A picture containing graphical user interface

Description automatically generated**Purpose**

The local area network is the most vulnerable part of the network, open to the most direct access to the devices. ARP spoofing is an attack on local area networks that exploits this vulnerability by swapping the hardware identification of devices, routing the traffic through a monitored path. This allows threat actors to steal data passing through the network, potentially including sensitive passwords and confidential data. The exploration of this network attack is useful to understanding the tools available to potential threat actors.

**Background Information**

ARP spoofing is a layer two attack that interrupts the transmission of data across a local area network. ARP spoofing puts a threat actor between two devices, allowing the threat actor to spy on all data that passes through the points.

On a local area network, there are two address by which devices can identify each other: IP addresses and MAC addresses. IP addresses are used to define and route traffic across networks. These addresses are implemented in software and are easily changeable. MAC addresses are used to direct traffic in the local area network to specific devices. MAC addresses are implemented in hardware and are much more specific to the device. When data is sent across a local area network, an IP address can identify that the destination is within the same network, but the IP address cannot determine the physical device to send the data to. Since MAC addresses determine the physical device to send it to, the devices need a way to convert IP addresses into MAC addresses for transmission from device to device. ARP, which stands for address resolution protocol, does this function for IPv4 networks to convert software defined address to hardware defined addresses.

ARP functions by sending a packet with a broadcast destination MAC address and the target IP address. This effectively asks every physical device whether they have the target IP address. When a device receives this message seeing that it is targeted towards their IP address, they return a packet to the source device using their own MAC address. Then the source device saves the MAC address returned and saves the MAC address and IP address pair in an ARP table. This process is like knowing there is a person in the room that goes by the name John and asking everyone in the room whether their name is John. When you meet someone named John, they tell you their name and you write down that they are wearing a red hat. The next time someone asks you to give a pencil to John, you now know to hand it off to someone wearing a red hat.

When devices are added to the network, they can broadcast a message to all devices on the network using their MAC address and IP address so that all devices can receive an ARP address pairing. This is the device’s way of introducing themselves when joining the network. This way of advertisement is called gratuitous ARP messages, and are the main source of vulnerability. Because new devices are advertising their own ARP pairing, a threat actor can advertise their MAC address as using the IP address of another device. This makes it so that all data that is destined for that IP address will be sent towards the physical address of the threat actor. This exploitation of gratuitous ARP messages to advertise false ARP information can be scaled to make all traffic within a local area network have to flow through the threat actor, effectively allowing them to steal and spy on data.

To launch cyberattacks, the most used operating system is Kali Linux. Linux is used because the open source tools have the best compatibility and flexibility for networking attacks. Kali Linux is just a distribution of the Linux operating system that comes with many preinstalled networking tools such as WireShark, a program to read the data of packets transmitted, and Ettercap, another ARP spoofing utility. The process for ARP spoofing requires using the ARPspoof utility from the dsniff software package. This will create gratuitous ARP messages to plant false ARP entries into other devices. The Kali machine is also configured to forward packets as if they are a router to enable packets to flow through it. Then, WireShark is used to capture the packets that flow through the threat actor.

A limitation of this attack is that it requires the manipulation of ARP entries. With the introduction of IPv6, ARP has been replaced by a new protocol known as neighbor discovery. As a result, ARP spoofing is only effective against the aging IPv4 networks.

**Lab Summary**

Three routers were connected in a line, with one edge router representing a typical home network and the other edge router using a loopback address to represent a remote network. IPv4 addresses were configured for all networks. OSPF was used between the routers for dynamic routing. The home network includes a router that uses NAT overloading (PAT) and serves as a DHCP server. The home network contains two hosts, one Windows host and a Kali Linux host on the VirtualBox virtual machine acting as the threat actor. The dsniff package’s arpspoof utility was used as well as Wireshark as a protocol analyzer.

**Lab Commands**

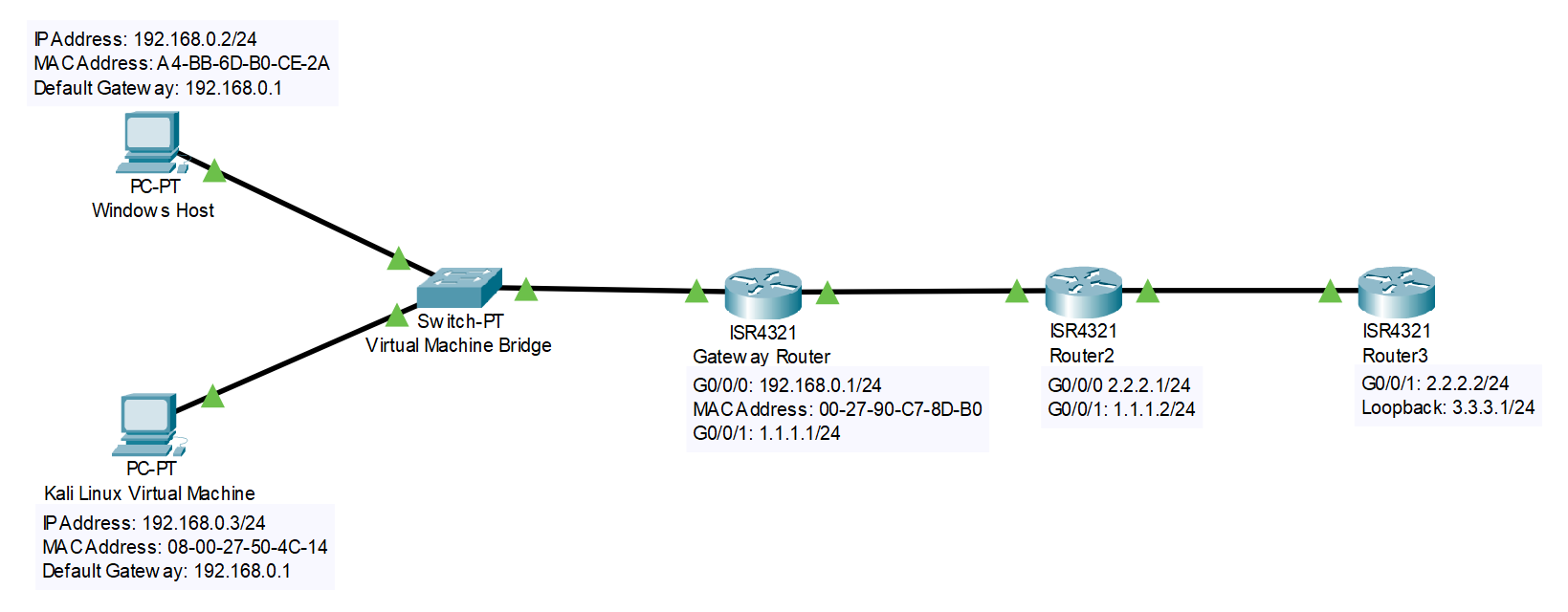
**arpspoof -t <victim IP address> <spoofed IP address>**

The -t flag tells the arpspoof utility to specify a victim to send a gratuitous ARP message. The victim IP address is the IP address of the device to intercept data from. The spoofed IP address is the IP address that the threat actor wants to pose as, typically, the IP address of another device to intercept data from.

**echo 1 > /proc/sys/net/ipv4/ip\_forward**

Command to turn on IPv4 forwarding on the virtual machine which makes it forward packets like a router. Must be run from the root directory.

**Network Diagram**



**Configurations**

**Gateway Router**

**Gateway#show running-config**

Current configuration : 1729 bytes

version 15.5

service timestamps debug datetime msec

service timestamps log datetime msec

no platform punt-keepalive disable-kernel-core

hostname Gateway

boot-start-marker

boot-end-marker

vrf definition Mgmt-intf

address-family ipv4

exit-address-family

address-family ipv6

exit-address-family

no aaa new-model

ip dhcp excluded-address 192.168.0.1

ip dhcp pool HOME\_NETWORK

network 192.168.0.0 255.255.255.0

dns-server 8.8.8.8

default-router 192.168.0.1

domain-name home.network.com

subscriber templating

multilink bundle-name authenticated

license udi pid ISR4321/K9 sn FDO214333H6

spanning-tree extend system-id

redundancy

mode none

vlan internal allocation policy ascending

interface GigabitEthernet0/0/0

ip address 192.168.0.1 255.255.255.0

ip nat inside

negotiation auto

no shutdown

interface GigabitEthernet0/0/1

ip address 1.1.1.1 255.255.255.0

ip nat outside

negotiation auto

no shutdown

interface Serial0/1/0

no ip address

shutdown

interface Serial0/1/1

no ip address

shutdown

interface GigabitEthernet0

vrf forwarding Mgmt-intf

no ip address

shutdown

negotiation auto

interface Vlan1

no ip address

shutdown

router ospf 10

network 1.1.1.0 0.0.0.255 area 0

network 192.168.0.0 0.0.0.255 area 0

ip nat inside source list HOME\_ACL interface GigabitEthernet0/0/1 overload

ip forward-protocol nd

no ip http server

no ip http secure-server

ip tftp source-interface GigabitEthernet0

ip access-list standard HOME\_ACL

permit 192.168.0.0 0.0.0.255

control-plane

line con 0

stopbits 1

line aux 0

stopbits 1

line vty 0 4

login

end

**Router 2**

**R2#show running-config**

Current configuration : 1283 bytes

version 15.5

service timestamps debug datetime msec

service timestamps log datetime msec

no platform punt-keepalive disable-kernel-core

hostname R2

boot-start-marker

boot-end-marker

vrf definition Mgmt-intf

address-family ipv4

exit-address-family

address-family ipv6

exit-address-family

no aaa new-model

subscriber templating

multilink bundle-name authenticated

license udi pid ISR4321/K9 sn FDO214420HY

spanning-tree extend system-id

redundancy

mode none

vlan internal allocation policy ascending

interface GigabitEthernet0/0/0

ip address 2.2.2.1 255.255.255.0

negotiation auto

no shutdown

interface GigabitEthernet0/0/1

ip address 1.1.1.2 255.255.255.0

negotiation auto

no shutdown

interface Serial0/1/0

no ip address

interface Serial0/1/1

no ip address

interface GigabitEthernet0

vrf forwarding Mgmt-intf

no ip address

negotiation auto

interface Vlan1

no ip address

router ospf 10

network 1.1.1.0 0.0.0.255 area 0

network 2.2.2.0 0.0.0.255 area 0

ip forward-protocol nd

no ip http server

no ip http secure-server

control-plane

line con 0

stopbits 1

line aux 0

stopbits 1

line vty 0 4

login

end

**Router 3**

**R3#show running-config**

Current configuration : 1412 bytes

version 15.5

service timestamps debug datetime msec

service timestamps log datetime msec

no platform punt-keepalive disable-kernel-core

hostname R3

boot-start-marker

boot-end-marker

vrf definition Mgmt-intf

address-family ipv4

exit-address-family

address-family ipv6

exit-address-family

no aaa new-model

subscriber templating

multilink bundle-name authenticated

license udi pid ISR4321/K9 sn FDO214420QQ

spanning-tree extend system-id

redundancy

mode none

vlan internal allocation policy ascending

interface Loopback0

ip address 3.3.3.1 255.255.255.0

interface GigabitEthernet0/0/0

ip address 2.2.2.2 255.255.255.0

negotiation auto

no shutdown

interface GigabitEthernet0/0/1

no ip address

shutdown

negotiation auto

interface Serial0/1/0

no ip address

shutdown

interface Serial0/1/1

no ip address

shutdown

interface GigabitEthernet0

vrf forwarding Mgmt-intf

no ip address

shutdown

negotiation auto

interface Vlan1

no ip address

shutdown

router ospf 10

network 2.2.2.0 0.0.0.255 area 0

network 3.3.3.0 0.0.0.255 area 0

ip forward-protocol nd

no ip http server

no ip http secure-server

ip tftp source-interface GigabitEthernet0

control-plane

line con 0

stopbits 1

line aux 0

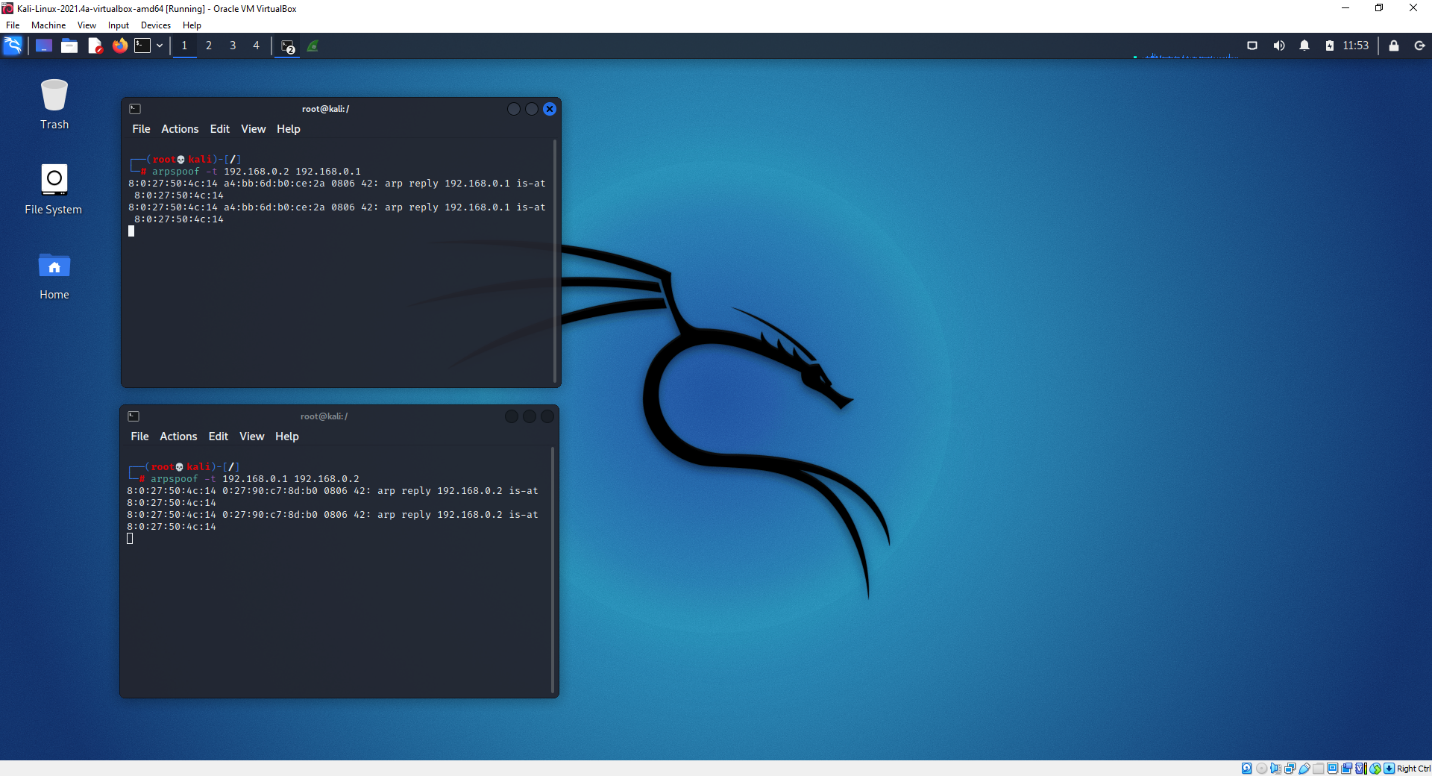
stopbits 1

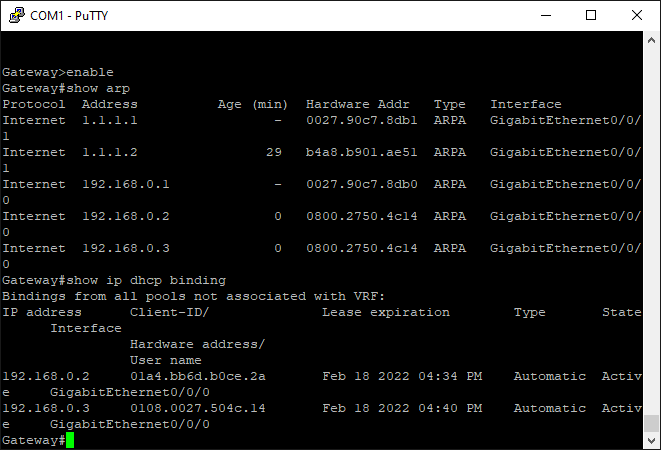
line vty 0 4

login

end

**Verification**





C:\Users\user\Desktop>arp -a

Interface: 192.168.0.2 --- 0x8

Internet Address Physical Address Type

192.168.0.1 08-00-27-50-4c-14 dynamic

192.168.0.3 08-00-27-50-4c-14 dynamic

192.168.0.255 ff-ff-ff-ff-ff-ff static

224.0.0.2 01-00-5e-00-00-02 static

224.0.0.5 01-00-5e-00-00-05 static

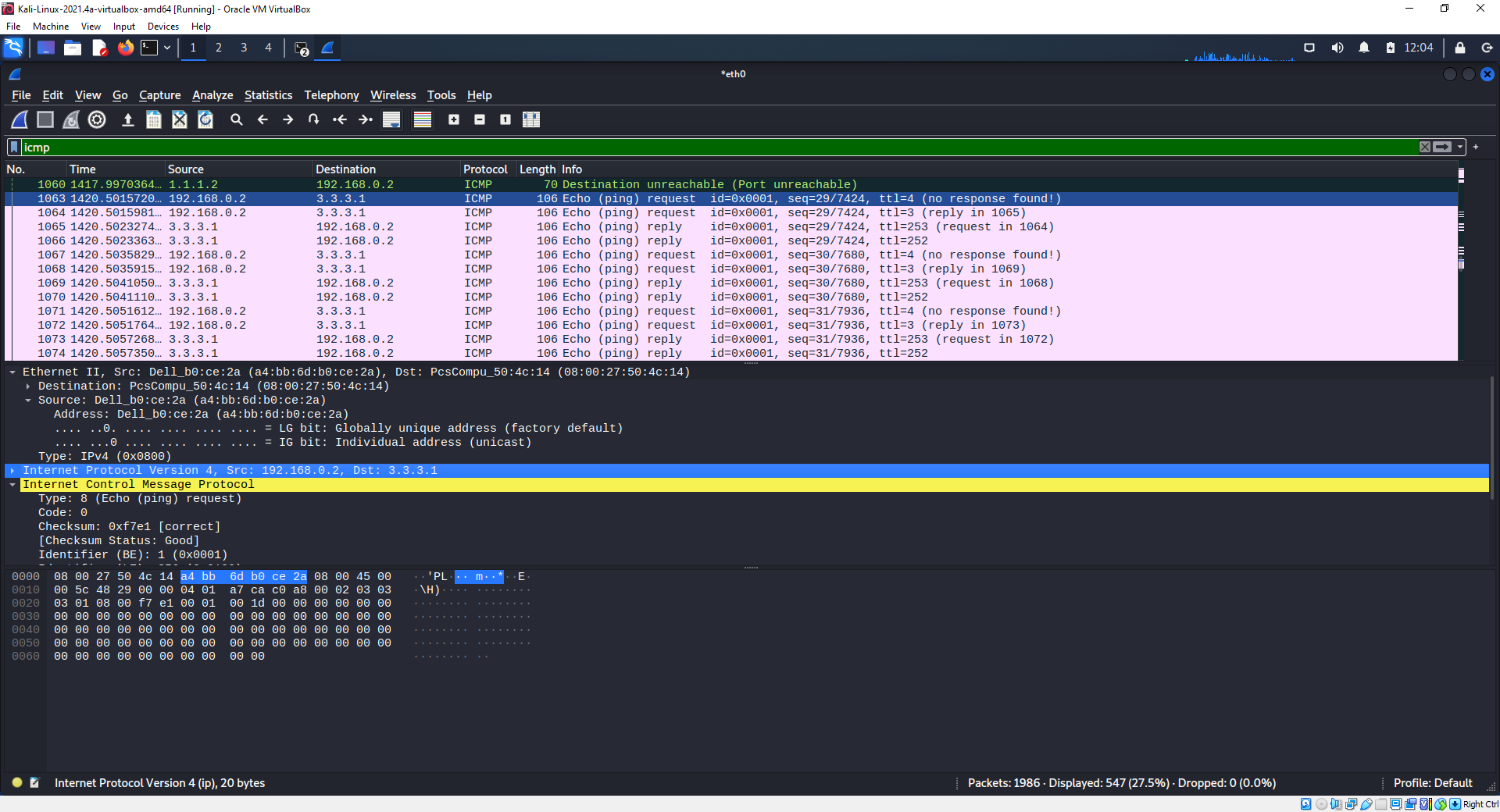
224.0.0.22 01-00-5e-00-00-16 static

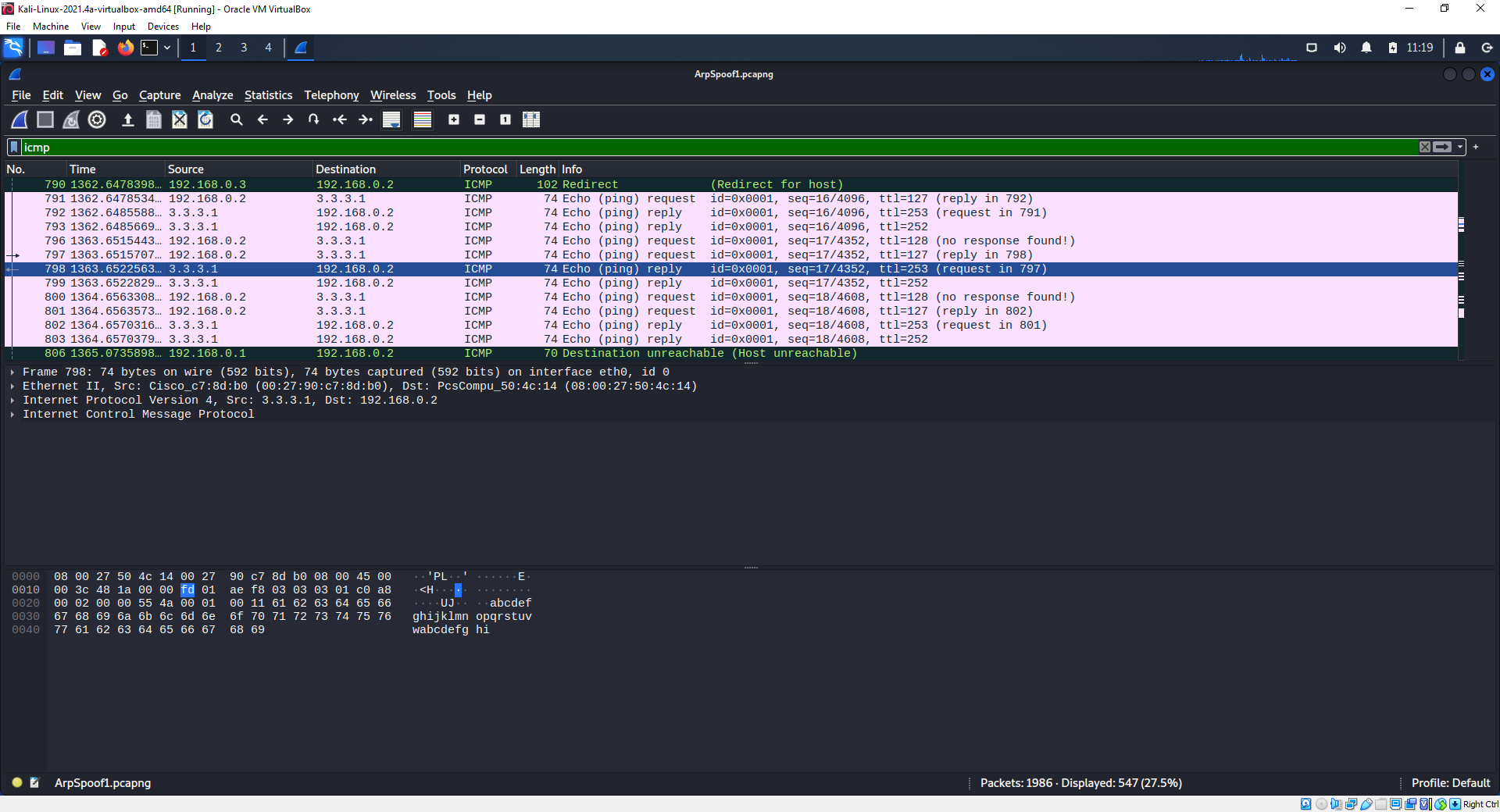
224.0.0.251 01-00-5e-00-00-fb static

224.0.0.252 01-00-5e-00-00-fc static

239.255.255.250 01-00-5e-7f-ff-fa static

255.255.255.255 ff-ff-ff-ff-ff-ff static





**Problems**

The DHCP server was having troubles assigning an IP address to the virtual machine. Turning the wired connection on and off in the virtual machine was tried but had no effect. This was fixed by restarting the virtual machine. A common reason for this is that the virtual machine was not set into bridged mode, meaning that it was not connected to the physical network. Because the virtual machine experienced difficulties while in bridged mode, this is not the cause. The actual cause is unknown and is potentially a bug within VirtualBox.

Another potential bug is that the Kali virtual machine could not ping the Windows host but the Windows host was able to ping the Kali virtual machine. Upon further pings, the Kali virtual machine was also unable to ping the default gateway. However, the Kali virtual machine was able to sniff incoming and outgoing packets as well as ping remote destinations. This problem remains unresolved as it has no impact on the ARP spoofing process however it could be the result of some hidden VirtualBox firewall specifically against echo replies sent back from the local network to the virtual machine.

**Conclusion**

With advanced networking attack tools easily accessible through Kali Linux tools, network security is becoming even more important. ARP spoofing is a dangerous man in the middle attack that effectively steals all desired data on a local area network before it is passed to the internet. By simulating and understanding the actions of a potential threat actor, systems to prevent gratuitous ARP messages from overriding existing ARP table entries can be implemented to increase the security of a local area network. Investigating network attacking tools is one of the most effective ways to preventing network attacks.