# CE 340 Cryptography & Network Security Assignment 2

**Title:** Building a simple pentest tool
**Defined by:** Süleyman KONDAKCI
**Date to deliver:** 19.05.2022, 17:15
**Project members:** Max. 2 students

You will write a script (pentest.py) containing a set of **Python** functions (tasks), which will be invoked via a main menu of pentest.py. This script is your source file, which should be executed in a shell window (not using a Python IDE. For example, one types the following Shell commands to execute the script (here **192.168.1.30** is the host IP under test).

```
$ chmod +x pentest.py
$ sudo ./pentest.py 192.168.1.30
```

Your script will contain many (10) tasks that are explained below. The script will also contain a menu showing the tasks. When you execute the script, a menu will be displayed from which you will choose a function to invoke. The List of the functions (task) you will implement is given in the following table.

| | |
|---|---|
| **ICMP ping** | Ping an IP range and collect IP addresses of the hosts that are alive and save the result in a text file, call this **icmp.dat**. |
| **Port identification** | 1) Get the IP addresses from the **imcp.dat** file, scan, and validate these IP addresses. If an IP address is a valid live host address, append it to a string (live hosts) that contains the network of the live hosts. A live host is an active host that can be monitored by a Scapy ICMP request.<br>2) Now make port scan on the live hosts. The scanning must find and identify ports on each host and save the results into a text file, call this **ports.dat**. The text file will contain Host IPs, ports numbers, and service names (if any) of each port. |
| **Open port identification** | 1) Get the IP addresses from the Port Identification (**ports.dat** ) file and scan and check and validate these IP addresses. If the address is a valid live host address, append it to a string (live hosts) that contains the network of the live hosts.<br>2) Now scan the live hosts. The scanning must find open ports from each host and save the results into a text file, call this **open_ports.dat**. |
| **OS Fingerprint identification** | This function will get the host IPs from the text file (open_ports.dat) and identify operating systems (OS) and OS versions of the hosts with open ports. |
| **Router & Firewall detection** | Scan and find neighbor router and firewall addresses, protocols, and ports of each router. Save the result into text file, call this **wall.dat**. |
| **Web server detection** | Scan and find 10 web-server addresses, protocols, and ports of each web server. Save the result into text file, call this **web.dat**. |
| **SNMP detection** | Scan and find neighbor hosts addresses having the SNMP-protocols, and ports of each host. Save the result into text file, call this **snmp.dat**. |
| **SYN_flood** | This function will launch SYN-flood attack to a given destination (IP) and |

| | |
|---|---|
| | port(s). This tool must also enable you to choose the number of flooding, e.g., 10.000 SYN attacks. While performing the attack start Wireshark or tcpdump to monitor the attacks.<br>Example: **`SYN_flood –pT 1-80 193.60.70.5`**<br>This will attack all TCP (T) ports (p) between 1 and 80 on the machine with the IP = **`193.60.70.5.`** |
| **Show** | This function will ask and display the contents of the files that your tools have created so far. |
| **Sniff** | This function will sniff a network and display the network traffic on the screen. It should allow you to select one or multiple input arguments. For example, source and destination ports, source and destination hosts, source and destination protocols, and source and destination networks.<br>Example: **`sniff –pT 23,80 193.60.70.*`**<br>That will sniff TCP (T) protocol ports (p) on all machines found in network **193.60.70.** |

## What to deliver?

1) Execution trace (e.g., screenshots) of each operation
2) Source files and User guide zipped and uploaded to Blackboard
3) Make sure that you can successfully present the project in the classroom (25% of total score)

Good Luck!

*S. Kondakci*