

1.

With admin privilege:

```
###[ Ethernet ]###
  dst      = d8:94:03:fa:4b:43
  src      = 9c:fc:e8:06:a3:a6
  type     = IPv4
###[ IP ]###
  version  = 4
  ihl      = 5
  tos      = 0x0
  len      = 60
  id       = 52020
  flags    =
  frag     = 0
  ttl      = 128
  proto    = icmp
  chksum   = 0x35e2
  src      = 10.30.31.125
  dst      = 8.8.8.8
  \options \
###[ ICMP ]###
  type     = echo-request
  code     = 0
  chksum   = 0x4d30
  id       = 0x1
  seq      = 0x2b
  unused   = ''
###[ Raw ]###
  load     = 'abcdefghijklmnopqrstuvwabcdefghi'
```

Without admin privilege:

```

###[ Ethernet ]###
  dst      = d8:94:03:fa:4b:43
  src      = 9c:fc:e8:06:a3:a6
  type     = IPv4
###[ IP ]###
  version  = 4
  ihl      = 5
  tos      = 0x0
  len      = 60
  id       = 52019
  flags    =
  frag     = 0
  ttl      = 128
  proto    = icmp
  chksum   = 0x35e3
  src      = 10.30.31.125
  dst      = 8.8.8.8
  \options \
###[ ICMP ]###
  type     = echo-request
  code     = 0
  chksum   = 0x4d31
  id       = 0x1
  seq      = 0x2a
  unused   = ''
###[ Raw ]###
  load     = 'abcdefghijklmnopqrstuvwabcdefghi'

```

I see no difference on my Windows machine.

1).

```

>>> ls(IP(src='10.0.2.4', dst='www.mit.edu'))
version    : BitField (4 bits)      = 4          ('4')
ihl        : BitField (4 bits)      = None       ('None')
tos        : XByteField             = 0          ('0')
len        : ShortField             = None       ('None')
id         : ShortField             = 1          ('1')
flags      : FlagsField             = <Flag 0 ()> ('<Flag 0 ()>')
frag       : BitField (13 bits)     = 0          ('0')
ttl        : ByteField             = 64         ('64')
proto      : ByteEnumField          = 0          ('0')
chksum     : XShortField            = None       ('None')
src        : SourceIPField          = '10.0.2.4' ('None')
dst        : DestIPField            = Net("www.mit.edu/32") ('None')
options    : PacketListField        = []         ('[]')
>>>

```

2).

```
>>> ls(UDP(sport=5000, dport=53))
sport      : ShortEnumField          = 5000      ('53')
dport      : ShortEnumField          = 53        ('53')
len         : ShortField              = None       ('None')
chksum     : XShortField              = None       ('None')
>>>
```

```
>>> ls(ICMP())
type       : ByteEnumField           = 8          ('8')
code       : MultiEnumField (Depends on 8) = 0          ('0')
chksum     : XShortField              = None       ('None')
id         : XShortField (Cond)       = 0          ('0')
seq        : XShortField (Cond)       = 0          ('0')
ts_ori     : ICMPTimeStampField (Cond) = None       ('13348291')
ts_rx      : ICMPTimeStampField (Cond) = None       ('13348291')
ts_tx      : ICMPTimeStampField (Cond) = None       ('13348291')
gw         : IPField (Cond)           = None       ('0.0.0.0')
ptr        : ByteField (Cond)         = None       ('0')
reserved   : ByteField (Cond)         = None       ('0')
length     : ByteField (Cond)         = None       ('0')
addr_mask  : IPField (Cond)           = None       ('0.0.0.0')
nexthopmtu : ShortField (Cond)        = None       ('0')
unused     : MultipleTypeField (ShortField, IntField, StrFixedLenField) = b''        (b'')
>>>
```

3). stacking IP header over ICMP()

```
>>> (IP(dst="8.8.8.8")/ICMP()).show2()
###[ IP ]###
  version   = 4
  ihl       = 5
  tos       = 0x0
  len       = 28
  id        = 1
  flags     =
  frag      = 0
  ttl       = 64
  proto     = icmp
  checksum  = 0xbeba
  src       = 172.19.0.3
  dst       = 8.8.8.8
  \options  \
###[ ICMP ]###
  type      = echo-request
  code      = 0
  checksum  = 0xf7ff
  id        = 0x0
  seq       = 0x0
  unused    = ''
```

Similary, UDP()

```
>>> ( IP(src="10.0.2.4", dst='8.8.8.8')/UDP(dport=53) ).show2()
###[ IP ]###
  version  = 4
  ihl      = 5
  tos      = 0x0
  len      = 28
  id       = 1
  flags    =
  frag     = 0
  ttl      = 64
  proto    = udp
  checksum = 0x5ebd
  src      = 10.0.2.4
  dst      = 8.8.8.8
  \options \
###[ UDP ]###
  sport = domain
  dport = domain
  len    = 8
  checksum = 0xe360
```

4).

UDP segment:

```
>>> ( IP(src="10.0.2.4", dst='8.8.8.8')/UDP(dport=53) )['UDP'].show2()
###[ UDP ]###
  sport = domain
  dport = domain
  len    = 8
  checksum = 0xe360
>>>
```

ICMP segment

```
>>> ( IP(dst="8.8.8.8")/ICMP() )['ICMP'].show2()
###[ ICMP ]###
  type      = echo-request
  code      = 0
  checksum  = 0xf7ff
  id        = 0x0
  seq       = 0x0
  unused    = ''
```

2. Sniffing Packets

Note, for this lab, I have to use dst instead of pkt[IP].src, as from my machine it is not accessible to 8.8.8.8

```

SNIFFING PACKETS.....
###[ Ethernet ]###
  dst      = d8:94:03:fa:4b:43
  src      = 9c:fc:e8:06:a3:a6
  type     = IPv4
###[ IP ]###
  version  = 4
  ihl      = 5
  tos      = 0x0
  len      = 60
  id       = 52000
  flags    =
  frag     = 0
  ttl      = 128
  proto    = icmp
  chksum   = 0x35f6
  src      = 10.30.31.125
  dst      = 8.8.8.8
  \options \
###[ ICMP ]###
  type     = echo-request
  code     = 0
  chksum   = 0x4d44
  id       = 0x1
  seq      = 0x17
  unused   = ''
###[ Raw ]###
  load     = 'abcdefghijklmnopqrstuvwabcdefghi'

```

3. Sniff function with BPF filters

Applied BPF filters as below:

```

pkt = sniff(filter='icmp and dst 8.8.8.8',prn=print_pkt)
result:

```

```

###[ Ethernet ]###
  dst      = d8:94:03:fa:4b:43
  src      = 9c:fc:e8:06:a3:a6
  type     = IPv4
###[ IP ]###
  version  = 4
  ihl      = 5
  tos      = 0x0
  len      = 60
  id       = 52009
  flags    =
  frag     = 0
  ttl      = 128
  proto    = icmp
  chksum   = 0x35ed
  src      = 10.30.31.125
  dst      = 8.8.8.8
  options  \
###[ ICMP ]###
  type     = echo-request
  code     = 0
  chksum   = 0x4d3b
  id       = 0x1
  seq      = 0x20
  unused   = ''
###[ Raw ]###
  load     = 'abcdefghijklmnopqrstuvwabcdefghi'

```

4. Spoofing ICMP Packets

8.8.8.8 is not reachable from my machine, switching to www.bing.com

From Wireshark no response was observed, as a fake src was in the request package, the response probably has been directed to the fake one rather than the real src.

No.	Time	Source	Destination	Protocol	Length	Info
100	3.549235	10.30.31.125	131.253.33.200	ICMP	60	

```

Frame 100: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface \Device\NPF_{3AC9B254-ED0B-48EC-849D-241CB320FEE0}, id 0
Ethernet II, Src: IntelCor_06:a3:a6 (9c:fc:e8:06:a3:a6), Dst: HewlettP_fa:4b:43 (d8:94:03:fa:4b:43)
Internet Protocol Version 4, Src: 10.30.31.125, Dst: 131.253.33.200
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 28
    Identification: 0x0001 (1)
  > Flags: 0x00
    Fragment Offset: 0
    Time to Live: 64
    Protocol: ICMP (1)
    Header Checksum: 0xab80 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 10.30.31.125
    Destination Address: 131.253.33.200
Internet Control Message Protocol
  Type: 8 (Echo (ping) request)
  Code: 0
  Checksum: 0xf7ff [correct]
  [Checksum Status: Good]
  Identifier (BE): 0 (0x0000)
  Identifier (LE): 0 (0x0000)
  Sequence Number (BE): 0 (0x0000)
  Sequence Number (LE): 0 (0x0000)
  > [No response seen]

```

5. sniffing-then-spoofing

8.8.8 is not reachable, therefore, using '10.5.40.12' instead. So here are the code executed:

```
#!/usr/bin/python3
from scapy.all import *

def spoof_pkt(pkt):
    if ICMP in pkt and pkt[ICMP].type == 8:

        ip = IP(src=pkt[IP].dst, dst=pkt[IP].src, ihl=pkt[IP].ihl)
        icmp = ICMP(type=0, id=pkt[ICMP].id, seq=pkt[ICMP].seq)
        data = pkt[Raw].load
        newpkt = ip/icmp/data

        print("Spoofed Packet.....")
        print("Source IP : ", newpkt[IP].src)
        print("Destination IP :", newpkt[IP].dst)

        send(newpkt, verbose=0)

pkt = sniff(filter='icmp and host 10.5.40.12', prn=spoof_pkt)
```

and result in Wireshark, I observed there is an unreachable error after everything normal ping-response round.

3331	117.729249	10.30.31.125	10.5.40.12	ICMP	102 Destination unreachable (Protocol unreachable)
3382	118.729406	10.30.31.125	10.5.40.12	ICMP	74 Echo (ping) request id=0x0001, seq=95/24326
3383	118.733123	10.5.40.12	10.30.31.125	ICMP	74 Echo (ping) reply id=0x0001, seq=95/24326
3384	118.733589	10.30.31.125	10.5.40.12	ICMP	102 Destination unreachable (Protocol unreachable)
3419	119.742821	10.30.31.125	10.5.40.12	ICMP	74 Echo (ping) request id=0x0001, seq=96/24576
3420	119.745322	10.5.40.12	10.30.31.125	ICMP	74 Echo (ping) reply id=0x0001, seq=96/24576
3421	119.747665	10.30.31.125	10.5.40.12	ICMP	102 Destination unreachable (Protocol unreachable)


```
> Frame 3299: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF_{3AC98254-ED08-48EC-8490-241CB320FEE0}, id 0
> Ethernet II, Src: IntelCor_06:a3:a6 (9c:fc:e8:06:a3:a6), Dst: HewlettP_fa:4b:43 (d8:94:03:fa:4b:43)
✓ Internet Protocol Version 4, Src: 10.30.31.125, Dst: 10.5.40.12
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
    > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
        Total Length: 60
        Identification: 0x39da (14810)
    > Flags: 0x00
        Fragment Offset: 0
        Time to Live: 128
        Protocol: ICMP (1)
        Header Checksum: 0xa53b [validation disabled]
        [Header checksum status: Unverified]
        Source Address: 10.30.31.125
        Destination Address: 10.5.40.12
✓ Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0x4cfe [correct]
    [Checksum Status: Good]
    Identifier (BE): 1 (0x0001)
    Identifier (LE): 256 (0x0100)
    Sequence Number (BE): 93 (0x005d)
    Sequence Number (LE): 23808 (0x5d00)
    [Response frame: 3300]
✓ Data (32 bytes)
    Data: 6162636465666768696a6b6c6d6e6f7071727374757677616263646566676869
    [Length: 32]
```

