IPsec (Internet Protocol Security)

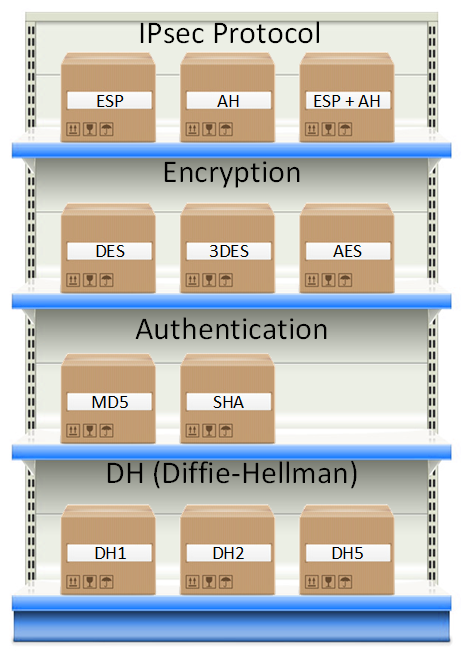
**Lesson Contents**

* [IKE (Internet Key Exchange)](https://networklessons.com/cisco/ccie-routing-switching/ipsec-internet-protocol-security#IKE_Internet_Key_Exchange)
  + [IKE Phase 1](https://networklessons.com/cisco/ccie-routing-switching/ipsec-internet-protocol-security#IKE_Phase_1)
    - [Main Mode](https://networklessons.com/cisco/ccie-routing-switching/ipsec-internet-protocol-security#Main_Mode)
      * [Message 1](https://networklessons.com/cisco/ccie-routing-switching/ipsec-internet-protocol-security#Message_1)
      * [Message 2](https://networklessons.com/cisco/ccie-routing-switching/ipsec-internet-protocol-security#Message_2)
      * [Message 3](https://networklessons.com/cisco/ccie-routing-switching/ipsec-internet-protocol-security#Message_3)
      * [Message 4](https://networklessons.com/cisco/ccie-routing-switching/ipsec-internet-protocol-security#Message_4)
      * [Message 5](https://networklessons.com/cisco/ccie-routing-switching/ipsec-internet-protocol-security#Message_5)
      * [Message 6](https://networklessons.com/cisco/ccie-routing-switching/ipsec-internet-protocol-security#Message_6)
    - [Aggressive Mode](https://networklessons.com/cisco/ccie-routing-switching/ipsec-internet-protocol-security#Aggressive_Mode)
      * [Message 1](https://networklessons.com/cisco/ccie-routing-switching/ipsec-internet-protocol-security#Message_1-2)
      * [Message 2](https://networklessons.com/cisco/ccie-routing-switching/ipsec-internet-protocol-security#Message_2-2)
      * [Message 3](https://networklessons.com/cisco/ccie-routing-switching/ipsec-internet-protocol-security#Message_3-2)
  + [IKE Phase 2](https://networklessons.com/cisco/ccie-routing-switching/ipsec-internet-protocol-security#IKE_Phase_2)
    - [Message 1](https://networklessons.com/cisco/ccie-routing-switching/ipsec-internet-protocol-security#Message_1-3)
    - [Message 2](https://networklessons.com/cisco/ccie-routing-switching/ipsec-internet-protocol-security#Message_2-3)
    - [Message 3](https://networklessons.com/cisco/ccie-routing-switching/ipsec-internet-protocol-security#Message_3-3)
* [IPsec Protocols](https://networklessons.com/cisco/ccie-routing-switching/ipsec-internet-protocol-security#IPsec_Protocols)
  + [Authentication Header Protocol](https://networklessons.com/cisco/ccie-routing-switching/ipsec-internet-protocol-security#Authentication_Header_Protocol)
    - [Transport Mode](https://networklessons.com/cisco/ccie-routing-switching/ipsec-internet-protocol-security#Transport_Mode)
    - [Tunnel Mode](https://networklessons.com/cisco/ccie-routing-switching/ipsec-internet-protocol-security#Tunnel_Mode)
  + [ESP (Encapsulating Security Payload) Protocol](https://networklessons.com/cisco/ccie-routing-switching/ipsec-internet-protocol-security#ESP_Encapsulating_Security_Payload_Protocol)
    - [Transport Mode](https://networklessons.com/cisco/ccie-routing-switching/ipsec-internet-protocol-security#Transport_Mode-2)
    - [Tunnel Mode](https://networklessons.com/cisco/ccie-routing-switching/ipsec-internet-protocol-security#Tunnel_Mode-2)
  + [AH and ESP](https://networklessons.com/cisco/ccie-routing-switching/ipsec-internet-protocol-security#AH_and_ESP)
    - [Transport Mode](https://networklessons.com/cisco/ccie-routing-switching/ipsec-internet-protocol-security#Transport_Mode-3)
    - [Tunnel Mode](https://networklessons.com/cisco/ccie-routing-switching/ipsec-internet-protocol-security#Tunnel_Mode-3)
* [Conclusion](https://networklessons.com/cisco/ccie-routing-switching/ipsec-internet-protocol-security#Conclusion)

IPsec (Internet Protocol Security) is a framework that helps us to protect IP traffic on the network layer. Why? because the IP protocol itself doesn’t have any security features at all. IPsec can protect our traffic with the following features:

* **Confidentiality**: by encrypting our data, nobody except the sender and receiver will be able to read our data.
* **Integrity**: we want to make sure that nobody changes the data in our packets. By calculating a hash value, the sender and receiver will be able to check if changes have been made to the packet.
* **Authentication**: the sender and receiver will authenticate each other to make sure that we are really talking with the device we intend to.
* **Anti-replay**: even if a packet is encrypted and authenticated, an attacker could try to capture these packets and send them again. By using sequence numbers, IPsec will not transmit any duplicate packets.

As a framework, IPsec uses a variety of protocols to implement the features I described above. Here’s an overview:



Don’t worry about all the boxes you see in the picture above, we will cover each of those. To give you an example, for encryption we can choose if we want to use DES, 3DES or AES. For authentication you can choose between MD5 or SHA.

IPsec can be used on many different devices, it’s used on routers, firewalls, hosts and servers. Here are some examples how you can use it:

* Between two routers to create a site-to-site VPN that “bridges” two LANs together.
* Between a firewall and windows host for remote access VPN.
* Between two linux servers to protect an insecure protocol like telnet.

IPsec is pretty complex and there are a lot of different ways to implement it. In this lesson I will start with an overview and then we will take a closer look at each of the components.

Before we can protect any IP packets, we need two IPsec peers that build the IPsec tunnel.

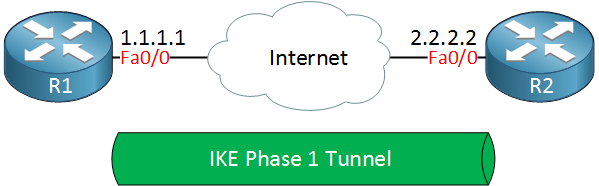
To establish an IPsec tunnel, we use a protocol called **IKE (Internet Key Exchange)**.

There are **two phases** to build an IPsec tunnel:

* **IKE phase 1**
* **IKE phase 2**

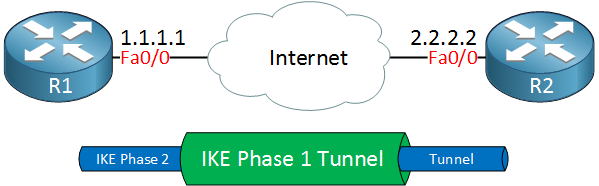
In IKE phase 1, two peers will negotiate about the encryption, authentication, hashing and other protocols that they want to use and some other parameters that are required. In this phase, an **ISAKMP (Internet Security Association and Key Management Protocol)** session is established. This is also called the **ISAKMP tunnel** or **IKE phase 1** tunnel.

The collection of parameters that the two devices will use is called a **SA (Security Association)**. Here’s an example of two routers that have established the IKE phase 1 tunnel:

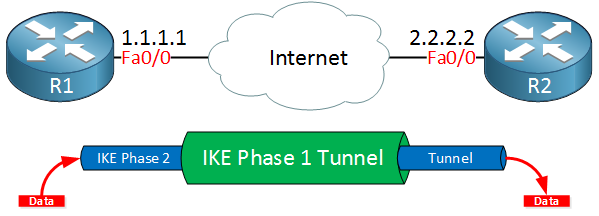


The IKE phase 1 tunnel is only used for **management traffic**. We use this tunnel as a *secure method to establish the second tunnel* called the **IKE phase 2 tunnel** or**IPsec tunnel**and for management traffic like keepalives.

Here’s a picture of our two routers that completed IKE phase 2:



Once IKE phase 2 is completed, we have an IKE phase 2 tunnel (or IPsec tunnel) that we can use to protect our user data. This user data will be sent through the IKE phase 2 tunnel:



IKE builds the tunnels for us but it doesn’t authenticate or encrypt user data. We use two other protocols for this:

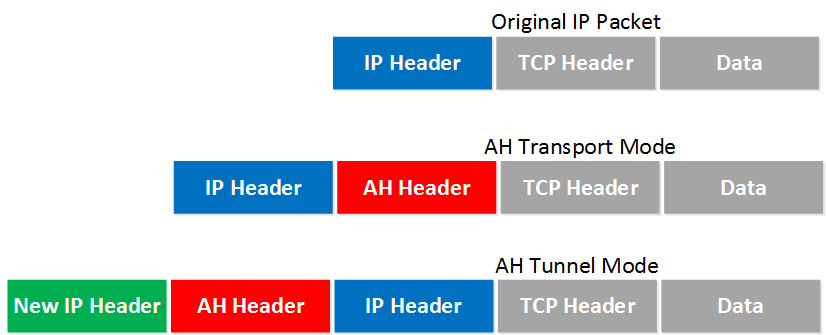
* **AH (Authentication Header)**
* **ESP (Encapsulating Security Payload)**

AH and ESP both offer authentication and integrity but only **ESP supports encryption**. Because of this, ESP is the most popular choice nowadays.

Both protocols support two different modes:

* **Transport mode**
* **Tunnel mode**

The main difference between the two is that with transport mode we will use the **original IP header** while in tunnel mode, we use a new **IP header**. Here’s an example to help you visualize this:



Transport mode is often between two devices that want to protect some insecure traffic (example: telnet traffic). Tunnel mode is typically used for site-to-site VPNs where we need to encapsulate the original IP packet since these are mostly private IP addresses and can’t be routed on the Internet. I will explain these two modes in detail later in this lesson.

The entire process of IPsec consists of five steps:

* **Initiation**: something has to trigger the creation of our tunnels. For example when you configure IPsec on a router, you use an access-list to tell the router what data to protect. When the router receives something that matches the access-list, it will start the IKE process. It’s also possible to manually initiate the tunnel.
* **IKE phase 1**: we negotiate a security association to build the IKE phase 1 tunnel (ISAKMP tunnel).
* **IKE phase 2**: within the IKE phase 1 tunnel, we build the IKE phase 2 tunnel (IPsec tunnel).
* **Data transfer**: we protect user data by sending it through the IKE phase 2 tunnel.
* **Termination**: when there is no user data to protect then the IPsec tunnel will be terminated after awhile.

Now you have an idea of the basics of IPsec, let’s take a closer look at each of the different components.

IKE (Internet Key Exchange)

IKE (Internet Key Exchange) is one of the primary protocols for IPsec since it establishes the security association between two peers. There are two versions of IKE:

* **IKEv1**
* **IKEv2**

IKEv1 was introduced around 1998 and superseded by IKEv2 in 2005. There are some differences between the two versions:

* IKEv2 requires less bandwidth than IKEv1.
* IKEv2 supports EAP authentication (next to pre-shared keys and digital certificates).
* IKEv2 has built-in support for NAT traversal (required when your IPsec peer is behind a NAT router).
* IKEv2 has a built-in keepalive mechanism for tunnels.

The list with advantages goes on but for now, let’s focus on understanding IKE. As explained before, IKE uses two phases:

* IKE Phase 1
* IKE Phase 2

Let’s discuss what happens at each phase. Everything I explain below applies to IKEv1.

IKE Phase 1

The main purpose of IKE phase 1 is to establish a secure tunnel that we can use for IKE phase 2.

We can break down phase 1 in three simple steps:

**Step 1 : Negotiation**

The peer that has traffic that should be protected will initiate the IKE phase 1 negotiation. The two peers will negotiate about the following items:

* **Hashing**: we use a hashing algorithm to verify the integrity, we use MD5 or SHA for this.
* **Authentication**: each peer has to prove who he is. Two commonly used options are a pre-shared key or digital certificates.
* **DH (Diffie Hellman) group**: the DH group determines the strength of the key that is used in the key exchange process. The higher group numbers are more secure but take longer to compute.
* **Lifetime**: how long does the IKE phase 1 tunnel stand up? the shorter the lifetime, the more secure it is because rebuilding it means we will also use new keying material. Each vendor uses a different lifetime, a common default value is 86400 seconds (1 day).
* **Encryption**: what algorithm do we use for encryption? For example, DES, 3DES or AES.

**Step 2: DH Key Exchange**

Once the negotiation has succeeded, the two peers will know what policy to use. They will now use the DH group that they negotiated to exchange keying material. The end result will be that both peers will have a shared key.

**Step 3: Authentication**

The last step is that the two peers will authenticate each other using the authentication method that they agreed upon on in the negotiation. When the authentication is successful, we have completed IKE phase 1. The end result is a IKE phase 1 tunnel (aka ISAKMP tunnel) which is bidirectional. This means that both peers can send and receive on this tunnel.

The three steps above can be completed using two different modes:

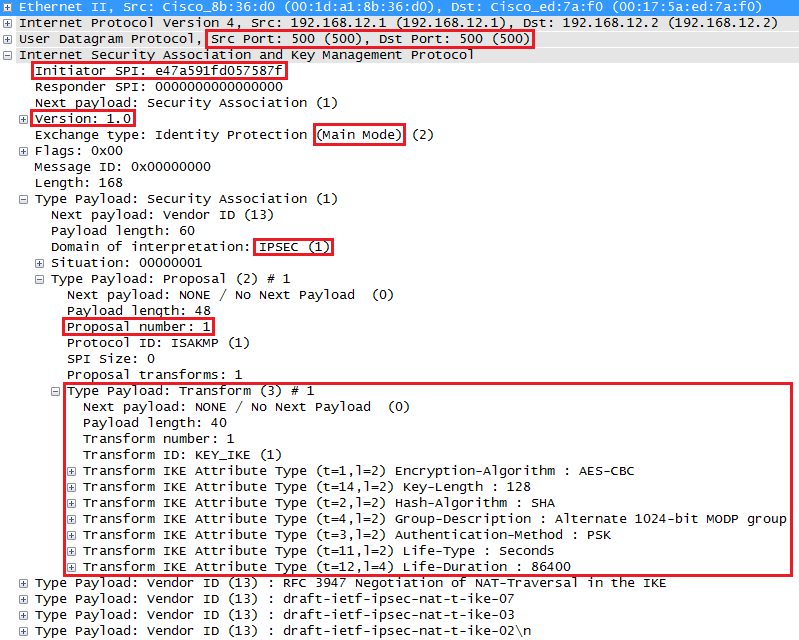
* **Main mode**
* **Aggressive mode**

Main mode uses six messages while aggressive mode only uses three messages. Main mode is considered more secure. Let’s take a look at closer look at both modes.

Main Mode

IKEv1 main mode uses 6 messages. I will show you these in Wireshark and I’ll explain the different fields.

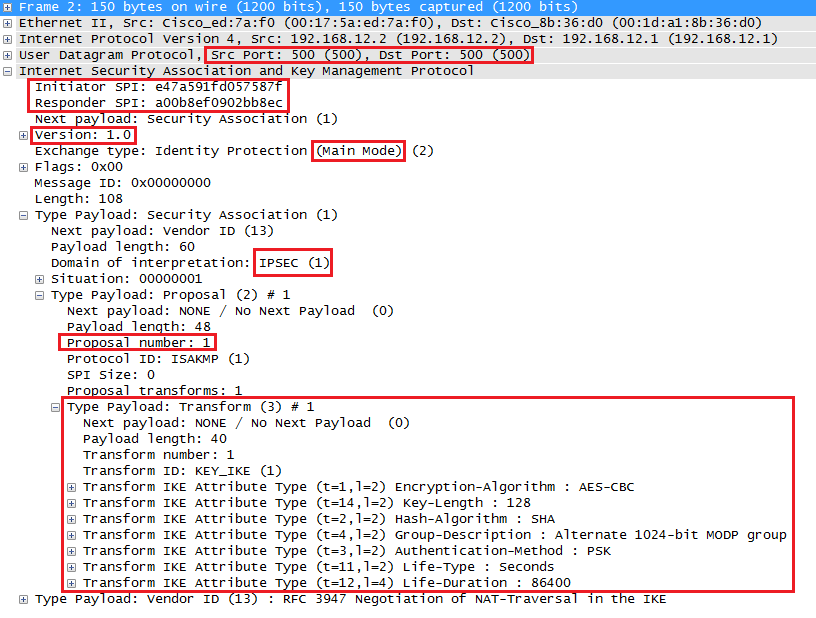
Message 1



The initiator (peer that wants to build the tunnel) will send the first message. This is a proposal for the security association. Above you can see that the initiator uses IP address 192.168.12.1 and is sending a proposal to responder (peer we want to connect to) 192.168.12.2. IKE uses **UDP port 500** for this. In the output above you can see an initiator **SPI (Security Parameter Index)**, this is a unique value that identifies this security association.

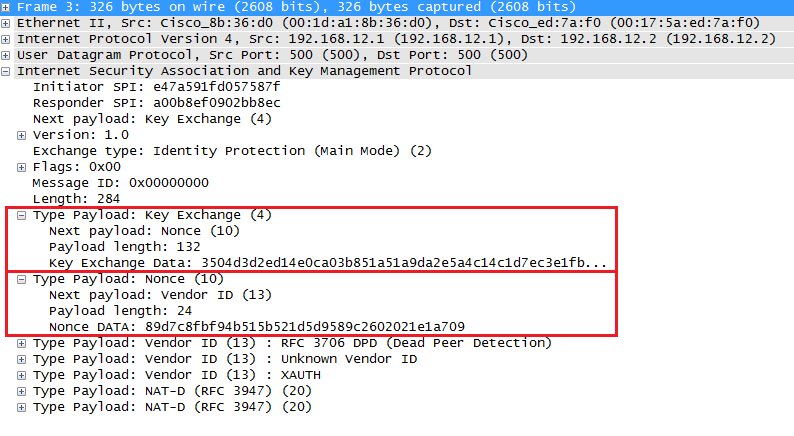
We can see the IKE version (1.0) and that we are using main mode. The domain of interpretation is IPsec and this is the first proposal. In the **transform payload** you can find the attributes that we want to use for this security association.

Message 2



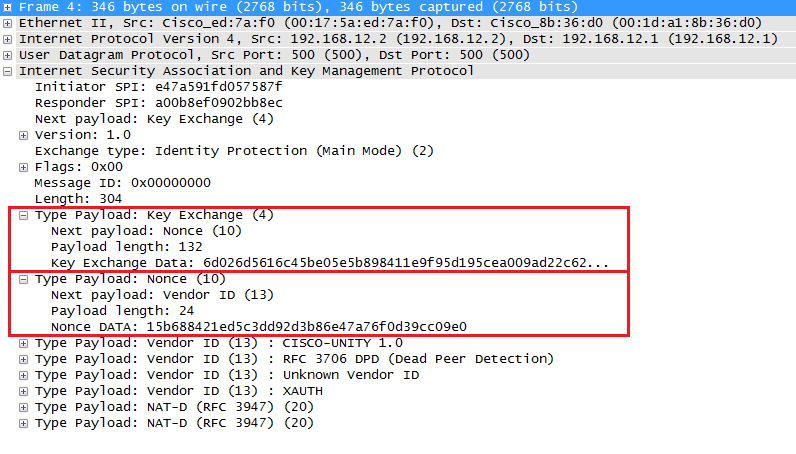
When the responder receives the first message from the initiator, it will reply. This message is used to inform the initiator that we agree upon the attributes in the transform payload. You can also see that the responder has set its own SPI value.

Message 3



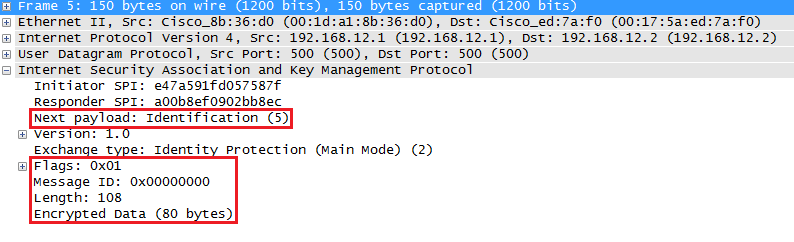
Since our peers agree on the security association to use, the initiator will start the Diffie Hellman key exchange. In the output above you can see the payload for the key exchange and the nonce.

Message 4



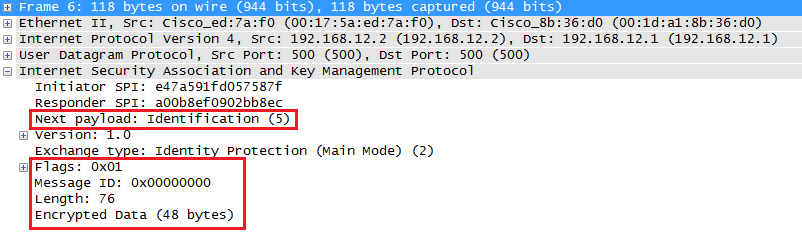
The responder will also send his/her Diffie Hellman nonces to the initiator, our two peers can now calculate the Diffie Hellman shared key.

Message 5



The last two messages are encrypted so we can’t see its contents anymore. These two are used for identification and authentication of each peer. The initiator starts.

Message 6



And above we have the 6th message from the responder with its identification and authentication information. IKEv1 main mode has now completed and we can continue with IKE phase 2.

Before we continue with phase 2, let me show you aggressive mode first.

Aggressive Mode

IKEv1 aggressive mode only requires three messages to establish the security association. It’s quicker than main mode since it adds all the information required for the DH exchange in the first two messages. Main mode is considered more secure since identification is encrypted, aggressive mode does this in clear-text.

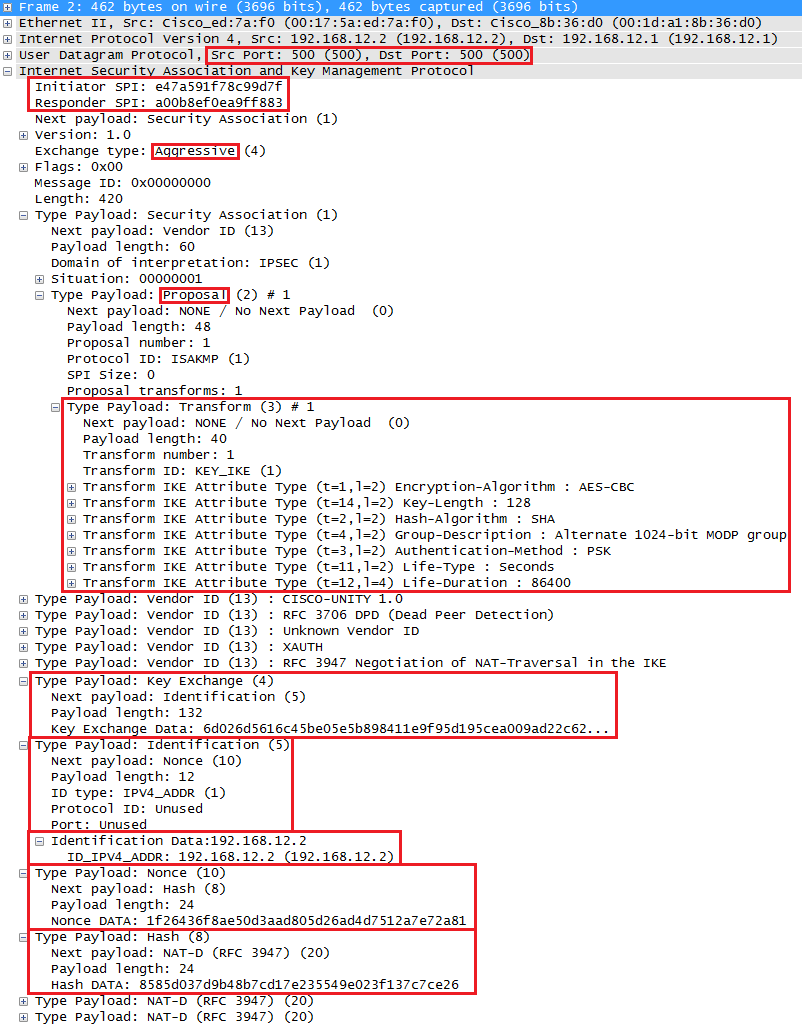
Let’s take a look at the different messages.

Message 1



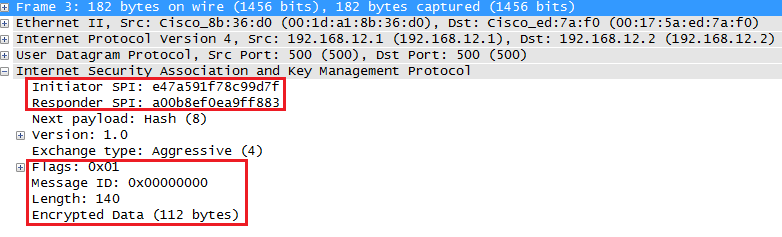
The first message is from the initiator (192.168.12.1) to the responder (192.168.12.2).  You can see the transform payload with the security association attributes , DH nonces and the identification (in clear text) in this single message.

Message 2



The responder now has everything in needs to generate the DH shared key and sends some nonces to the initiator so that it can also calculate the DH shared key. It also calculates a hash that is used for authentication.

Message 3



Both peers have everything they need, the last message from the initiator is a hash that is used for authentication.

Our IKE phase 1 tunnel is now up and running and we are ready to continue with IKE phase 2.

IKE Phase 2

The IKE phase 2 tunnel (IPsec tunnel) will be actually used to protect user data. There is only one mode to build the IKE phase 2 tunnel which is called **quick mode**.

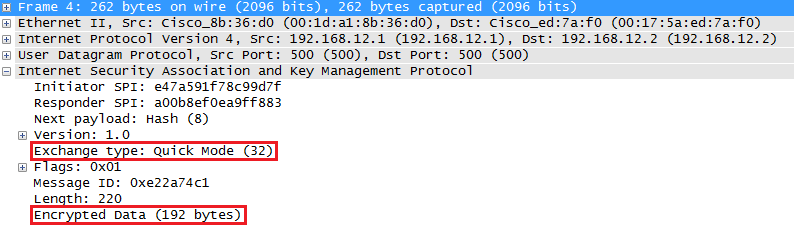
Just like in IKE phase 1, our peers will negotiate about a number of items:

* **IPsec Protocol**: do we use AH or ESP?
* **Encapsulation Mode**: transport or tunnel mode?
* **Encryption**: what encryption algorithm do we use? DES, 3DES or AES?
* **Authentication**: what authentication algorithm do we use? MD5 or SHA?
* **Lifetime**: how long is the IKE phase 2 tunnel valid? When the tunnel is about to expire, we will refresh the keying material.
* **(Optional) DH exchange**: used for PFS (Perfect Forward Secrecy).

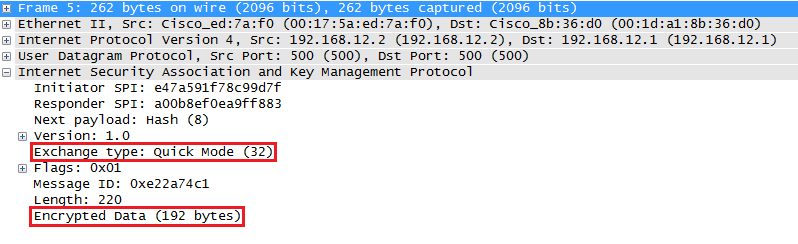
PFS is optional and forces the peers to run the DH exchange again to generate a new shared key in each IKE phase 2 quick mode.

This negotiation happens within the protection of our IKE phase 1 tunnel so we can’t see anything. Just for the sake of completeness, here’s what it looks like in wireshark:

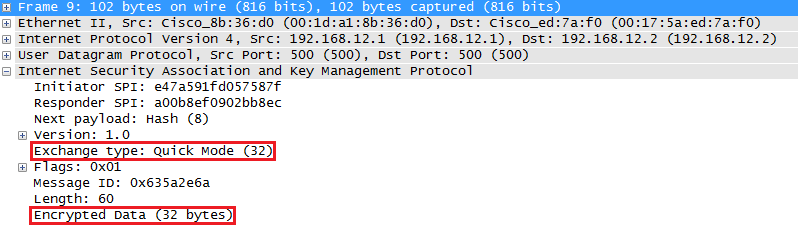
Message 1



Message 2



Message 3



Once IKE phase 2 has completed, we are finally ready to protect some user data. Let’s see how this is done…

IKEv2 doesn’t have a main or aggressive mode for phase 1 and there’s no quick mode in phase 2. It still has two phases though, phase 1 is called the IKE\_SA\_INIT and the second phase is called IKE\_AUTH. Only four messages are required for the entire exchange.

IPsec Protocols

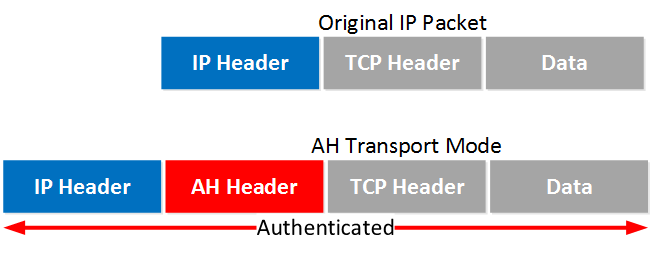
AH and/or ESP are the two protocols that we use to actually protect user data. Both of them can be used in transport or tunnel mode, let’s walk through all the possible options.

Authentication Header Protocol

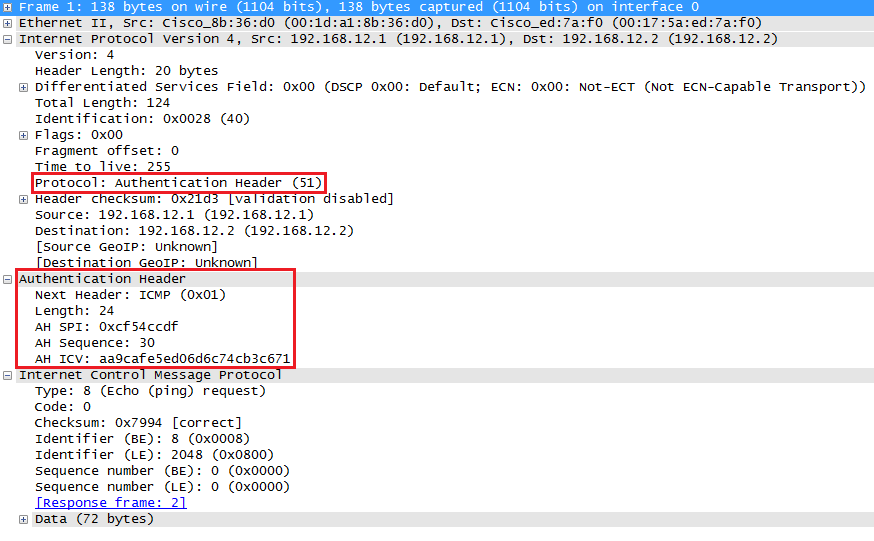
AH offers authentication and integrity but it doesn’t offer any encryption. It protects the IP packet by calculating a hash value over almost all fields in the IP header. The fields it excludes are the ones that can be changed in transit (TTL and header checksum). Let’s start with transport mode…

Transport Mode

Transport mode is simple, it just adds an AH header after the IP header. Here’s an example of an IP packet that carries some TCP traffic:



And here’s what that looks like in Wireshark:



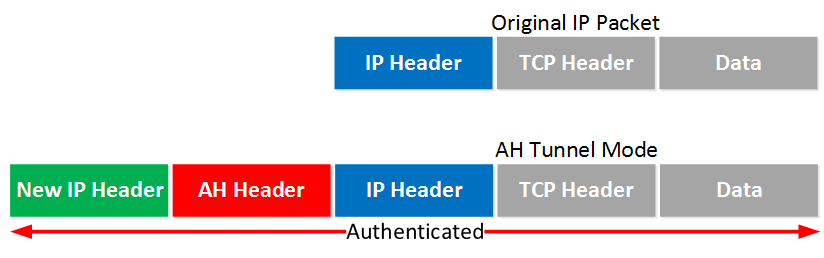
Above you can see the AH header in between the IP header and ICMP header. This is a capture I took of a ping between two routers. You can see that AH uses 5 fields:

* **Next Header**: this identifies the next protocol, ICMP in our example.
* **Length**: this is the length of the AH header.
* **SPI (Security Parameters Index)**: this is an 32-bit identifier so the receiver knows to which flow this packet belongs.
* **Sequence**: this is the sequence number that helps against replay attacks.
* **ICV (Integrity Check Value)**: this is the calculated hash for the entire packet. The receiver also calculates a hash, when it’s not the same you know something is wrong.

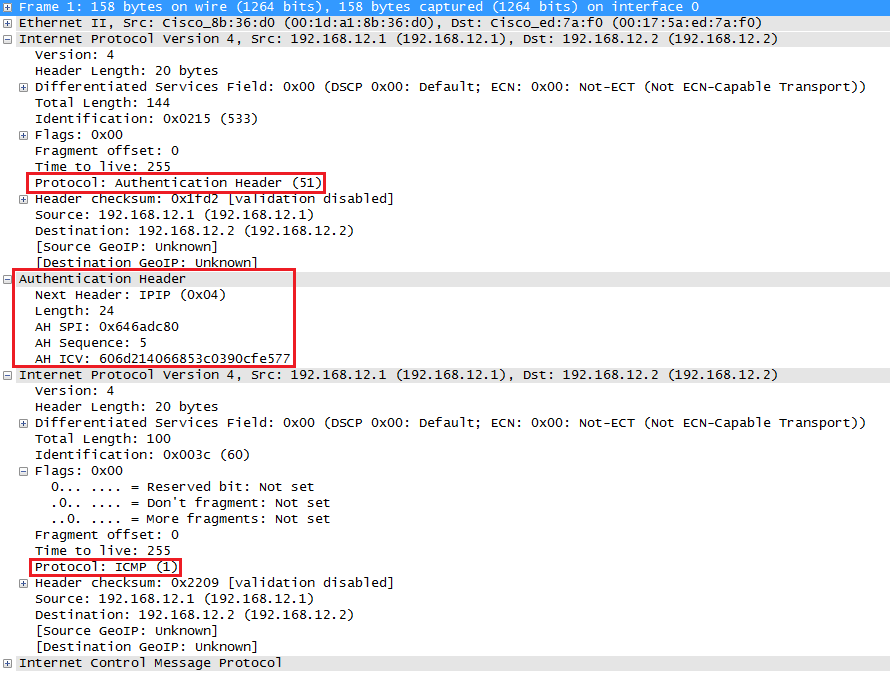
Let’s continue with tunnel mode.

Tunnel Mode

With tunnel mode we add a new IP header on top of the original IP packet. This could be useful when you are using private IP addresses and you need to tunnel your traffic over the Internet. It’s possible with AH but it doesn’t offer encryption:



The entire IP packet will be authenticated. Here’s what it looks like in wireshark:



Above you can see the new IP header, then the AH header and finally the original IP packet that carries some ICMP traffic.

One problem with AH is that it doesn’t play well with NAT / PAT. Fields in the IP header like TTL and the checksum are excluded by AH because it knows these will change. The IP addresses and port numbers however are included. If you change these with NAT, the ICV of AH fails.

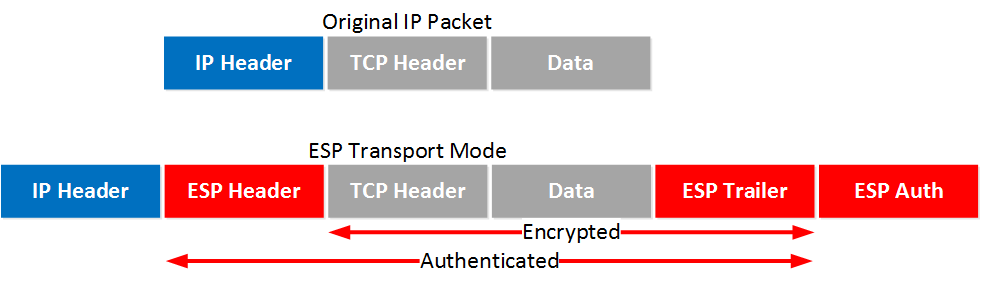
Let’s continue with ESP…

ESP (Encapsulating Security Payload) Protocol

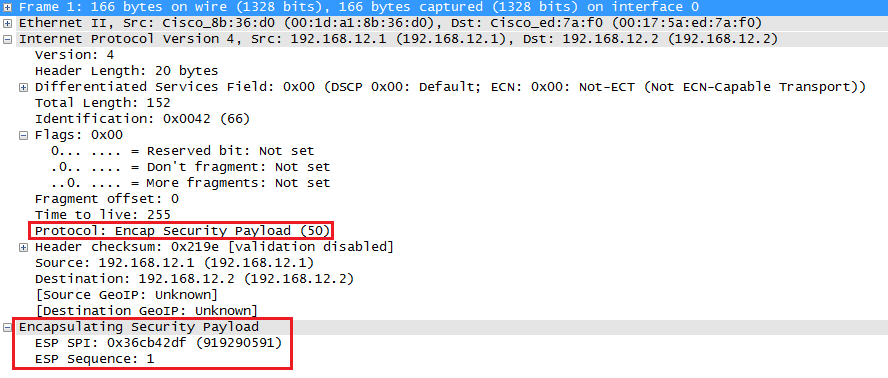
ESP is the more popular choice of the two since it allows you to encrypt IP traffic. We can use it in transport or tunnel mode, let’s look at both.

Transport Mode

When we use transport mode, we use the original IP header and insert an ESP header. Here’s what it looks like:



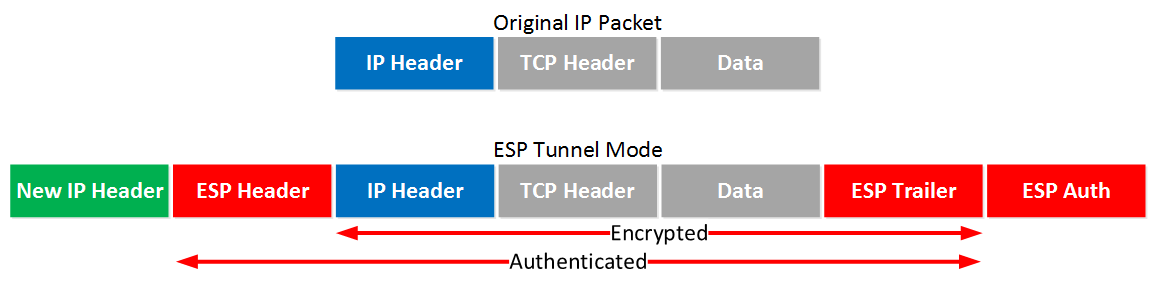
Above you can see that we add an ESP header and trailer. Our transport layer (TCP for example) and payload will be encrypted. It also offers authentication but unlike AH, it’s not for the entire IP packet. Here’s what it looks like in wireshark:



Above you can see the original IP packet and that we are using ESP. The IP header is in cleartext but everything else is encrypted.

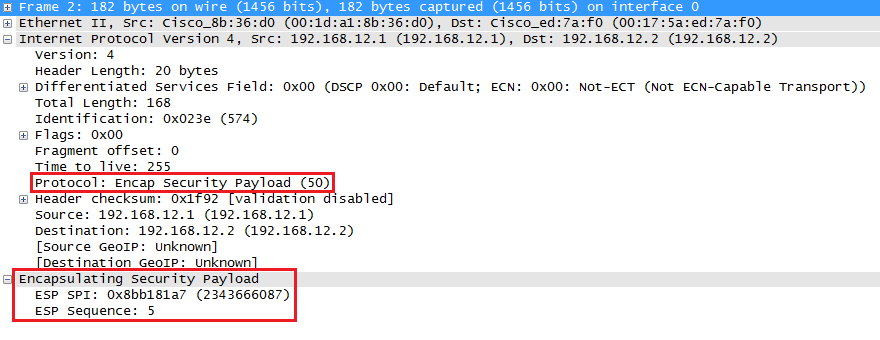
Tunnel Mode

How about ESP in tunnel mode? This is where we use a new IP header which is useful for site-to-site VPNs:



It’s similar to transport mode but we add a new header. The original IP header is now also encrypted.

Here’s what it looks like in wireshark:



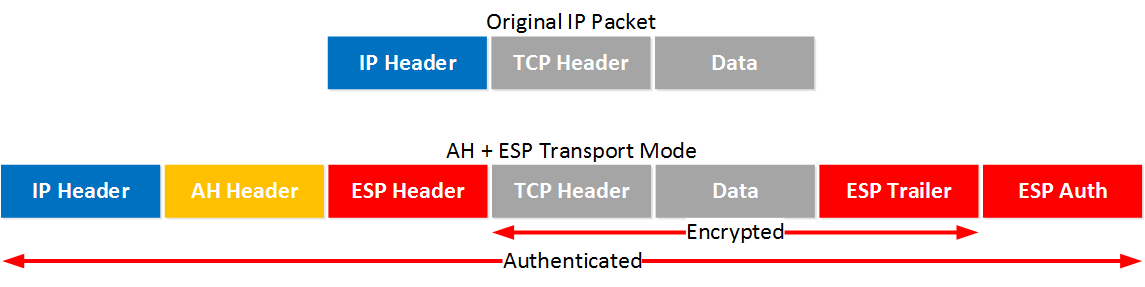
The output of the capture is above is similar to what you have seen in transport mode. The only difference is that this is a new IP header, you don’t get to see the original IP header.

AH and ESP

This one confuses a lot of people, it’s possible to use AH and ESP at the same time. Let’s check it out!

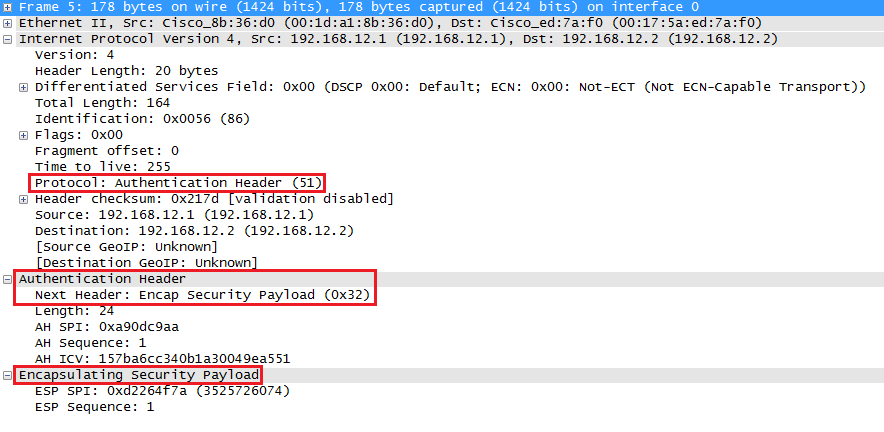
Transport Mode

Let’s start with transport mode, here’s what the IP packet will look like:



With transport mode we will use the original IP header, followed by an AH and ESP header. The transport layer, payload and ESP trailer will be encrypted.

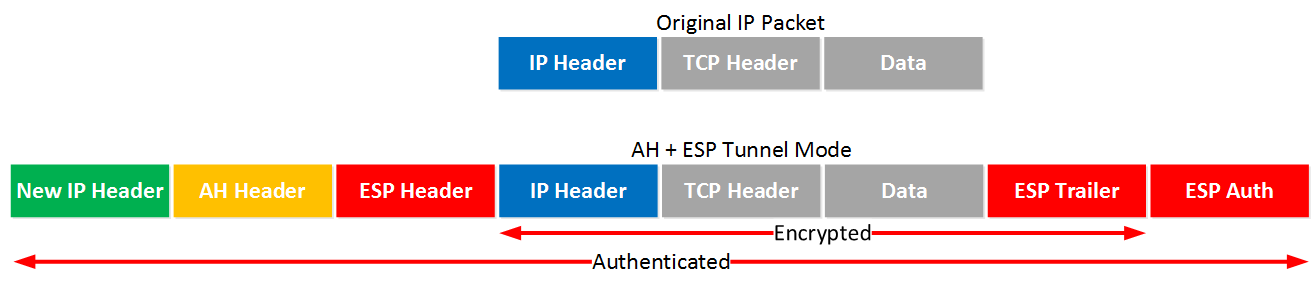
Because we also use AH, the entire IP packet is authenticated. Here’s what it looks like in wireshark:



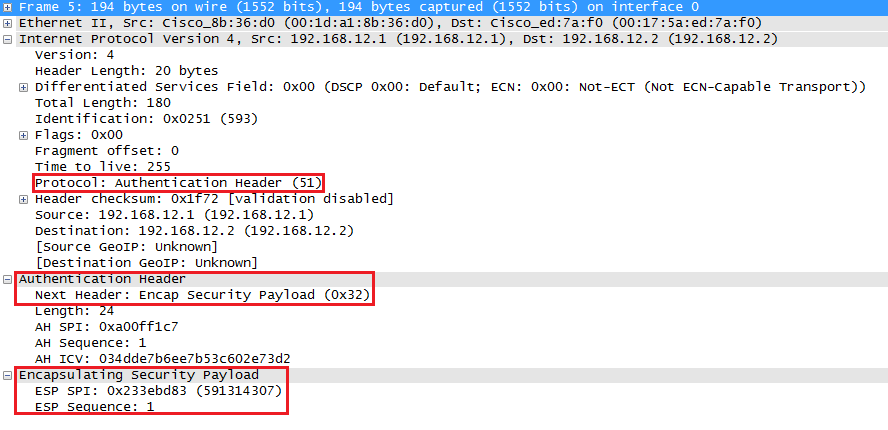
Above you can see the original IP packet, the AH header and the ESP header.

Tunnel Mode

What about tunnel mode? We will add a new IP header:



First we will have a new IP header followed by the AH and ESP header. The original IP packet will be completely encrypted and everything will be authenticated thanks to AH. Here’s what it looks like in wireshark:



Above you can see the new IP header followed by the AH and ESP header.

Conclusion

IPsec is pretty complex…you have now seen how IKE is used to build the IPsec tunnel and how we can use AH and/or ESP to protect our traffic. Do you want to take a look at these wireshark captures yourself? I saved all of them for you😊

[**IPsec IKEv1 phase 1 main mode**](https://www.cloudshark.org/captures/ff740838f1c2)

[**IPsec IKEv1 phase 1 aggressive mode**](https://www.cloudshark.org/captures/e51f5c8a6b24)

[**IPsec IKEv2**](https://www.cloudshark.org/captures/767a93d720ad)

[**IPsec AH transport mode**](https://www.cloudshark.org/captures/9c563cd2501e)

[**IPsec AH tunnel mode**](https://www.cloudshark.org/captures/4d1561a5935f)

[**IPsec ESP transport mode**](https://www.cloudshark.org/captures/993215a4a0d9)

[**IPsec ESP tunnel mode**](https://www.cloudshark.org/captures/58006657c867)

[**IPsec AH+ESP transport mode**](https://www.cloudshark.org/captures/b7e2a9ad7a06)