The longest short IP Sec Paper

@ Articles -> Networking Aug 13 2004, 00:02 (UTC+0)

delete852 writes:

The longest short guide to IPSec
By Nick Vasilyev

Recently it has come to my attention that I don't know as much about VPN as I wanted to know so I started to do research, I was able to find good material on almost every aspect of it except IPSEC. Most of the papers I found on it were either too long, or too short, or didn't go in detail enough or were written in about 150% English which made reading them so dry I had to have a high caffeine drink every 10-15 minutes. So I decided to write my own paper to assist other people in their studies of security and IPSec. Now I don't mean to be bashing anyone's work or anything, and I am not saying that this one will be better, but I think there is room on the internet for another IPSec paper. I will try to keep this paper pretty short; I know how everyone hates long papers that seem to go on and on.

Ok, so lets get straight in it, when a company has a lot of users who work from home, or a lot of small branch offices, they have 2 main options to link everyone together. To have a private line set up from all the offices into the main office, which would cost a hell of a lot, or to set up VPN connections. We are going to cover the second case. Virtual Private Networks are set up over an un-trusted network such as the internet to secure the data flow within them. So for example if home users would need to connect in a company XYZ the users would connect to their ISP's and then once that connection has been established they would use that connection to connect to the VPN.

Since the data is flowing through an untrusted space, there is an obvious need for security of data. There are a number of protocols that you can use to transfer the data, however the one that is pretty much always used is L2TP. It was developed by Cisco and Microsoft when they combined their VPN protocols L2P and PPTP respectively. L2TP doesn't have built in encryption, that's why you need something to work with it for data security, and that something is usually IPSec. Before we get into technical aspects of IPSec let me just review some basic security information, and what we are looking for in a security protocol.

First of there is this CIA of security; it stands for Confidentiality Integrity and Authentication. If the data meets all of these standards it means that it is secure.

Confidentiality – Means that no one else can read the data as it travels down to your computer. So if you send an e-mail it would be confidential if on the way down even if it was intercepted, an attacker would not be able to read it.

Integrity – Integrity is often even more important that Confidentiality. Integrity means that the data can not be modified by anyone on its way to the user. If you think about it, integrity is a lot more important that confidentiality in some aspects, mainly because if data is changed it could cause a lot more problems. If a research scientist was waiting for an e-mail from his assistant about results of some experiment, now if an attacker intercepted that message, and modified the data, it could be much worse, than him just receiving the data.

Authentication - Means that first of all, the sending party is who it says it is, and two that the packets that are coming in, are coming in from a machine that they say they come in from. I will expand on this later. Authentication also applies to users, services, machines, and etc.

Ok so now these categories also break down in specific parts that needed by a secure protocol. I got the list from
an IP Sec paper written by Ghislaine Labouret. So far it is the best documentation of the protocol that I was able to find. Most of the information in this paper can also be found there. Here is how CIA breaks down in IP Sec:

Confidentiality:
- Means that only the intended parties has access to the information
- Also we hope to prevent Traffic analysis with Confidentiality too. Traffic Data Analysis can be achieved by analyzing structure of the packet, the header, source/destination addresses, size of the packets and how frequently they are sent.

Integrity:
- **Connectionless integrity** – means that you can verify that packets have not been modified. Personally I have no idea why they called this connectionless integrity, it's just a longer and a harder to remember name, so from now on Data Integrity and connectionless integrity will mean the same thing.
- **Integrity in connected mode** – This means that there is no loss of packets, and that they all came in, in the correct order, and everything is fine. I also don't know why they called this Integrity in connected mode, so from now on I think I will refer to this as connection integrity.

Authentication:
- **Authentication of a party** – means that the computer that is sending the data has been authenticated to be who it says it is. I am fine with this name so we aren't going to change

it.
- **Data Origin Authentication** – Means that if a packet's source address is 10.0.0.1 then the computer 10.0.0.1 actually sent the data. Basically this checks for spoofing. I don't really like the name of this one either, but we are going to leave it as it is.

Ok, so good stuff right? Right! These are the security elements that we need to achieve to have a secure communication of hosts; I suggest you make yourself real familiar with this list because it is going to be covered in a lot of cases from now on. Now remember, that Authentication and Integrity go hand in hand because if the Data is authenticated, but data is captured, and then changed and resent (like in a man-in-the middle attack) the victim could be receiving what he thinks is authenticated data, but it could be modified. Now if there is no authentication but good integrity then the host can be receiving any data from a spoofed IP.

Now, one of the best things about IP Sec is its flexibility. First of all IP Sec's security is broken up in 2 modules. ESP (Encapsulating Security Payload) and AH (Authentication Header), each one of those modules provides some of the CIA parts above. ESP usually handles all of the confidentiality and AH usually handles authentication and integrity. Now that is already cool by itself, and as if this wasn't enough IP Sec also operates on the Network Layer of the OSI model, which allows it to be implemented independent of the application used, and since it is on the Network Layer it can hide Network Layer information which would prevent a lot of traffic analysis.

One other point that I think should be mentioned here is that there are 2 modes for IPSec data transmission:

Transport Mode – Data is protected.

Tunnel Mode – IP Header is also encrypted. In Tunnel mode a whole new header is applied, IPSec would encrypt the original TCP/IP Header, and would shove a new one on the packet, personally I think this is extremely cool.

So this means that although IPSec operates on Network Layer it only encrypted Network Layer data if it operated in Tunnel Mode. And this is one of the beauties of IP Sec, that it is so flexible.

Authentication Header

Ok, here is how Authentication Header breaks down:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Next Header   |  Payload Len  |          RESERVED             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 Security Parameters Index (SPI)               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Sequence Number Field                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+               Authentication Data (variable)                  |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                 ASCII picture taken from the IP AH RFC 2402
```

Next Header – This field indicates the type of payload that this packet is carrying. This is not very important, so I am going to go more in detail about this.

Payload Length – 8 bit field indicated the length of Authentication Header.

SPI Security Parameters Index – 32 bit arbitrary value that in combination with AH identified the Security Association (more on this later).

Sequence Number – Sequence number of the packet, must always be present.

Authentication Data – Field that contains ICV (Integrity Check Value). The field must be 32 bits, sometimes padding is used to fill up empty space. In more detailed this would be the MAC (Message Authentication Code), this is sort of like a checksum string, a one-way hash usually to be exact. This is used to make sure that the message has not been changed.

Now here is how the packet looks once the AH has been applied in Transport mode:

```
| Original Header | AH | Data |
```

Now in Tunnel mode this would look like

```
| New Header | AH | Original header | Data |
```

And AH would authenticate all of these fields. Now here are the things that Authentication header provides:

header provides.

Data Integrity (a.k.a. Connectionless integrity)

This is provided by the Authentication Data field that is added for the ICV (Integrity Check Value). Usually this is done through a process called a one-way hashing. When a hashing algorithm is run with some data fed into it, it generates a value, which we called the hash, and this process is called hashing. Now what makes it a one-way hash is that this hash can't be reversed. So you wouldn't be able to get the data just by looking at the hash, but you would be able to get the hash by looking at the data.

So say for example that you want to make a one way hash, and you make a code to take each letters, then translates the letters in numbers that correspond with their position in the alphabet then multiply each number by 2 and add 1. Then for a parameter of abc it would work as follows: ((1*2+1))+(2*2+1)+(3*2+1))= 15. Now if you were an attacker, and you didn't know the algorithm that was used, or maybe you knew that you were supposed to multiply by 2, but you didn't know that you were supposed to add 1, how would you get the value of abc? You wouldn't! Now remember that this is a VERY basic example of how one-way hashing works.

So the message would be hashed, the entire message, even the TCP/IP Header, and it would go in Authentication Data. And this would provide Integrity because it would be very difficult to generate the message with the same hash as another one. Some of the hashing protocols that AH supports is HMAC-MD5, and HMAC-SHA-1. SHA-1 generated a longer hash than MD5. Since the IP header data is also hashed it cuts down on spoofing.

Data origin Authentication

Data authentication is very important, like I mentioned above integrity and authentication go hand in hand. The way that AH provides authentication is through 2 mechanisms, that confirm senders validity:

Sealing – This is the hash in the ICV. The hash is calculated with a secret key, so therefore the out come of the hash depends on both the date, and they key that the peers have. So without knowing the key an attacker would not be able to generate the hash.

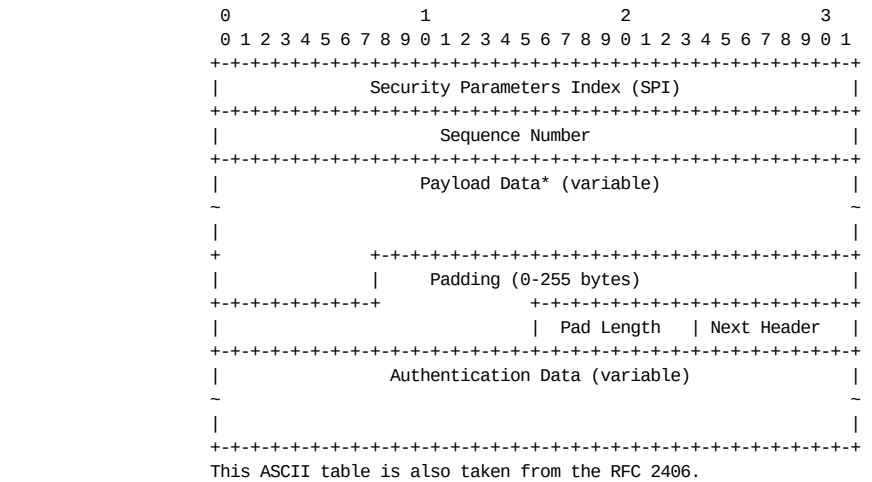
Digital Signature – This makes sure that a person who sent the data is actually who he says he is, also makes sure that when a sender sends a message to the receiver, the sender can't later say that he didn't send the message, this is called non-repudiation.

And protection against replay-attacks

IP Sec has sequence numbers, sequence numbers is the main tool that is used to prevent replay attacks.

Encapsulation Security Payload (ESP)

Now we are getting to the fun stuff. Here is the Packet Format:



This ASCII table is also taken from the RFC 2406.

SPI – Security Parameters Index – 32 bit value identified Security Association for the packet. Ok, I think I will get into the Security Association right now; basically it is sort of like in some weird way like a one-way trust between hosts. Since it is a one-way trust it means it is a uni-directional, what else does it mean? That means that there has to be 2 of them per session. SPI is thru Sequence numbers, and SAs.

Sequence Number – Is always present, and is a "32-bit field contains a monotonically increasing counter value (sequence number)" That's from RFC 2406, and in plain English it

means. That sequence numbers can never be repeated per SA. This would prevent Replay attacks, since a new SA would be needed to be established in order to replay those packets, and even then the timing would be impossible to get. This also means that once the session has been established for a while and the sequence numbers are reaching 2^32 a new SA and a new key has to be established.

Payload Data – This is the data that is carried, the type of this data is also specified in the Next Header field in the AH. But in any case this is the encrypted data that is carried in the packet.

Padding – A lot of encryption algorithms are set up so that they are designed to throw the output in blocks. Sometimes this might be a 4-byte blocks. So the padding is used to make sure that if there is not enough data to fill up the packet the 4-byte blocks will take up space as padding, so that the data will be the size required by the algorithm.

Authentication Data – This field contains the ICV (Integrity Check Value), in other words this is the hash that was computed of the whole ESP packet minus the Authentication Data field.

Now the header placement for ESP is very similar to AH. You have your tunnel and your transport mode and depending on which mode you pick ESP either hides the Original Header or doesn't.

```
     BEFORE APPLYING ESP
----------------------------
|orig IP hdr  |     |      |
|(any options)| TCP | Data |
----------------------------
     AFTER APPLYING ESP
-------------------------------------------------
|orig IP hdr  | ESP |     |      |   ESP   | ESP|
|(any options)| Hdr | TCP | Data | Trailer |Auth|
-------------------------------------------------
                    |<----- encrypted ---->|
               |<------ authenticated ----->|
```

This little chart is also taken from the RFC 2406, I hope these guys don't mind.

As the chart shows that the ESP kind of wraps all the data in itself. And although this is quite a happy thought, this is still only the transport mode. The tunnel mode looks more like:

```
-----------------------------------------------------------
| new IP hdr* |     | orig IP hdr*  |   |    | ESP   | ESP|
|(any options)| ESP | (any options) |TCP|Data|Trailer|Auth|
-----------------------------------------------------------
                    |<--------- encrypted ---------->|
               |<------------ authenticated --------&#61664;|
     This chart is also taken from RFC 2406
```

This ESP in tunnel mode also encrypts the original IP header, which is very cool. You can also see that everything is in the original packet in encrypted and authenticated You can also see that the ESP Authentication data is not encrypted, this is because the receiver checks to make sure that the hash actually matches before it begins the process of decrypting the packet.

Diffie-Hellman algorithm

This encryption algorithm is used to prevent a person who has been sniffing the session from the beginning from getting the key, and understanding what is going on in the session. Here is how it works, this next bit of the steps is taken from an excellent guide on IP Sec, has everything you need to know about it, and I highly recommend checking it out once you are done with this one.

"Alice and Bob agree on a large prime n and on g such that g is primitive modulo n. These two integers arepublic.

1. Alice randomly chooses a large integer a, which she keeps secret, and computes her public value A=ga mod n. Bob does the same thing and generates b and B=gb mod n.
2. Alice sends A to Bob; Bob sends B to Alice.
3. Alice computes KAB=Ba mod n; Bob computes KBA=Ab mod n. KAB=KBA=gab mod n is the secret shared by Alice and Bob.

A person who listens to the communication knows g, n, A=ga mod n and B=gb mod n, which does not enable him to compute gab mod n: for that, he would first have to compute the discrete logarithm of A or B so as to recover a or b. "

That is cool; however Diffie-Hellman algorithm is still vulnerable to Man in the middle attacks, I will talk about this later. But now I want to look at the overall security that IP Sec can provide for you.

IP Sec and what it can do for YOU!

Lets quickly outline different types of attacks that an intruder can carry out against your network, the following list is of attack techniques, and things we just want to avoid, and then it lists how IP Sec deals with that issue most of you will already be familiar with that list, but it's a good review:

Man in the middle attack – This attack would most likely happen on a LAN, the intruder would modify the switch's forwarding table, so that when a session is established between server1 to workstation1 the session is actually established through the intruder's computer. Now this problem is very tricky, first of any information that is sent between the workstation and the server is assumed to be through a secure tunnel, however it is not, and the attacker intercepts all information. This would be mainly combated with integrity and authentication.

Integrity and Authentication – Like I mentioned above, IP Sec would generate the hash based on the key of the sender, and based on the data. So therefore if the attacker did not know the key of the sender he would not be able to generate the same hash. And if he generated the hash with his own key, then the receiving computer would not accept the data because the hash wouldn't calculate correctly.

Sniffing – This is the attack when an attacker is receiving every packet that goes by. Now there is nothing you can do to prevent an intruder from sniffing your traffic, but what you can do is you can encrypt your data and the intruder wouldn't know what you are sending. So the problem of sniffing is combated with confidentiality.

Confidentiality – ESP encrypts the data that is sent across the network, and then hashes that data. This would prevent a person who is listening from understanding the data that travels across the network. However a lot of people (including me up until now) are confused because they think that if an attacker was listening since the beginning of the session the keys would be intercepted. However above I outlined how Diffie-Hellman works, and that would solve this issue.

Traffic Analysis – This happens when an attacker is running some sort of traffic analyzing tool on your network. Although the data is encrypted the attacker can still see what types of packets are being sent, who is sending them, and who is receiving them. IP Sec would combat this mainly with confidentiality, in the ESP or the AH. Look at the following example:

```
PC1 -------- Security Gateway ========== Security Gateway -------- PC2

- transport mode over trusted network
= tunnel mode over un trusted network
```

This means that if someone would be sniffing data on a large scale, like over the internet for example, the IP header would be encrypted in the tunnel, and it would look as if the two security gateways were talking. This would prevent the attacker from finding out which exact hosts were communicating, and therefore would not be able to decode most of the session.

Ok, this covered most of the security features of IP Sec. I hope you have learned a lot of this article, and have enjoyed reading it, as much as I have enjoyed writing it. You can contact me via e-mail for questions and comments at delete852-at-yahoo.com