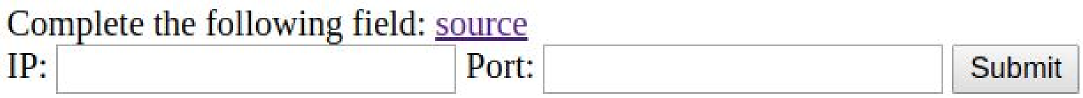
Web: Python

We are given the following description:

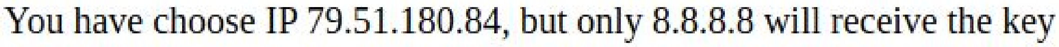
“This is my raspberry pi at home. I have some secrets that only me can received.
Do you want to try?”



Complete the following field: source
IP: Port:

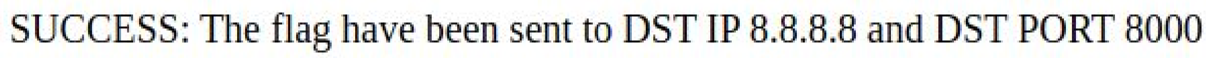
We need to insert an IP and a Port in order to do something, i.e., reach the flag. In addition, the app contains a source (see the link): if we open it, we can see a Python code. Let’s copy this code in local and try to figure it out what it’s doing.

But first, let’s try to see the output of the program. If we insert random values, and we are going to receive the following message:



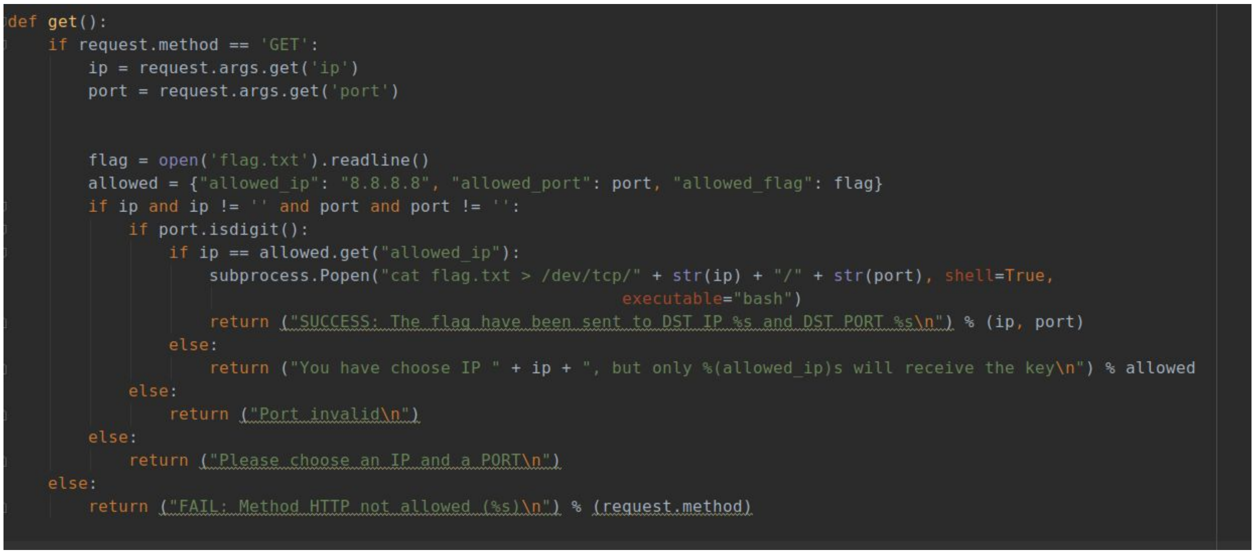
You have choose IP 79.51.180.84, but only 8.8.8.8 will receive the key

So, let’s try to use as an IP 8.8.8.8 and as port 8000.



SUCCESS: The flag have been sent to DST IP 8.8.8.8 and DST PORT 8000

Let’s go and analyze the source. First of all, we need, with patience, to clean the code. After that we can understand the various *if-else* statements.



```
def get():
    if request.method == 'GET':
        ip = request.args.get('ip')
        port = request.args.get('port')


        flag = open('flag.txt').readline()
        allowed = {"allowed_ip": "8.8.8.8", "allowed_port": port, "allowed_flag": flag}
        if ip and ip != '' and port and port != '':
            if port.isdigit():
                if ip == allowed.get("allowed_ip"):
                    subprocess.Popen("cat flag.txt > /dev/tcp/" + str(ip) + "/" + str(port), shell=True,
                                                                executable="bash")
                    return ("SUCCESS: The flag have been sent to DST IP %s and DST PORT %s\n") % (ip, port)
                else:
                    return ("You have choose IP " + ip + ", but only %(allowed_ip)s will receive the key\n") % allowed
            else:
                return ("Port invalid\n")
        else:
            return ("Please choose an IP and a PORT\n")
    else:
        return ("FAIL: Method HTTP not allowed (%s)\n") % (request.method)
```

Let’s start with the GET function, where three variables are involved:

- IP : the IP that we provide;
- Port: the port that we provide;
- flag : a string containing (?) the flag.

A dictionary called allowed contains a restriction on the IP number, which must be 8.8.8.8

The first IF checks that IP and Port are not empty. If not, we check if the port is a number: if not, we receive a message error.

Based on this, we get that:

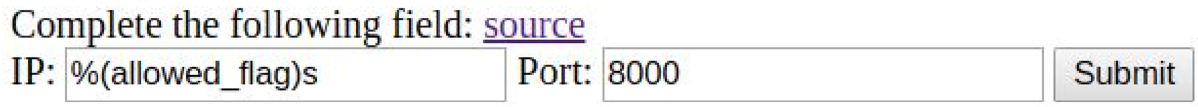
- IP = 8.8.8.8;
- Port = any digit number.

When we insert this right combination, a subprocess containing the flag is opened and sent to the given IP address and port.

However, we should focus on the *else* statement generated by a wrong IP. As you cans see the *print* returns two variables:

- Our inserted IP value;
- The correct IP value (the mandatory one).

Can you see it? The print uses the format “%(allowed_ip)s” for printing the value contained in the dictionary. Well, can we use this info for printing the flag?



Complete the following field: source

IP: %(allowed_flag)s Port: 8000

This returns a message (as before), but this time the inserted IP shows us the flag:

INSA{Y0u_C@n_H@v3_fUN_W1Th_pYth0n}