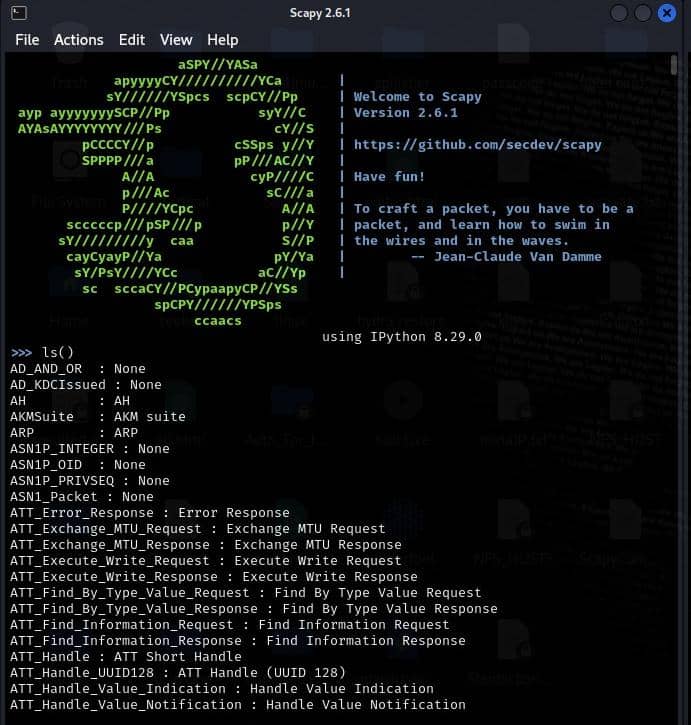
The image contains detailed information about Scapy, focusing on basics, packet crafting, layer fields, and altering packets. Let me transcribe and format the content into clean and presentable text.

Scapy Basics

**To list supported layers: python**

>>> ls()

Some key layers are: arp, ip, ipv6, tcp, udp, icmp.



**To view layer fields, use: python**

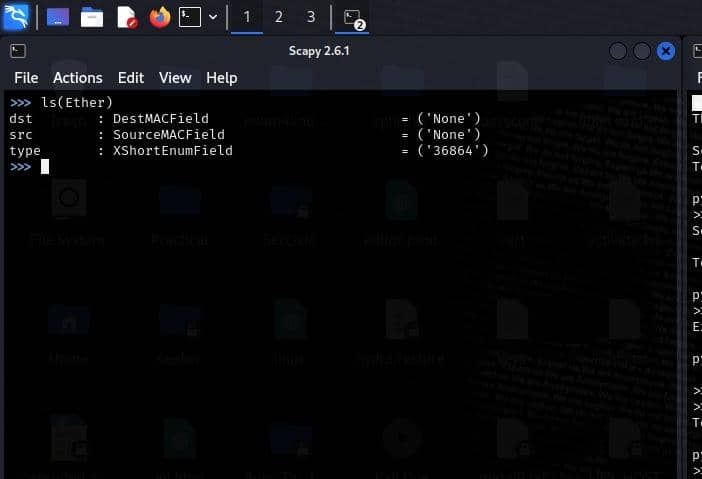
>>> ls(layer)

Example:

python

>>> ls(IPV6)

>>> ls(TCP)

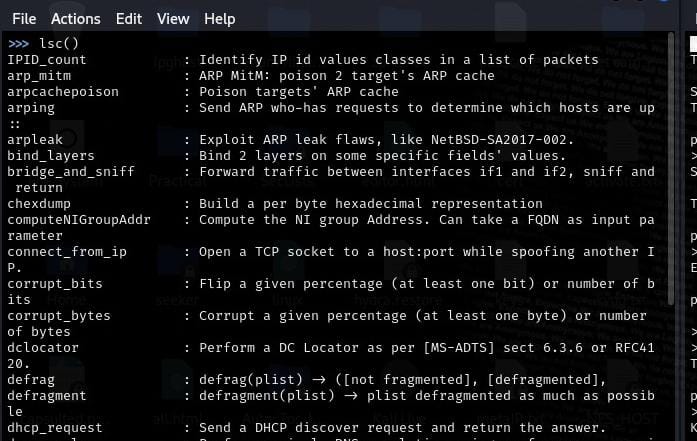


**To list the available commands: python**

>>> lsc()

Key commands for interacting with packets:

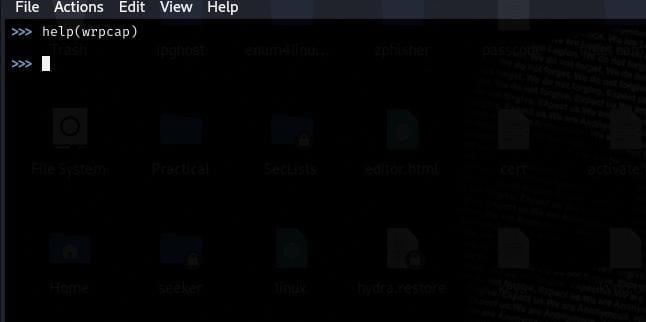
rdpcap, send, sr, sniff, wrpcap.



**Get help with commands: python**

>>> help(command)

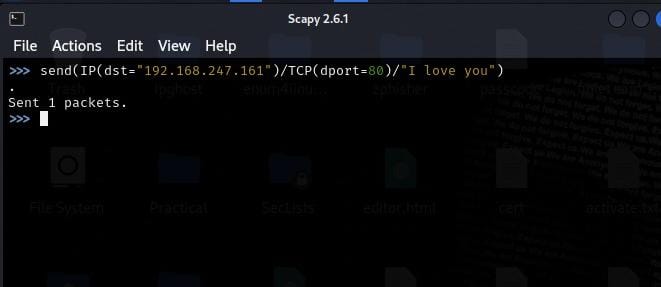
Example: python

>>> help(rdpcap)

**Basic Packet Crafting and Viewing**

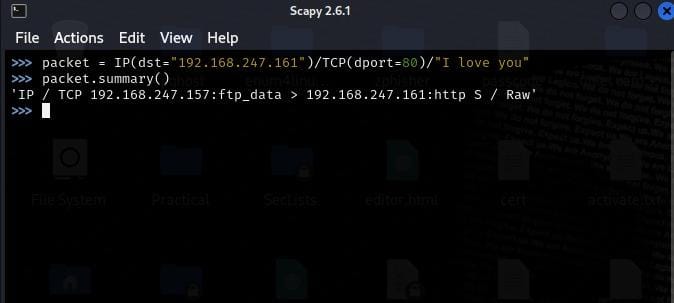
Scapy works with layers. Layers are functions linked together with the / character to construct packets.To build a basic TCP/IP packet with data as the payload: python

>>> send= IP(dst="1.2.3.4")/TCP(dport=22)/"data"

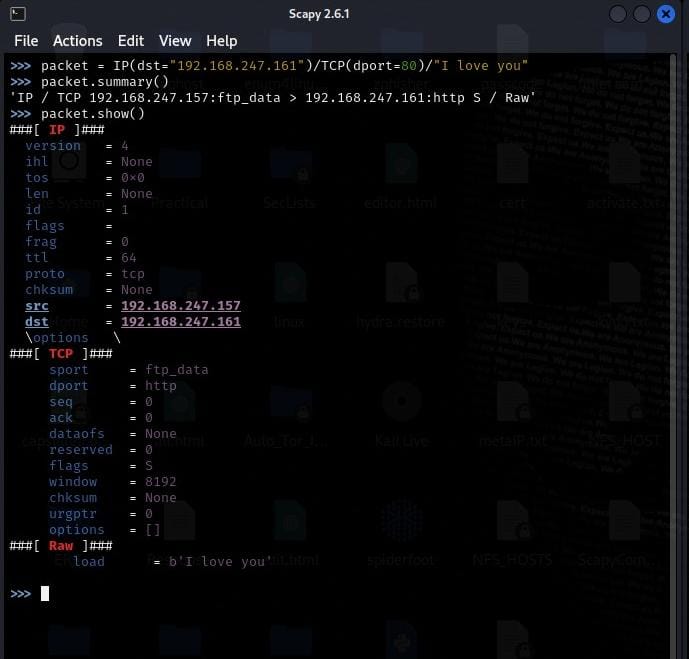
Note: Scapy allows crafting from the highest layer (e.g., TCP) to the lowest (Ethernet), using default field values if layers are omitted. Ensure proper order: Ethernet -> IP -> TCP.

**To get a packet summary: python**

>>> packet.summary()

To get more packet details:

python

>>> packet.show()

**Layer Fields and Default Values**

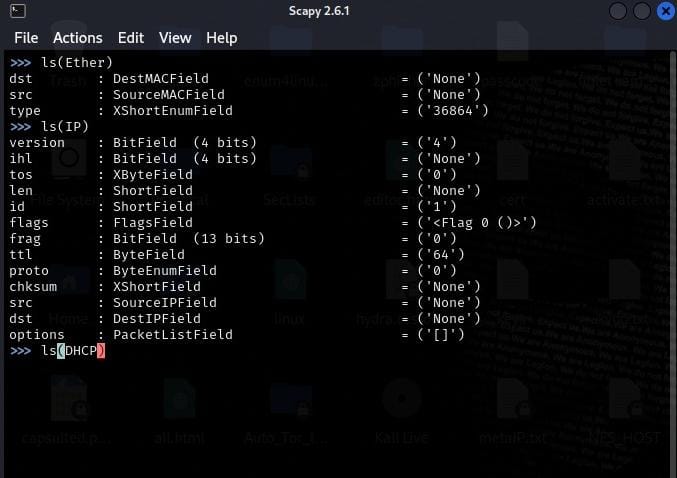
**Ethernet Layer Fields python**

>>> ls(Ether)

Field Type Default Value

dst DestMACField (None)

src SourceMACField (None)

type XShortEnumField (0)

**IPv4 Layer Fields: python**

>>> ls(IP)

Field Type Default Value

version BitField (4)

ihl BitField (None)

tos XByteField (0)

len ShortField (None)

id ShortField (1)

flags FlagsField (0)

frag BitField (0)

ttl ByteField (64)

proto ByteEnumField (0)

chksum XShortField (None)

src Emph (None)

dst Emph (None)

options PacketListField ([])

TCP Layer Fields

python

>>> ls(TCP)

Field Type Default Value

sport ShortEnumField (20)

dport ShortEnumField (80)

seq IntField (0)

ack IntField (0)

dataofs BitField (None)

reserved BitField (0)

flags FlagsField (2)

window ShortField (8192)

chksum XShortField (None)

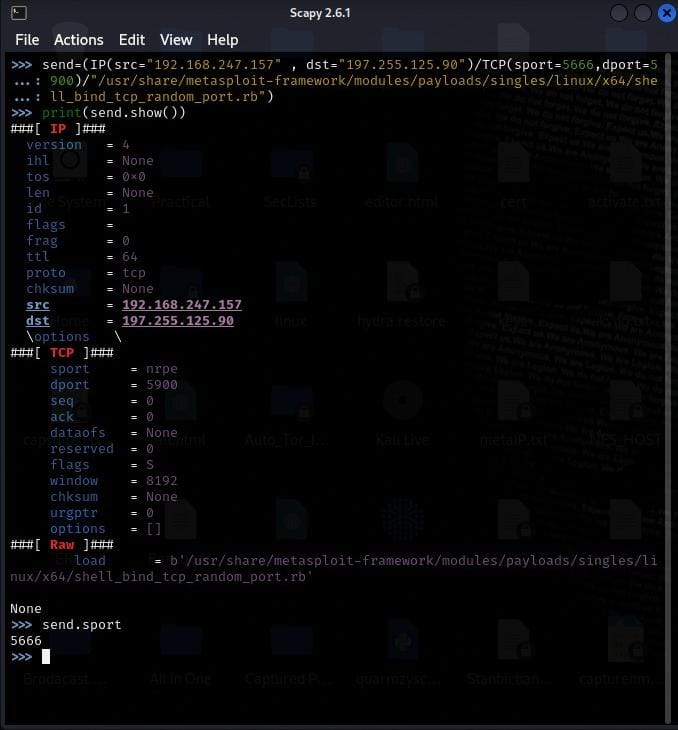
urgptr ShortField (0)

options TCPOptionsField (())

Altering Packets

**Packet layer fields are Python variables and can be modified.**

Example Packet: python

>>> send = IP(dst="10.10.10.50")/TCP(sport=80)

**Viewing a Field's Value (e.g., sport):**

python

>>> send.sport

Setting the Source Port:

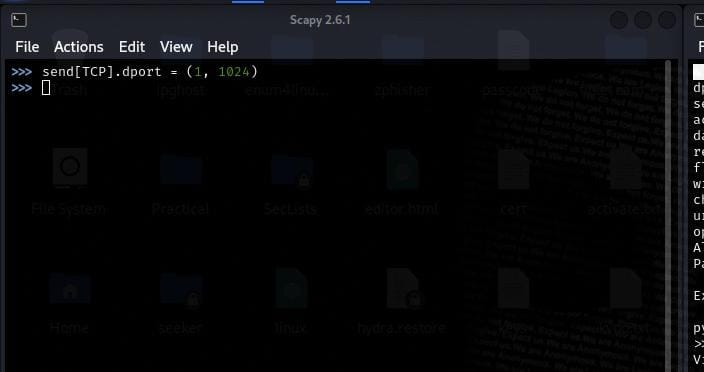
python

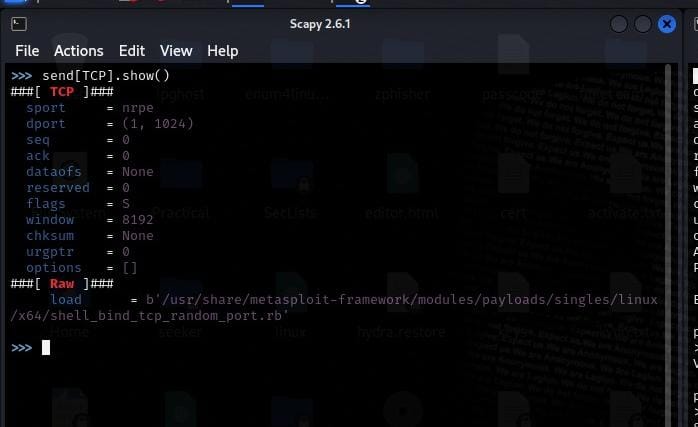
>>> send.sport = 5666

Setting Port Ranges:

Python

>>> send[TCP].dport = (1, 1024)

Setting a List of Ports:



python

>>> send[TCP].dport = [22, 80, 445]

Setting TCP Flags (Control Bits):

python

>>> send[TCP].flags = "SA"

>>> send[TCP].flags

18

>>> send.sprintf("%TCP.flags%")

'SA'

Setting Destination IP Address(es):python

>>> send[IP].dst = "1.2.3.4"

>>> send[IP].dst = ["1.2.3.4", "2.3.4.5", "5.6.7.8"]

Using CIDR:

python

>>> send[IP].dst = "1.2.3.4/16"

How to the output into a pcap or pcang file

>>>wrpcap(“capture.pcap” , send)