encapsulation of the original ULP data and its encryption requires more router processing resources than AH. Additionally, ESP processing also requires that 1500-byte Layer 4 datagrams be fragmented to support the additional security payload data. Like AH, ESP also supports transport and tunnel operational modes, but almost all vendor implementations implement ESP exclusively in tunnel mode. The ESP RFC does not define what protocols should be used to encrypt the data. Cisco IOS supports 56-DES, 3DES and AES. Other implementations have implemented Blowfish and IDEA as well. IANA has assigned ESP the protocol ID of 50.

Internet Security Association and Key Management Protocol (ISAKMP) and Internet Key Exchange: These provide the framework and processes for implementing IPsec VPN service negotiation. ISAKMP is defined in IETF RFC 2408. IKE is defined in IETF RFC 2409. ISAKMP defines the schemes, syntax and procedures for creating and deleting authentication keys and security

IPsec Protocol Details for Implementing VPNs

Michael J. Martin

associations (SAs). IPsec peers use SAs to keep track of the different aspects of the security services policies negotiated between different IPsec peers. This includes:

- ESP encryption algorithm
- Authentication protocol
- Key information
- Key lifetimes
- SA lifetimes
- AH authentication algorithm

SAs are negotiated between peers when the peer connection is first established. During the establishment (and subsequent re-establishment), each peer assigns the SA it has negotiated with the other its own security parameter index (SPI) number. SPIs are exchanged between peers and used to identify packets. When a peer receives an IPsec packet, it examines the SPI, looks it up in the SPI database, finds the corresponding SA, and then processes the packet according to the rules in the SA. A key point to remember about ISAKMP is that it is independent of the key management protocol, the encryption and authentication used to implement IPsec. It just defines the rules it does not execute them. That is what IKE is for.

IKE is a hybrid of the Oakley key determination protocol and SKEME key exchange protocol. The IKE protocol manages the IPsec security associations within the ISAKMP of IPsec peers. IKE is a protocol available to ISAKMP; they are not one and the same. IKE is the mechanism that establishes the IPsec "connection" between IPsec peers. This requires the negotiation of:

- Authentication algorithms: IKE uses Diffie-Hellman to establish shared secret session keys over the unsecured network transport.
- Confidentiality algorithms: IKE manages the negotiation of the security protocols the peers will use, be they AH, ESP, or AH and ESP in combination.
- Hash algorithms: IKE uses hash algorithms to authenticate packet data.
- Identification keys: IKE supports the use of pre-shared keys, RSA keys (or nonces), digital certificates and Extended Authentication (Xauth) to support identity management.

IKE operates in three modes: main, aggressive, and quick. The main and aggressive modes both achieve the same end, establishing the initial phase or phase 1 IKE SAs. The phase 1 SA bootstraps the IKE process. Once the phase 1 negotiation is completed, quick mode can be used for phase 2 IKE operations that allow for the full SA negotiation and refreshing of SA information when the SA has expired.

The differences between main, aggressive, and quick modes have to do with the degree of security needed and the number of messages exchanged. Main mode uses six (three from the initiator and three from the responder) message exchanges. Main mode commences with the connection initiator proposing a negotiation SA to secure the Diffie-Hellman key exchange. Once the negotiation SA has been established, the Diffie-Hellman keys for quick mode authentication, IKE authentication and SA encryption are generated and exchanged and identity management is completed.

IPsec Protocol Details for Implementing VPNs

Michael J. Martin

Aggressive mode starts with the initiator generating a Diffie-Hellman key, purposing a phase 1 SA and the peer's identity. Then the responder replies with an SA and identity data, with the initiator completing the process with verification data. The entire aggressive mode exchange is done without a negotiation SA, leaving all data exchanged unencrypted. So although the same information is exchanged with both phase 1 modes, one is more secure and the other faster. And although quick mode utilizes the same number of message exchanges as aggressive mode, quick mode does rely on the identity and security integrity established during the phase 1 negotiation.

By now it should be clear that with IPsec everything is negotiated. The security services supported between IPsec peers are negotiated between the two peers when communication is initiated. Depending on the type of peer (i.e., gateway vs. host), there may be a number of IKE policies supported or only one. When a session is initiated, the connecting peer will send all of its supported IKE policies. The remote peer will then respond with a match by comparing the purposed policy with its highest priority policy and subsequent policies in descending order. Only the IPsec services that can be supported by both peers are utilized.

For example, peer Alpha can support data encryption and integrity validation services, but peer Zeta only supports data encryption services. Both Alpha and Zeta need to have a common IKE policy. In this case, Alpha would need to have two different IKE policies: a policy that supports data and integrity services and another policy that supports only data services. In order for Alpha and Zeta to communicate, only data encryption services will be utilized. If Alpha only has a single IKE policy that supports data and integrity, then IKE will terminate the negotiation and the peers will not establish an IPsec connection.