SCAPY

packetlife.net

Basic Commands

ls()
List all available protocols and protocol options

lsc()
List all available scapy command functions

conf
Show/set scapy configuration parameters

Constructing Packets

```
# Setting protocol fields
>>> ip=IP(src="10.0.0.1")
>>> ip.dst="10.0.0.2"

# Combining layers
>>> l3=IP()/TCP()
>>> l2=Ether()/l3

# Splitting layers apart
>>> l2.getlayer(1)
<IP  frag=0 proto=tcp |<TCP  |>>
>>> l2.getlayer(2)
<TCP  |>
```

Displaying Packets

```
# Show an entire packet
>>> (Ether()/IPv6()).show()
###[ Ethernet ]###
  dst= ff:ff:ff:ff:ff:ff
  src= 00:00:00:00:00:00
  type= 0x86dd
###[ IPv6 ]###
     version= 6
     tc= 0
     fl= 0
     plen= None
     nh= No Next Header
     hlim= 64
     src= ::1
     dst= ::1

# Show field types with default values
>>> ls(UDP())
sport  : ShortEnumField = 1025 (53)
dport  : ShortEnumField = 53   (53)
len    : ShortField     = None (None)
chksum : XShortField    = None (None)
```

Fuzzing

```
# Randomize fields where applicable
>>> fuzz(ICMP()).show()
###[ ICMP ]###
  type= <RandByte>
  code= 227
  chksum= None
  unused= <RandInt>
```

Specifying Addresses and Values

```
# Explicit IP address (use quotation marks)
>>> IP(dst="192.0.2.1")

# DNS name to be resolved at time of transmission
>>> IP(dst="example.com")

# IP network (results in a packet template)
>>> IP(dst="192.0.2.0/24")

# Random addresses with RandIP() and RandMAC()
>>> IP(dst=RandIP())
>>> Ether(dst=RandMAC())

# Set a range of numbers to be used (template)
>>> IP(ttl=(1,30))

# Random numbers with RandInt() and RandLong()
>>> IP(id=RandInt())
```

Sending Packets

send(pkt, inter=0, loop=0, count=1, iface=N)
Send one or more packets at layer three

sendp(pkt, inter=0, loop=0, count=1, iface=N)
Send one or more packets at layer two

sendpfast(pkt, pps=N, mbps=N, loop=0, iface=N)
Send packets much faster at layer two using tcpreplay

```
>>> send(IP(dst="192.0.2.1")/UDP(dport=53))
.
Sent 1 packets.
>>> sendp(Ether()/IP(dst="192.0.2.1")/UDP(dport=53))
.
Sent 1 packets.
```

Sending and Receiving Packets

sr(pkt, filter=N, iface=N), srp(…)
Send packets and receive replies

sr1(pkt, inter=0, loop=0, count=1, iface=N), srp1(…)
Send packets and return only the first reply

srloop(pkt, timeout=N, count=N), srploop(…)
Send packets in a loop and print each reply

```
>>> srloop(IP(dst="packetlife.net")/ICMP(), count=3)
RECV 1: IP / ICMP 174.143.213.184 > 192.168.1.140
RECV 1: IP / ICMP 174.143.213.184 > 192.168.1.140
RECV 1: IP / ICMP 174.143.213.184 > 192.168.1.140
```

Sniffing Packets

sniff(count=0, store=1, timeout=N)
Record packets off the wire; returns a list of packets when stopped

```
# Capture up to 100 packets (or stop with ctrl-c)
>>> pkts=sniff(count=100, iface="eth0")
>>> pkts
<Sniffed: TCP:92 UDP:7 ICMP:1 Other:0>
```