Authentication Header

An Authentication Header or AH is a security mechanism used in authenticating the origins of datagrams (packets of data transmitted under Internet Protocol or IP conditions), and in guaranteeing the integrity of the information that's being sent. Authentication Headers are a protocol under the Internet Protocol Security (IPSec) suite.

When a datagram is sent across the internet, it consists of a payload (the main body of the data itself) and a header (a prefix describing and identifying the packet being sent). An Authentication Header verifies the original source of the packet and ensures that both payload and header have not been altered during transmission.

Placement of an Authentication Header between a datagram's IP header and transport protocol header (layer 4) provides authentication and ensures integrity. An AH must be inserted after the IP header and before any upper layer security protocol such as UDP, TCP, or IDMP. The Authentication Header must also be inserted before any other IPSec header if a combination of security protocols is being used.

Fields Within an Authentication Header

There are several fields which make up a complete Authentication Header. These include:

Next Header: This identifies the next header that will use the specified IP protocol ID (or encapsulated protocol). It has a maximum length of 8 bits.

Payload Length: This indicates the length of the Authentication Header, in 32-bit words.

Security Parameters Index (SPI): In conjunction with a packet's destination address and the security protocol being used (typically AH or Encapsulating Security Payload ESP), this identifies the appropriate security association for the data transmission. At the destination, the receiver uses this information to determine which security association has been used to identify the packet.

Sequence Number: This provides protection against replay attacks, such as those in which an intercepted or captured data stream is continually sent to the same server to precipitate a Denial of Service. The Sequence Number's length may vary, but it has to be a multiple of 32-bit words which once set cannot be allowed to cycle. It indicates the packet number that's been sent over by the security association (SA) for a particular communication. At the receiving end, the Sequence Number may be checked to verify that a packet for its specified security association has not been received before. The packet is rejected, if a datagram with that association has already been received.

Integrity Check Value (ICV): This is authentication data used to verify the integrity of a transmission. The recipient calculates a hash value and compares it to this ICV number (which was calculated by the sender) to verify the message's integrity.

Transport Mode

When used in transport mode, the description of a datagram occurs with its IP header as the outermost identifier, followed by the Authentication Header, then the datagram itself.

Tunnel Mode

In tunnel mode, new IP headers are created dynamically and used in the outermost IP header of a data packet. Authentication Headers may still be used, but this method demands considerably more processor power than transport mode communications.

Ensuring Data Integrity

Authentication Headers ensure data integrity through the use of checksums generated via an authentication code. HMAC algorithms are used to sign data packets for integrity.

To authenticate a datagram's origin, the Authentication Header algorithm incorporates a secret shared key. Relay protection is assured through the Sequence Number field of the Authentication Header.

At the receiver's discretion, Authentication Headers may offer an anti-replay service (partial sequence integrity), as a hedge against Denial of Service or DoS attacks.

Authentication Headers may also be deployed to provide protection for selected parts of an IP header, as for example where the integrity of an IPv6 extension header or an IPv4 option has to be protected in transit.

Encapsulating Security Payload (ESP)

Authentication Headers may be used on their own, or in conjunction with the Encapsulating Security Payload (ESP) protocol. Security services may be initiated between two communicating hosts, between two communicating security gateways, or between a host and a gateway.

Placement and Linking

IPv4 and IPv6 use different methods for placing an Authentication Header into a datagram, and for linking its various headers together. But the AH protocol was essentially designed to use the IPv6 mechanism, which inserts an Authentication Header into the IP datagram as an extension header, according to IPv6 rules for linking extension headers.

The AH is linked by the previous (extension or main) header, which puts the assigned value of the Authentication Header into its Next Header field. The AH in turn links to the next extension header or transport layer header via its own Next Header field.

Limitations

Authentication Headers provide authentication, integrity, and (when specified) anti-replay protection for entire data packets. However, they don't ensure confidentiality, as the AH protocol has no native provision for encrypting data transmissions. Packets protected by an Authentication Header are protected from being modified, but they are still readable to anyone who might happen to gain access to them.

So for situations where confidentiality isn't a requirement or where legal restrictions on encryption may be imposed, Authentication Headers are an appropriate solution. If however encryption is required, then an auxiliary protocol such as ESP (which does provide an encryption service) must be considered.

Internet Security Association and Key Management Protocol - ISAKMP

ISAKMP defines the procedures for authenticating a communicating peer, creation and management of Security Associations, key generation techniques and threat mitigation (e.g. denial of service and replay attacks). As a framework, ISAKMP typically utilizes IKE for key exchange, although other methods have been implemented such as Kerberized Internet Negotiation of Keys. A Preliminary SA is formed using this protocol; later a fresh keying is done.

ISAKMP defines procedures and packet formats to establish, negotiate, modify and delete Security Associations. SAs contain all the information required for execution of various network security services, such as the IP layer services (such as header authentication and payload encapsulation), transport or application layer services or self-protection of negotiation traffic. ISAKMP defines payloads for exchanging key generation and authentication data. These formats provide a consistent framework for transferring key and authentication data which is independent of the key generation technique, encryption algorithm and authentication mechanism.

ISAKMP is distinct from key exchange protocols in order to cleanly separate the details of security association management (and key management) from the details of key exchange. There may be many different key exchange protocols, each with different security properties. However, a common framework is required for agreeing to the format of SA attributes and for negotiating, modifying and deleting SAs. ISAKMP serves as this common framework.

ISAKMP can be implemented over any transport protocol. All implementations must include send and receive capability for ISAKMP using UDP on port 500.

Internet Key Exchange (IKE)

In computing, **Internet Key Exchange** (**IKE**, sometimes **IKEv1** or **IKEv2**, depending on version) is the protocol used to set up a security association (SA) in the IPsec protocol suite. IKE builds upon the Oakley protocol and ISAKMP. IKE uses X.509 certificates for authentication – either pre-shared or distributed using DNS (preferably with DNSSEC) – and a Diffie–Hellman key exchange to set up a shared session secret from which cryptographic keys are derived. In addition, a security policy for every peer which will connect must be manually maintained

Most IPsec implementations consist of an IKE daemon that runs in user space and an IPsec stack in the kernel that processes the actual IP packets. User-space daemons have easy access to mass storage containing configuration information, such as the IPsec endpoint addresses, keys and certificates, as required. Kernel modules, on the other hand, can process packets efficiently and with minimum overhead—which is important for performance reasons.

The IKE protocol uses UDP packets, usually on port 500, and generally requires 4–6 packets with 2–3 round trips to create an SA (security association) on both sides. The negotiated key material is then given to the IPsec stack. For instance, this could be an AES key, information identifying the IP endpoints and ports that are to be protected, as well as what type of IPsec tunnel has been created. The IPsec stack, in turn, intercepts the relevant IP packets if and where appropriate and performs encryption/decryption as required. Implementations vary on how the interception of the packets is done—for example, some use virtual devices, others take a slice out of the firewall, etc.