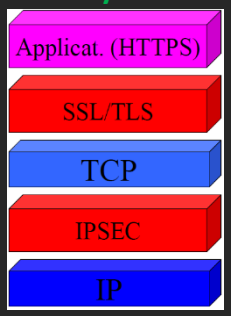
IPSEC

We started by saying '*we use cryptography for information security*', by listing requirements for information security (confidentiality, data integrity, authentication…) and we showed how to use cryptography for achieving those requirements. After a while we started seeing more complex applications of cryptographical approach, for instance we started seeing protocols where Alice and Bob were exchanging messages in order to reach some important goal, authentication maybe but not only authentication, even the last homework is an example of how simple messages can be setup to reach something useful for one application. The first complex protocol we have seen is **Kerberos**, it is complex because the parties are exchanging complex messages and they are using session keys, hashing and so on. What is a protocol? It is an implementation of some idea, but it is more than just cryptography because a protocol also provides details about how to decide a key, in what format you should exchange informations and so on. Then there can be more implementations of a same protocol and it could be possible that they are not interoperating, they are not exchangeable. We aim at a protocol that has not this lack. We aim at a *way to do*, not at a real protocol.

Now we start to see some security protocols **(IPSEC, SSL/TLS)** but the idea is that we don't want to be able to develop an implementation of the protocol due to the high number of details these protocols we are going to discuss have. You will be asked to make choices, the committer will ask you "you have to design a secure application, it is a web application, make it secure", but what is means secure? The committer replies with "I don't know, it is up to you to decide what is secure". What we mean is that we do not want to know what kind of encryptor to use, we want to know for example whether IPSEC is able to grant data integrity, or authentication, or both and so on. This is called *black-box analysis*. Sometimes at the exam I ask such kind of analysis and many people they open the box and tell me "ok this works by deciding the D-H key exchange…" but this is not a *black-box analysis*.

Security architecture and protocol stack



In the classical protocol stack we have what is called the *application level*, what is called the *transport level (TCP)*, then the *network level (IP)* and then lower levels. We are not considering lower levels, we just consider from *IP level* to higher levels. **IPSEC** works between TCP (transport level) and IP (network level), so TCP uses IPSEC (remember that *using* in this context means that the higher level uses the lower level). IPSEC is therefore providing security on *datagrams*. It is also called **IP Security**. We are not considering data belonging to other network levels, we only focus on datagrams. Remark that datagrams travel router to router until they reach the destination. What is important is that from datagram above everybody uses the same standards. Implementing security on datagrams means we are considering one datagram at each time and we are securying one datagram at each time.

So we may think to encrypt a datagram for confidentiality purposes, but datagram is composed by header and payload, we cannot encrypt whole datagram because for example in the header there is the address of destination, so I cannot encrypt it so I can encrypt the payload, this would be a good approach. We will see this.

Important: if you secure the application level, the security is bounded within that application, if you want to secure all applications you have to implement security on lower levels, and that's what IPSEC does, IPSEC secures *all the datagrams* of *all the applications*.

IPSEC is used for *authentication*, *confidentiality*, and *key management*.

- Authentication → we are able to undertand who are the 2 parties that are exchanging messages. Authentication also implies *Data Integrity* in some sense.
- Confidentiality → we know perfectly what it means
- Key management → it is a standard for generate session keys, for communicating sessio keys, for handling permanent keys and so on. We will not see this feature of IPSEC.

IPSEC implements all this but this not means I want all of them. Maybe I want only authentication, or only confidentiality. It is possible. There are several documents explaining IPSEC standards. Few years ago IETF (Internet Engineering Task Force) decided that IPSEC should be considered no longer mandatory for IPv6. It means that for some times IPv6 was also inlcuding IPSEC as a standard, like if IPSEC is a subset of IPv6. Now it is an optional (we can decide to add IPSEC to IPv6).
In order to make things simple we will ignore IPv6, also because it is not so used as IPv4. So will only consider IPv4.
Slide 7: the drawing is showing some use cases for IPSEC. The most common case is using IPSEC between the ending point of a LAN and the starting point of another LAN. This means that within a LAN packets are normal packets, but packets between different LANs are protected by IPSEC, because basically we expect to know who is in our LAN (maybe our home) and who is in the other LAN (maybe our office), but what is between the two LAN's is unknown.

VPN (Virtual Private Network)
From the user perspection VPN is a service where the user sends all the traffic to some remote Server (that can be wherever), you will be building a connection that is protected under many points of views so that all your traffic outgoing your adapter is going to that Server, if you try to analize your traffic it is all encrypted, it is like there is a very secure tunnel. The traffic cannot be seen by some eavesdropper. The server will act as a **proxy**, so your packets will have as source the VPN server, not your address. One way to build a VPN server is implementing IPSEC. There are many VPN server that base security on IPSEC. Why all of a sudden I decided to talk about VPN? Because in the usecase of the slide 7 the user is at home, using a standard host in home and he wants to connect to the enterprise network, he needs to be protected because enterprises are issuing several policies, so enterprise wants that the packets that it receives are protected, so user must use a VPN. The most popular implementation of VPN is Cisco-based VPN which relies on IPSEC.
Benefits of IPSEC:

- you can setup your gateaway in order to provide strong security for the traffic outcoming and incoming your gateway, so you do not need to protect the whole local traffic, it is sufficient to protect the gateway
- you cannot bypass the gateway just because it is not working, the attacker is not able to change the behaviour of the gateway from a remote position
- it is transparent to applications, so changing things at low level is okey because applications will continue to work in the same way
- it is transparent to end users, and this is important because end users can be not so expert in IT and Computer Science

IPSEC Implementation

Basically IPSEC is implemented as an extension header that follows the main IP header. The IPSEC header is composed by two parties:

- **AH** (Authentication Header): it is the header that provides *data integrity* and *authentication*. It is based on MACs (in particular HMAC) and requires a shared secret key.
- **ESP (**Encapsulating Security Payload): it is the header used for *encryption*

You can choose to use one of them, the other one, or both (that's why we said IPSEC provides many services but it is not mandatory to choose them all).
Actually ESP can also provide some authentication features, that overlap with the ones of AH, therefore ESP can be divided into ESP and ESP_a.

IPSEC implicitly offers a lot of services as:
Access Control: prevents unauthorized use of resources by means of authentication.

Connectionless Integrity: data integrity without having a connection, IPSEC is able to check changes to individual IP datagrams.
Data Origin Authentication: it verifies the identity of the claimed source of data, providing security also againts replay attack.
Confidentiality: again IPSEC can provide confidentiality, encryption of data, but it is optional.
Limited Traffic Flow Confidentiality: prevents the attacker from analyzing data in order to retrieve some information about the datagrams incoming and outgoing our interface.

IPSEC: AH vs ESP vs ESP_a

	AH	ESP (encryption only)	ESP (encryption plus authentication)
Access control	✔	✔	✔
Connectionless integrity	✔		✔
Data origin authentication	✔		✔
Rejection of replayed packets	✔	✔	✔
Confidentiality		✔	✔
Limited traffic flow confidentiality		✔	✔

ESP_a is overlapping some authentication features provided by AH. Now we start to learn about the possibility to have some overlap. It is possible to use whatever one likes, so it is possible to use only ESP, only AH, both and so on, but often it is not so interesting combining AH and ESP_a.

Security Assotiations (SA)

Security Association is an association between sender and receiver. It is a one-way association. It is defining the parameters of sending datagrams from Alice to Bob. Message from Bob to Alice will be described by another assotiation. All security associations are stored into a database (SABD). The standard states that for every entity (in particular for every interface) there must be implemented two SADBs, one that stores Sas for incoming traffic and one for Sas for outgoing traffic. We want them separate.
A Security Association is defined by 3 parameters:

1. **SPI** (Security Parameter Index): it is a pointer to information about the security you are going to use
2. **IP Address of Destination**
3. **Security Protocol Identifier**: tells whether you are using ESP or AH (*important, if you want both ESP and AH you will need two Security Assotiations)*

It is interesting to take note of some technical details about these Security Assotiations. If you want to have a bidirectional traffic, since a SA is one-way, of course you need two Security Assotiations.

SA's parameters

- **Sequence Number Counter**: counter similar to the TCP one, it is used to generate sequence numbers for AH and ESP headers
- **Sequence Number Overflow**: it is a flag saying whether the counter goes overflow, what happens when the counter reaches the maximum value? Should I just restart from scratch? No!
- **Anti-Replay Window:** used to determine whether traffic is authentic or replayed
- **AH informations:** what authentication algorithm is used, which is the key and so on
- **ESP informations**: what are the authentication and the encryption algorithms, what is the initialization value, keys and so on
- …

Fragmentation attack: typical *DoS* attack, I'm sending data fragmented in datagrams, you know that fragments have headers containing bits that say "this is last frag" and so on such that the receiver can reconstruct the order of packets. The concerned attack is based on faking the sequence number of fragments, so the attacker changes the offset or the bit that says whether the fragment is the last or not.

Security Policy Database and SA selectors

It is an important Dasatabse, this is not the one collecting the SAs, each entry of such database is associated with a set of IP and a set of Upper Layer protocol fields that you may be using from above, such fields are called *Selectors*. Then each entry is pointing to a SA for that type of traffic.

Selectors are used for implementing filters, so you can analyze the pattern of the traffic against the filter. In order to realize these type of filters the process is like this one: compare the numbers you are reading in the several fields with the pattern you are storing in the entries of the database, and when you find a match, the match will tell you what is the correct assotiation (if no match, it means bypass of IPSEC). Some selector are showed on *slide 25*.

Transport & Tunnel Modes

IPSEC can be used in two different modes: *Transport Mode* and *Tunnel Mode*. Keep in ming that they are *modes of operation*.

The difference is that one of them is more secure, the other one is less secure. Why not using always the most secure way? Because it is demanding in terms of computational effort, so using one instead of another depends on the case.

Transport mode: you can immagine you have some original datagram, it has a header and a payload. In this mode we have IP header, a paylod, and we add an extra header (IPsec) between them. The structure is:

|IP header|IPsec|payload|

While you are sending the packet over the internet the many routers that are processing the packet they are seeing datagram, they will see only IP header and payload and process them as normal fragments, when the packet reaches the destination the destination will process the IPsec header. This is *end-to-end* protection. Since the original IP header is not touched, the routing path is intact. This is not completely true because there are some exceptions (you can protect not only the payload but also some part of the header, this with additional header).

Tunnel mode: payload and IP header will be *encrypted* and will become a payload of a new datagram having a new IP header and the IPsec with them. The structure is:

|new IP header|IPsec|[old IP header|old payload]

In this approach you are incapsulating every single datagram within a new payload.

Usecase: imagine you want to connect to your workstation in your office from your computer at home in a protected way. Tunnel mode is what you need because you will encrypt the full datagram (included header that will contain the IP address of your workstation), and in the new header you will put the IP address of the gateway of the LAN of your office. Then when the gateway reaches the datagram, it will decrypt the payload and it will get from the original header the IP address of the workstation. With tunnel mode you are hiding in the original IP header the IP/port number of destination. Take care of the fact that in this mode the final fragment is much bigger than the original one and you cannot always afford such effort.

Warning: datagrams can be fragmented, if you increase the size of your datagram it will be fragmented and you will be vulnerable to *fragmentation attack*.

The transport mode is preferred in the cases of *end-to-end* communications. It means that you are connecting the two hosts by your private link, your personal link, without using the internet, so you do not need a secure "tunnel" as the other mode does.

The tunnel mode is preferred for *network-to-network* communications, *host-to-network* communication and *private host-to-host* communications like a private chat.

Question: in tunnel mode, since we encrypt the original header, how can we be sure that the datagram will reach the destination?

Answer: the new header will contain the address of the gatewey which implements IPSEC. Since it implements IPSEC it knows that it will find in the original header the address of the computer in the LAN where the gateway is connected.

We have seen tunnel and transport mode in a general way, we have not introduced AH and ESP ingredients.

AH+Transport Mode → you will be authenticating the payload and some portion of IP header that is meant to be never changing

AH+Tunnel Mode → you will be authenticating the whole original datagram and some portion of the new header
ESP+Transport Mode → you are encrypting the original payload
ESP+Tunnel Mode → you are encrypting both payload and IP header
ESP_a+Transport Mode → you are encrypting the payload and you are authenticating the payload, not the IP header. This is the difference between ESP_a and AH.
ESP_a+Tunnel Mode → you are able to encrypt and authenticate all the original datagram.