

PACKET_SNIFFER

- **Capture** data flowing through an interface.
- **Filter** this data.
- Display interesting information such as:
 - Login info (usernames & **passwords**).
 - Visited **websites**.
 - Images.
 - ...etc



PACKET_SNIFFER

CAPTURE & FILTER DATA

- **scapy** has a sniffer function.
- Can capture data sent to/from **iface**.
- Can call a function specified in **prn** on each packet.

Syntax:

```
import scapy.all as scapy
```

```
scapy.sniff(iface=[interface], prn=[call back function])
```

PACKET_SNIFFER

CAPTURE & FILTER DATA

- `scapy` has a sniffer function.
- Can capture data sent to/from `iface`.
- Can call a function specified in `prn` on each packet.

Syntax:

```
import scapy.all as scapy
scapy.sniff(iface=[interface], prn=[call back function])
```

PACKET_SNIFFER

FILTERING DATA



- Each packet contains a number of layers.
- Each layer contains a number of fields.
- Fields contain data (possibly interesting data).

Assuming packet is a variable that contains a packet:

```
packet.show()           #shows all layers, fields and values  
print(packet[layer_name]) #prints fields & values for given layer  
print(packet[layer_name].field_name) #prints value in given field
```


ARP_SPOOF + PACKET_SNIFFER

- Target a computer on the same network.
- arp_spoof to redirect flow of packets (become **MITM**).
- Packet_sniffer to see **URLs, usernames and passwords** sent by target.

