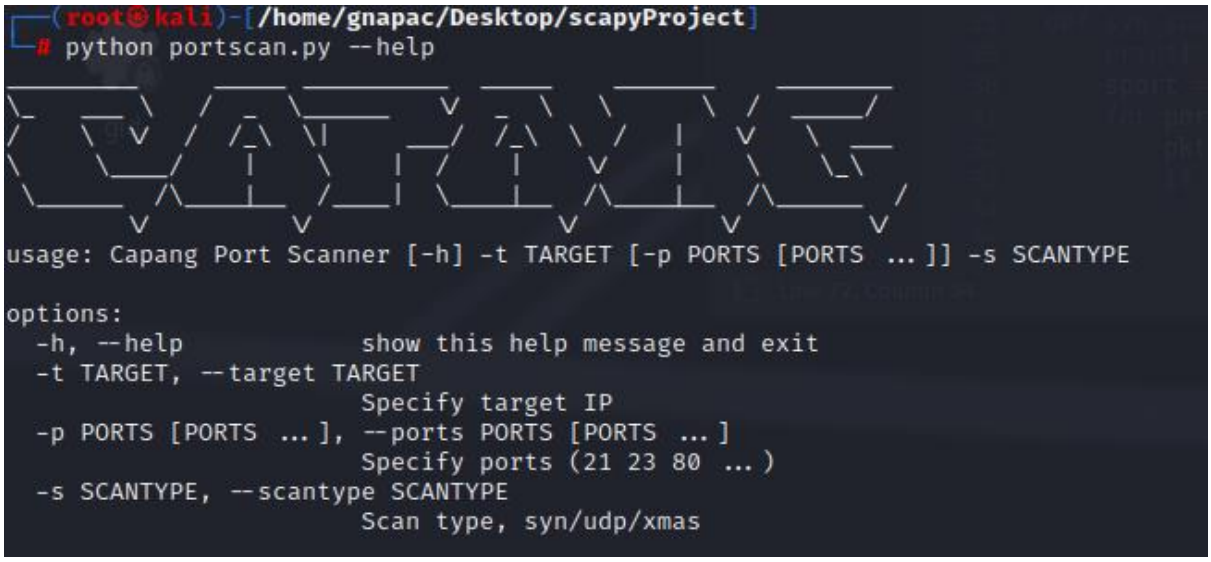
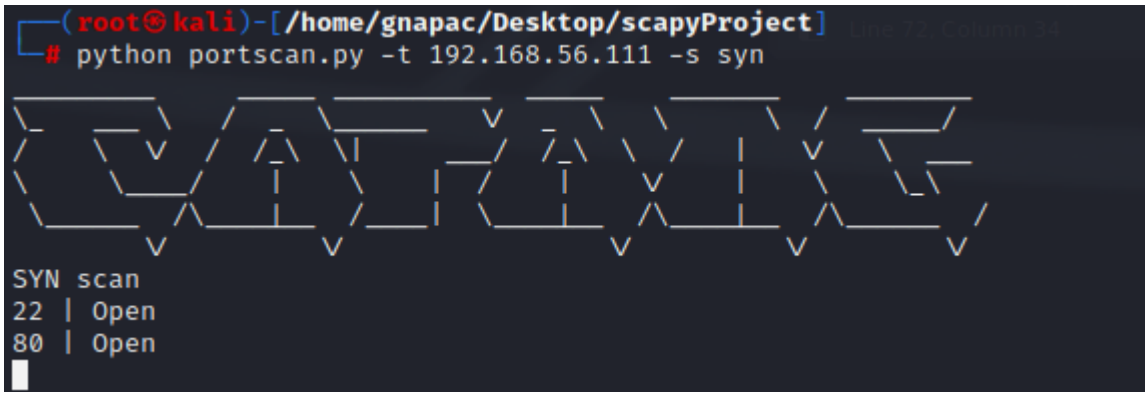
Port Scanner

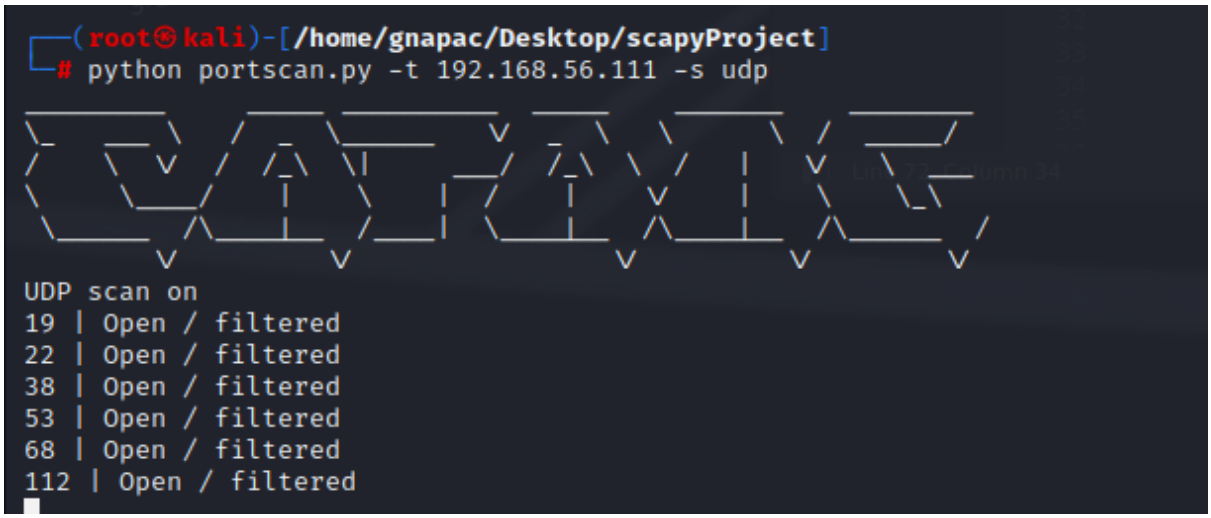


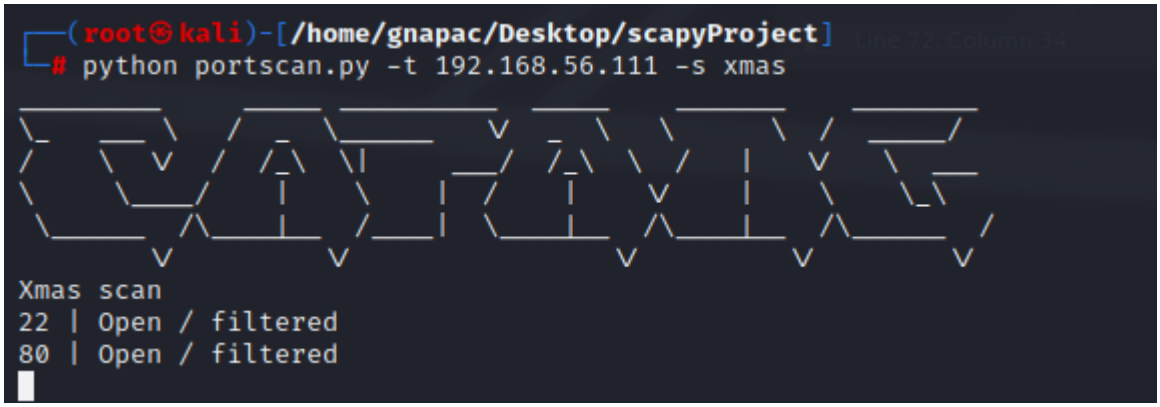
3 different type of scan

- SYN
- UDP
- XMAS

Can specify target ports if not specified, will scan top 1000 tcp/udp ports







Source Code

```
#!/usr/bin/env python

# port scanner
import argparse
from signal import signal, SIGINT
from scapy.all import *

def handler(signal_received, frame):
    # Handle any cleanup here
    print('\n.....Exiting gracefully')
    front()
    exit(0)

def front():
    # Molek sikit
    print('_____  _____ _____ _____  _____ ')
    print('\\_  ___ \\  / _ \\_____  \\/ _ \\  \\  \\ / _____/')
    print('/   \\ \\/ / /_\\ \\|    ___/ /_\\ \\ \\ /  |  \\/  \\ ___ ')
    print('\\    \\____/   |   \\   | /   |   \\/   |   \\   \\_\\ ')
    print(' \\_____ /\\____|__ /____| \\____|__ /\\____|__ /\\_____ /')
    print('       \\/       \\/           \\/       \\/      \\/ ')
```

```
# output format # TODO make prettier
def print_ports(port, state):
    print("%s | %s" % (port, state))


# syn scan
def syn_scan(target, ports):
    print("SYN scan")
    sport = RandShort()
    for port in ports:
        pkt = sr1(IP(dst=target)/TCP(sport=sport, dport=port, flags="S"), timeout=1, verbose=0)
        if pkt != None:
            if pkt.haslayer(TCP):
                if pkt[TCP].flags == 20:
                    pass
                elif pkt[TCP].flags == 18:
                    print_ports(port, "Open")
                else:
                    print_ports(port, "TCP packet resp / filtered")
            elif pkt.haslayer(ICMP):
                print_ports(port, "ICMP resp / filtered")
            else:
                print_ports(port, "Unknown resp")
                print(pkt.summary())
        else:
            print_ports(port, "Unanswered")


# udp scan
def udp_scan(target, ports):
    print("UDP scan on")
    for port in ports:
```

```
        pkt = sr1(IP(dst=target)/UDP(sport=port, dport=port), timeout=2, verbose=0)
        if pkt == None:
            print_ports(port, "Open / filtered")
        else:
            if pkt.haslayer(ICMP):
                pass
            elif pkt.haslayer(UDP):
                print_ports(port, "Open / filtered")
            else:
                print_ports(port, "Unknown")
                print(pkt.summary())

# xmas scan
def xmas_scan(target, ports):
    print("Xmas scan")
    sport = RandShort()
    for port in ports:
        pkt = sr1(IP(dst=target)/TCP(sport=sport, dport=port, flags="FPU"), timeout=1, verbose=0)
        if pkt != None:
            if pkt.haslayer(TCP):
                if pkt[TCP].flags == 20:
                    pass
                else:
                    print_ports(port, "TCP flag %s" % pkt[TCP].flag)
            elif pkt.haslayer(ICMP):
                print_ports(port, "ICMP resp / filtered")
            else:
                print_ports(port, "Unknown resp")
                print(pkt.summary())
        else:
            print_ports(port, "Open / filtered")
```

```
if __name__ == '__main__':
    front()
    signal(SIGINT, handler)
    # argument setup
    parser = argparse.ArgumentParser("Capang Port Scanner")
    parser.add_argument("-t", "--target", help="Specify target IP", required=True)
    parser.add_argument("-p", "--ports", type=int, nargs="+", help="Specify ports (21 23 80 ...)")
    parser.add_argument("-s", "--scantype", help="Scan type, syn/udp/xmas", required=True)
    args = parser.parse_args()

    # arg parsing
    target = args.target
    scantype = args.scantype.lower()
    # set ports if passed

    if args.ports:
        ports = args.ports
    else:
        # default port range
        if scantype == "syn" or scantype == "xmas" or scantype == "s" or scantype == "x":
            ports =
[1,3,4,6,7,9,13,17,19,20,21,22,23,24,25,26,30,32,33,37,42,43,49,53,70,79,80,81,82,83,84,85,88,89,9
0,99,100,106,109,110,111,113,119,125,135,139,143,144,146,161,163,179,199,211,212,222,254,255,
256,259,264,280,301,306,311,340,366,389,406,407,416,417,425,427,443,444,445,458,464,465,481,
497,500,512,513,514,515,524,541,543,544,545,548,554,555,563,587,593,616,617,625,631,636,646,
648,666,667,668,683,687,691,700,705,711,714,720,722,726,749,765,777,783,787,800,801,808,843,
873,880,888,898,900,901,902,903,911,912,981,987,990,992,993,995,999,1000,1001,1002,1007,100
9,1010,1011,1021,1022,1023,1024,1025,1026,1027,1028,1029,1030,1031,1032,1033,1034,1035,103
6,1037,1038,1039,1040,1041,1042,1043,1044,1045,1046,1047,1048,1049,1050,1051,1052,1053,105
4,1055,1056,1057,1058,1059,1060,1061,1062,1063,1064,1065,1066,1067,1068,1069,1070,1071,107
2,1073,1074,1075,1076,1077,1078,1079,1080,1081,1082,1083,1084,1085,1086,1087,1088,1089,109
0,1091,1092,1093,1094,1095,1096,1097,1098,1099,1100,1102,1104,1105,1106,1107,1108,1110,111
1,1112,1113,1114,1117,1119,1121,1122,1123,1124,1126,1130,1131,1132,1137,1138,1141,1145,114
7,1148,1149,1151,1152,1154,1163,1164,1165,1166,1169,1174,1175,1183,1185,1186,1187,1192,119
```

8,1199,1201,1213,1216,1217,1218,1233,1234,1236,1244,1247,1248,1259,1271,1272,1277,1287,129
6,1300,1301,1309,1310,1311,1322,1328,1334,1352,1417,1433,1434,1443,1455,1461,1494,1500,150
1,1503,1521,1524,1533,1556,1580,1583,1594,1600,1641,1658,1666,1687,1688,1700,1717,1718,171
9,1720,1721,1723,1755,1761,1782,1783,1801,1805,1812,1839,1840,1862,1863,1864,1875,1900,191
4,1935,1947,1971,1972,1974,1984,1998,1999,2000,2001,2002,2003,2004,2005,2006,2007,2008,200
9,2010,2013,2020,2021,2022,2030,2033,2034,2035,2038,2040,2041,2042,2043,2045,2046,2047,204
8,2049,2065,2068,2099,2100,2103,2105,2106,2107,2111,2119,2121,2126,2135,2144,2160,2161,217
0,2179,2190,2191,2196,2200,2222,2251,2260,2288,2301,2323,2366,2381,2382,2383,2393,2394,239
9,2401,2492,2500,2522,2525,2557,2601,2602,2604,2605,2607,2608,2638,2701,2702,2710,2717,271
8,2725,2800,2809,2811,2869,2875,2909,2910,2920,2967,2968,2998,3000,3001,3003,3005,3006,300
7,3011,3013,3017,3030,3031,3052,3071,3077,3128,3168,3211,3221,3260,3261,3268,3269,3283,330
0,3301,3306,3322,3323,3324,3325,3333,3351,3367,3369,3370,3371,3372,3389,3390,3404,3476,349
3,3517,3527,3546,3551,3580,3659,3689,3690,3703,3737,3766,3784,3800,3801,3809,3814,3826,382
7,3828,3851,3869,3871,3878,3880,3889,3905,3914,3918,3920,3945,3971,3986,3995,3998,4000,400
1,4002,4003,4004,4005,4006,4045,4111,4125,4126,4129,4224,4242,4279,4321,4343,4443,4444,444
5,4446,4449,4550,4567,4662,4848,4899,4900,4998,5000,5001,5002,5003,5004,5009,5030,5033,505
0,5051,5054,5060,5061,5080,5087,5100,5101,5102,5120,5190,5200,5214,5221,5222,5225,5226,526
9,5280,5298,5357,5405,5414,5431,5432,5440,5500,5510,5544,5550,5555,5560,5566,5631,5633,566
6,5678,5679,5718,5730,5800,5801,5802,5810,5811,5815,5822,5825,5850,5859,5862,5877,5900,590
1,5902,5903,5904,5906,5907,5910,5911,5915,5922,5925,5950,5952,5959,5960,5961,5962,5963,598
7,5988,5989,5998,5999,6000,6001,6002,6003,6004,6005,6006,6007,6009,6025,6059,6100,6101,610
6,6112,6123,6129,6156,6346,6389,6502,6510,6543,6547,6565,6566,6567,6580,6646,6666,6667,666
8,6669,6689,6692,6699,6779,6788,6789,6792,6839,6881,6901,6969,7000,7001,7002,7004,7007,701
9,7025,7070,7100,7103,7106,7200,7201,7402,7435,7443,7496,7512,7625,7627,7676,7741,7777,777
8,7800,7911,7920,7921,7937,7938,7999,8000,8001,8002,8007,8008,8009,8010,8011,8021,8022,803
1,8042,8045,8080,8081,8082,8083,8084,8085,8086,8087,8088,8089,8090,8093,8099,8100,8180,818
1,8192,8193,8194,8200,8222,8254,8290,8291,8292,8300,8333,8383,8400,8402,8443,8500,8600,864
9,8651,8652,8654,8701,8800,8873,8888,8899,8994,9000,9001,9002,9003,9009,9010,9011,9040,905
0,9071,9080,9081,9090,9091,9099,9100,9101,9102,9103,9110,9111,9200,9207,9220,9290,9415,941
8,9485,9500,9502,9503,9535,9575,9593,9594,9595,9618,9666,9876,9877,9878,9898,9900,9917,992
9,9943,9944,9968,9998,9999,10000,10001,10002,10003,10004,10009,10010,10012,10024,10025,10
082,10180,10215,10243,10566,10616,10617,10621,10626,10628,10629,10778,11110,11111,11967,
12000,12174,12265,12345,13456,13722,13782,13783,14000,14238,14441,14442,15000,15002,1500
3,15004,15660,15742,16000,16001,16012,16016,16018,16080,16113,16992,16993,17877,17988,18
040,18101,18988,19101,19283,19315,19350,19780,19801,19842,20000,20005,20031,20221,20222,
20828,21571,22939,23502,24444,24800,25734,25735,26214,27000,27352,27353,27355,27356,2771
5,28201,30000,30718,30951,31038,31337,32768,32769,32770,32771,32772,32773,32774,32775,32
776,32777,32778,32779,32780,32781,32782,32783,32784,32785,33354,33899,34571,34572,34573,
35500,38292,40193,40911,41511,42510,44176,44442,44443,44501,45100,48080,49152,49153,4915
4,49155,49156,49157,49158,49159,49160,49161,49163,49165,49167,49175,49176,49400,49999,50
000,50001,50002,50003,50006,50300,50389,50500,50636,50800,51103,51493,52673,52822,52848,
52869,54045,54328,55055,55056,55555,55600,56737,56738,57294,57797,58080,60020,60443,6153
2,61900,62078,63331,64623,64680,65000,65129,65389]

elif scantype == "udp" or scantype == "u":

ports =
[2,3,7,9,13,17,19,20,21,22,23,37,38,42,49,53,67,68,69,80,88,111,112,113,120,123,135,136,137,138,
139,158,161,162,177,192,199,207,217,363,389,402,407,427,434,443,445,464,497,500,502,512,513,

514,515,517,518,520,539,559,593,623,626,631,639,643,657,664,682,683,684,685,686,687,688,689,
764,767,772,773,774,775,776,780,781,782,786,789,800,814,826,829,838,902,903,944,959,965,983,
989,990,996,997,998,999,1000,1001,1007,1008,1012,1013,1014,1019,1020,1021,1022,1023,1024,1
025,1026,1027,1028,1029,1030,1031,1032,1033,1034,1035,1036,1037,1038,1039,1040,1041,1042,1
043,1044,1045,1046,1047,1048,1049,1050,1051,1053,1054,1055,1056,1057,1058,1059,1060,1064,1
065,1066,1067,1068,1069,1070,1072,1080,1081,1087,1088,1090,1100,1101,1105,1124,1200,1214,1
234,1346,1419,1433,1434,1455,1457,1484,1485,1524,1645,1646,1701,1718,1719,1761,1782,1804,1
812,1813,1885,1886,1900,1901,1993,2000,2002,2048,2049,2051,2148,2160,2161,2222,2223,2343,2
345,2362,2967,3052,3130,3283,3296,3343,3389,3401,3456,3457,3659,3664,3702,3703,4000,4008,4
045,4444,4500,4666,4672,5000,5001,5002,5003,5010,5050,5060,5093,5351,5353,5355,5500,5555,5
632,6000,6001,6002,6004,6050,6346,6347,6970,6971,7000,7938,8000,8001,8010,8181,8193,8900,9
000,9001,9020,9103,9199,9200,9370,9876,9877,9950,10000,10080,11487,16086,16402,16420,1643
0,16433,16449,16498,16503,16545,16548,16573,16674,16680,16697,16700,16708,16711,16739,16
766,16779,16786,16816,16829,16832,16838,16839,16862,16896,16912,16918,16919,16938,16939,
16947,16948,16970,16972,16974,17006,17018,17077,17091,17101,17146,17184,17185,17205,1720
7,17219,17236,17237,17282,17302,17321,17331,17332,17338,17359,17417,17423,17424,17455,17
459,17468,17487,17490,17494,17505,17533,17549,17573,17580,17585,17592,17605,17615,17616,
17629,17638,17663,17673,17674,17683,17726,17754,17762,17787,17814,17823,17824,17836,1784
5,17888,17939,17946,17989,18004,18081,18113,18134,18156,18228,18234,18250,18255,18258,18
319,18331,18360,18373,18449,18485,18543,18582,18605,18617,18666,18669,18676,18683,18807,
18818,18821,18830,18832,18835,18869,18883,18888,18958,18980,18985,18987,18991,18994,1899
6,19017,19022,19039,19047,19075,19096,19120,19130,19140,19141,19154,19161,19165,19181,19
193,19197,19222,19227,19273,19283,19294,19315,19322,19332,19374,19415,19482,19489,19500,
19503,19504,19541,19600,19605,19616,19624,19625,19632,19639,19647,19650,19660,19662,1966
3,19682,19683,19687,19695,19707,19717,19718,19719,19722,19728,19789,19792,19933,19935,19
936,19956,19995,19998,20003,20004,20019,20031,20082,20117,20120,20126,20129,20146,20154,
20164,20206,20217,20249,20262,20279,20288,20309,20313,20326,20359,20360,20366,20380,2038
9,20409,20411,20423,20424,20425,20445,20449,20464,20465,20518,20522,20525,20540,20560,20
665,20678,20679,20710,20717,20742,20752,20762,20791,20817,20842,20848,20851,20865,20872,
20876,20884,20919,21000,21016,21060,21083,21104,21111,21131,21167,21186,21206,21207,2121
2,21247,21261,21282,21298,21303,21318,21320,21333,21344,21354,21358,21360,21364,21366,21
383,21405,21454,21468,21476,21514,21524,21525,21556,21566,21568,21576,21609,21621,21625,
21644,21649,21655,21663,21674,21698,21702,21710,21742,21780,21784,21800,21803,21834,2184
2,21847,21868,21898,21902,21923,21948,21967,22029,22043,22045,22053,22055,22105,22109,22
123,22124,22341,22692,22695,22739,22799,22846,22914,22986,22996,23040,23176,23354,23531,
23557,23608,23679,23781,23965,23980,24007,24279,24511,24594,24606,24644,24854,24910,2500
3,25157,25240,25280,25337,25375,25462,25541,25546,25709,25931,26407,26415,26720,26872,26
966,27015,27195,27444,27473,27482,27707,27892,27899,28122,28369,28465,28493,28543,28547,
28641,28840,28973,29078,29243,29256,29810,29823,29977,30263,30303,30365,30544,30656,3069
7,30704,30718,30975,31059,31073,31109,31189,31195,31335,31337,31365,31625,31681,31731,31
891,32345,32385,32528,32768,32769,32770,32771,32772,32773,32774,32775,32776,32777,32778,
32779,32780,32798,32815,32818,32931,33030,33249,33281,33354,33355,33459,33717,33744,3386
6,33872,34038,34079,34125,34358,34422,34433,34555,34570,34577,34578,34579,34580,34758,34
796,34855,34861,34862,34892,35438,35702,35777,35794,36108,36206,36384,36458,36489,36669,
36778,36893,36945,37144,37212,37393,37444,37602,37761,37783,37813,37843,38037,38063,3829
3,38412,38498,38615,39213,39217,39632,39683,39714,39723,39888,40019,40116,40441,40539,40
622,40708,40711,40724,40732,40805,40847,40866,40915,41058,41081,41308,41370,41446,41524,
41638,41702,41774,41896,41967,41971,42056,42172,42313,42431,42434,42508,42557,42577,4262

7,42639,43094,43195,43370,43514,43686,43824,43967,44101,44160,44179,44185,44190,44253,44334,44508,44923,44946,44968,45247,45380,45441,45685,45722,45818,45928,46093,46532,46836,47624,47765,47772,47808,47915,47981,48078,48189,48255,48455,48489,48761,49152,49153,49154,49155,49156,49157,49158,49159,49160,49161,49162,49163,49165,49166,49167,49168,49169,49170,49171,49172,49173,49174,49175,49176,49177,49178,49179,49180,49181,49182,49184,49185,49186,49187,49188,49189,49190,49191,49192,49193,49194,49195,49196,49197,49198,49199,49200,49201,49202,49204,49205,49207,49208,49209,49210,49211,49212,49213,49214,49215,49216,49220,49222,49226,49259,49262,49306,49350,49360,49393,49396,49503,49640,49968,50099,50164,50497,50612,50708,50919,51255,51456,51554,51586,51690,51717,51905,51972,52144,52225,52503,53006,53037,53571,53589,53838,54094,54114,54281,54321,54711,54807,54925,55043,55544,55587,56141,57172,57409,57410,57813,57843,57958,57977,58002,58075,58178,58419,58631,58640,58797,59193,59207,59765,59846,60172,60381,60423,61024,61142,61319,61322,61370,61412,61481,61550,61685,61961,62154,62287,62575,62677,62699,62958,63420,63555,64080,64481,64513,64590,64727,65024]

```
    # scan types
    if scantype == "syn" or scantype == "s":
        syn_scan(target, ports)
    elif scantype == "udp" or scantype == "u":
        udp_scan(target, ports)
    elif scantype == "xmas" or scantype == "x":
        xmas_scan(target, ports)
    else:
        print("Scan type not supported")
```