به نام خدا

# گزارش کار آزمایش سوم آزمایشگاه شبکه‌های کامپیوتری

## فرزان رحمانی – 99521271

## امیر محمد خورشیدی – 99521244

ابتدا تغییرات لازم در فایل lanConfig.py را انجام می‌دهیم.

#!/usr/bin/python

"""

This example shows how to create a Mininet object and add nodes to it manually.

"""

"Importing Libraries"

from mininet.net import Mininet

from mininet.node import Controller

from mininet.cli import CLI

from mininet.log import setLogLevel, info

"Function definition: This is called from the main function"

def firstNetwork():

    "Create an empty network and add nodes to it."

    net = Mininet()

    info( '\*\*\* Adding controller\n' )

    net.addController( 'c0' )

    info( '\*\*\* Adding hosts\n' )

    h1 = net.addHost( 'h1', ip='10.10.14.1/24')

    h2 = net.addHost( 'h2', ip='10.10.24.2/24')

    h3 = net.addHost( 'h3', ip='10.10.34.3/24')

    h4 = net.addHost( 'h4', ip='10.10.14.4/24')

    info( '\*\*\* Adding switch\n' )

    s14 = net.addSwitch( 's14' )

    s24 = net.addSwitch( 's24' )

    s34 = net.addSwitch( 's34' )

    info( '\*\*\* Creating links\n' )

    net.addLink( h1, s14 )

    net.addLink( h4, s14 )

    net.addLink( h2, s24 )

    net.addLink( h4, s24 )

    net.addLink( h3, s34 )

    net.addLink( h4, s34 )

    h4.cmd('ip addr add 10.10.24.4/24 dev h4-eth1')

    h4.cmd('ip addr add 10.10.34.4/24 dev h4-eth2')

    h4.cmd('echo 1 > /proc/sys/net/ipv4/ip\_forward')

    h3.cmd('echo 1 > /proc/sys/net/ipv4/ip\_forward')

    info( '\*\*\* Starting network\n')

    net.start()

    h1.cmd('ip route add default via 10.10.14.4')

    h2.cmd('ip route add default via 10.10.24.4')

    h3.cmd('ip route add default via 10.10.34.4')

    "This is used to run commands on the hosts"

    info( '\*\*\* Starting terminals on hosts\n' )

    h1.cmd('xterm -xrm "XTerm.vt100.allowTitleOps: false" -T h1 &')

    h2.cmd('xterm -xrm "XTerm.vt100.allowTitleOps: false" -T h2 &')

    h3.cmd('xterm -xrm "XTerm.vt100.allowTitleOps: false" -T h3 &')

    h4.cmd('xterm -xrm "XTerm.vt100.allowTitleOps: false" -T h4 &')

    info( '\*\*\* Running the command line interface\n' )

    CLI( net )

    info( '\*\*\* Closing the terminals on the hosts\n' )

    h1.cmd("killall xterm")

    h2.cmd("killall xterm")

    h3.cmd("killall xterm")

    h4.cmd("killall xterm")

    info( '\*\*\* Stopping network' )

    net.stop()

"main Function: This is called when the Python file is run"

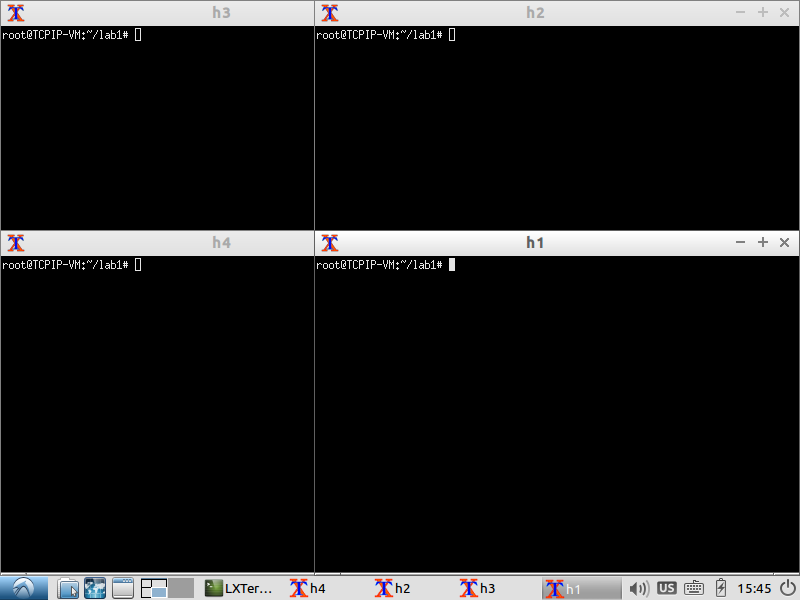
if \_\_name\_\_ == '\_\_main\_\_':

    setLogLevel( 'info' )

    firstNetwork()

سپس با دستور زیر شبکه شبیه سازی شده را بالا می‌آوریم:

sudo python lanConfig.py



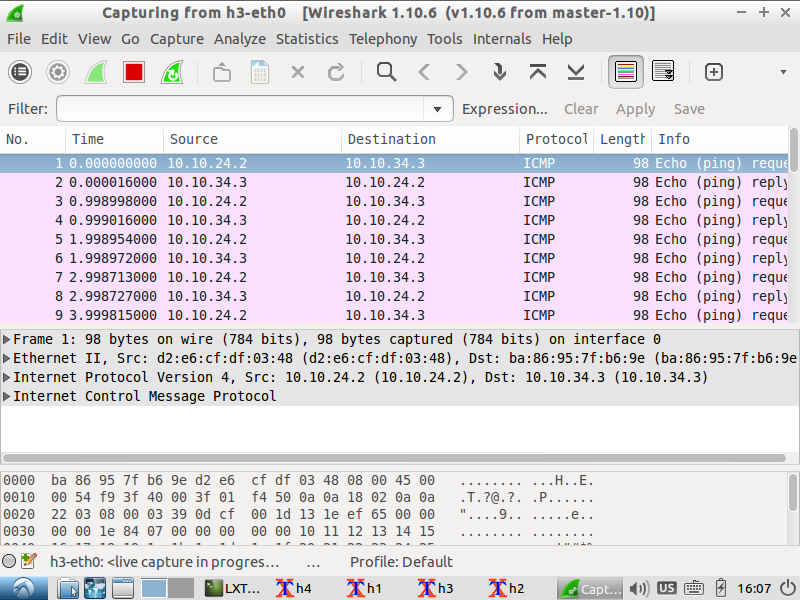
### سوال اول)

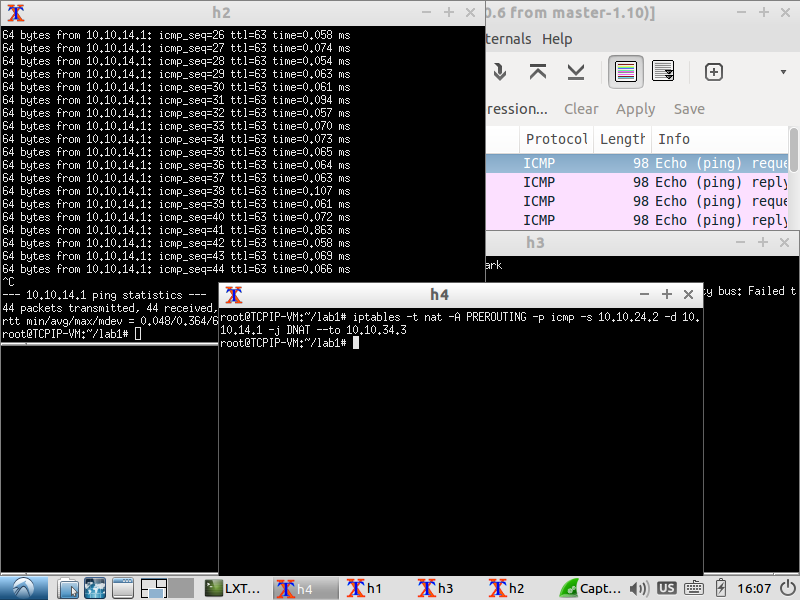
این دستور را در h4 اجرا می‌کنیم: (این دستور بسته های از نوع ICMP(Ping) که از هاست Alice می آیند و قرار است به هاست Bank بروند را به هاست Attacker ارسال میکند.)

Iptables -t nat -A PREROUTING -p icmp -s 10.10.24.2 -d 10.10.14.1 -j DNAT -to 10.10.34.3



اکنون Wireshark را روی هاست H3 یعنی Attacker اجرا میکنیم، سپس از هاست H2 یعنی Alice هاست H1 یعنی Bank را ping میکنیم. (ping 10.10.14.1)





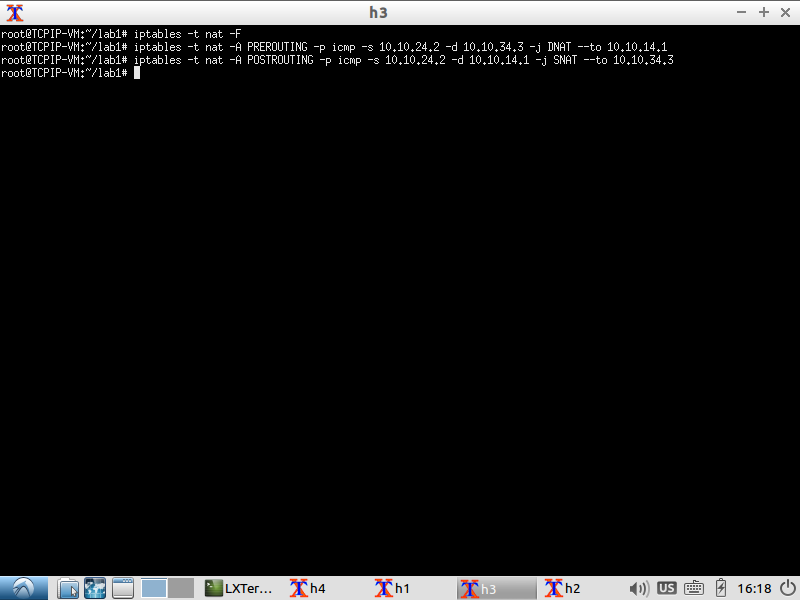
دو بسته Request و Reply از نوع ICMP در H3 یا همان Attacker دیده می‌شود.

### سوال دوم)

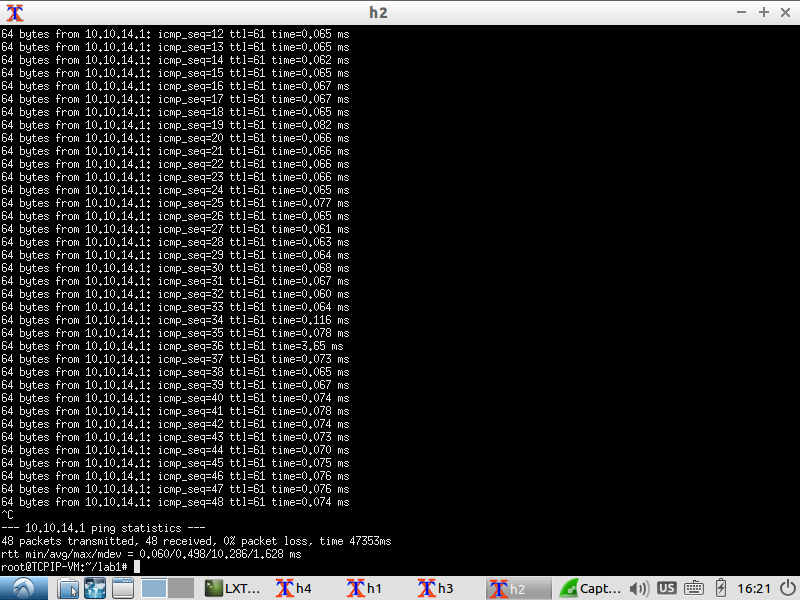
این دستور ها را در h3 این دستور را اجرا می‌کنیم: (روی هاست H3 یعنی Attacker باید قوانینی روی iptables تعریف کنیم که بسته ارسالی از Alice به Attacker را با نام Alice به Bank ارسال کنیم. و پاسخ آن را از Bank بگیریم و به Attacker ارسال کنیم سپس از Attacker با نام Bank به Alice ارسال کنیم.)

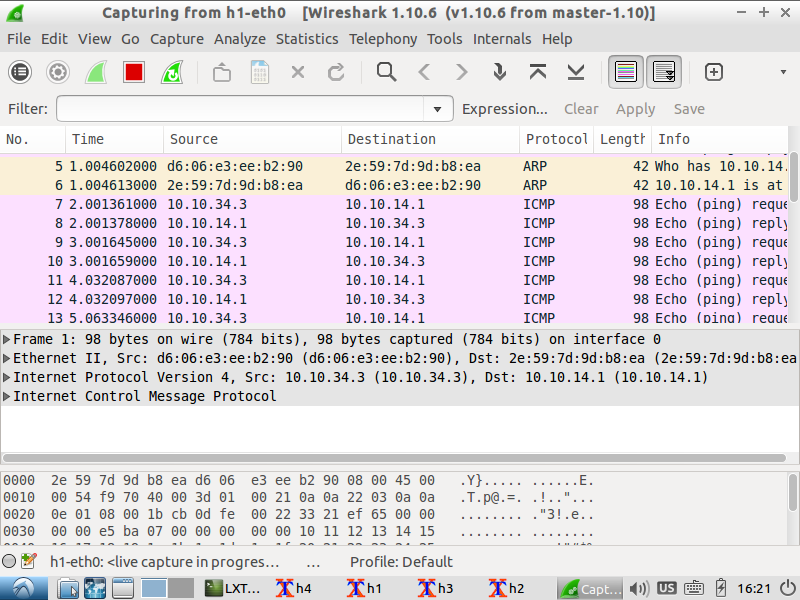
Iptables -t nat -A PREROUTING -p icmp -s 10.10.24.2 -d 10.10.34.3 -j DNAT -to 10.10.14.1

Iptables -t nat -A POSTREROUTING -p icmp -s 10.10.24.2 -d 10.10.14.1 -j SNAT -to 10.10.34.3



حال Wireshark را اجرا میکنیم و از هاست Alice ، بانک را پینگ می‌کنیم. (ping 10.10.14.1)



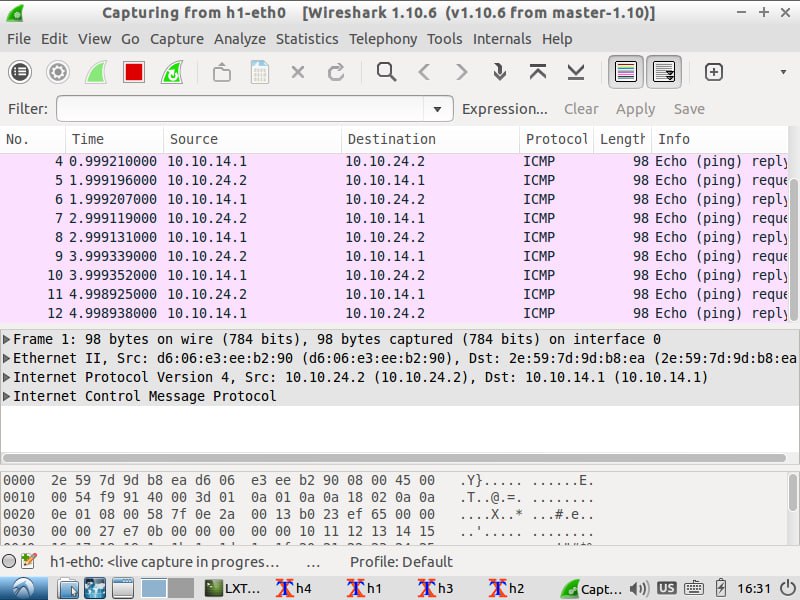


همانطور که مشاهده می‌شود هاست H3 یعنی Attacker توانسته ارتباط Alice و بانک را رصد کند بدون اینکه ردی از خود بر جای بگذارد.

این دستور را در h4 اجرا می‌کنیم:

Iptables -t nat -A POSTREROUTING -p icmp -s 10.10.34.3 -d 10.10.14.1 -j SNAT -to 10.10.24.2





### سوال سوم)

خیر. همانطور که در بخش های قبل دیدیم می‌بایست تغییراتی روی هاست خودمان (Attacker) انجام دهیم تا RPF با Drop کردن بسته ها جلوی حمله را نگیرد.

همچنین چون بسته های بانک را به h3 می‌فرستیم باید جداول h3 را هم تغییر بدهیم تا بسته ها به h1 بروند.

به علاوه اگر دو دستور iptables روی هاست Attacker را نمی‌زدیم، بانک و آلیس متوجه حمله ما می‌شدند زیرا IP متخاصم در پکت‌ها دیده می‌شد.

### سوال چهارم)

* با استفاده از rtt: مقدار زیاد آن، نشان می‌دهد که طول زمان رفت و برگشت بسته زیاد شده است پس مسیر عوض شده است یا به عبارتی طولانی تر شده است.
* با استفاده ازttl : با عبور از هر روتر مقدار آن یک واحد کاهش میابد پس اگر مسیر عوض شده باشد یا طولانی تر شده باشد مقدار ttl از حالت عادی کمتر میشود.
* همچنین میتوان از طریق لاگهای درون Router خود متوجه این حمله شد. زیرا IP متخاصم درون Router قابل مشاهده است. همچنین Routerها و مودم های جدید دارای قابلیت هایی هستند که جلوی IP Spoofing یاARP Spoofing را می‌گیرند.

پایان