Nmap:

Nmap is by far the most popular tool in the arsenal of any network administrator or security enthusiast. It comes pre-installed with Kali and other security-based Linux distributions. It comes with a GUI called Zenmap however, in this lab we will be working with the command line version of the tool. Let's have a look at some of the commands.

To scan a single IP:

```
nmap 10.10.10.10
```

To scan a webhost:

```
nmap www.upm.edu.sa
```

To scan a range of IP addresses:

```
nmap 10.10.10.10-20
```

To scan a subnet:

```
nmap 10.10.10.10/24
```

To scan a single port:

```
nmap -p 80 10.10.10.10
```

To scan a range of ports:

```
nmap -p 80-800 10.10.10.10
```

To use TCP SYN scan:

```
nmap -sS 10.10.10.10
```

To use TCP Connect scan:

```
nmap -sT 10.10.10.10
```

To use UDP scan:

```
nmap -sU 10.10.10.10
```

To detect the OS:

```
nmap -O 10.10.10.10
```

To detect the services running:

```
nmap -sV 10.10.10.10
```

You can set how aggressively Nmap should search for services by setting a parameter called the version intensity (value between 0 to 5). However, the more aggressive the setting the more noisy (detectable) Nmap becomes. For instance:

```
nmap -sV -version-intensity 5 10.10.10.10
```

To detect the OS and the services together:

```
nmap -A 10.10.10.10
```

Try some of the above commands on UPM's domain to see what the result are.

Nikto2:

Nikto2 comes pre-installed within Kali Linux and other Linux Operating Systems designed for pen-testers. If you are using another version of Linux, you can download Nikto by following the link below:
https://cirt.net/nikto2

Alternatively, install Nikto from a command terminal using the following command:

```
$ sudo apt-get install nikto
```

Once installed, you can find the documentation here: https://cirt.net/nikto2-docs/

First, open up a new command terminal and use command `nikto -h` to load help options or `nikto -H` for full help options. Read the various types of options available to get an understanding of what this tool can do for you. Once you understand the options a little and you are ready to proceed, choose what options you would want Nikto to use while scanning for vulnerabilities on a target webserver.

The most basic Nikto scan requires simply a host to target, port 80 is assumed if none is specified. The host can either be an IP or a hostname of a machine, and is specified using the -h (-host) option. The following command will scan the IP 192.168.0.1 on TCP port 80:

```
$ nikto -h 192.168.0.1
```

If the above command does not work, try the following:

```
$ perl nikto.pl -h 192.168.0.1
```

To check on a different port, specify the port number with the -p (-port) option. This command will scan the IP 192.168.0.1 on TCP port 443:

```
$ nikto -h 192.168.0.1 -p 443
```

Again, if the above command does not work, try the following:

```
$ perl nikto.pl -h 192.168.0.1 -p 443
```

Hosts, ports and protocols may also be specified by using a full URL syntax, and it will be scanned:

```
$ nikto -h https://192.168.0.1:443/
```

There is no need to specify that port 443 may be SSL, as Nikto will first test regular HTTP and if that fails, HTTPS. If you are sure it is an SSL server, specifying -s (-ssl) will speed up the test.

```
$ nikto -h 192.168.0.1 -p 443 -ssl
```

Nikto can scan multiple ports in the same scanning session. To test more than one port on the same host, specify the list of ports in the -p (-port) option. Ports can be specified as a range (i.e., 80-90), or as a comma-delimited list, (i.e., 80,88,90). The following command will scan the host on ports 80, 88 and 443.

```
$ nikto -h 192.168.0.1 -p 80,88,443
```

Nikto also supports scanning multiple hosts in the same session via a text file of host names or IP addresses. Instead of giving a single host name or IP for the -h (-host) option, a file name can be given. A file of hosts must be formatted as one host per line, with the port numbers at the end of each line. Ports can

be separated from the host and other ports via a colon or a comma. If no port is specified, port 80 is assumed. The following is an example of a valid hosts file:

```
192.168.0.1:80 http://192.168.0.1:8080/ 192.168.0.3
```

Put the text above in a file called target.txt and run the following command to initiate a scan on multiple hosts simultaneously:

```
$ nikto -host target.txt
```

Once a scan is initiated on a server, Nikto finds all relevant information about the target that could potentially be used to exploit the target server. Each vulnerability found will be labeled with an exploit code. We can then look up this code in various exploit databases such as Exploit-DB, Security Focus or the CVE (Common Vulnerabilities and Exposure) database. Sometimes a Nikto scan can reveal some false positives as well. This is because Nikto does not actually exploit each of the possible vulnerabilities. Rather it just scans to see if the server will respond, without any errors, to any known exploitable URLs.
Nikto is a very powerful tool. It contains a large collection of plugins that are designed to enhance the functionality and accuracy of the scans that Nikto can run. To list all plugins available to Nikto use command.

```
$ nikto --list-plugins
```

This command will show you a list of all the plugins supported. Plugins are very useful and different plugins can result in different outputs. They can also provide you more guarantees about whether or not a reported vulnerability is actually a vulnerability or just a false positive. For example, if a normal scan concluded that the server was vulnerable to the Apache Expect Header XSS attack (a popular cross site scripting attack) and we want to run a test to make sure that it is actually vulnerable by adding debugging, we can run the following command:

```
$ nikto -h target.com -Plugins "apache_expect_xss(verbose,debug)"
```

And then manually check the output to see whether it was truly vulnerable or not.
Nikto can also be used to fool the target. The victim can be misled into believing that the scan is actually not a scan and just a bunch of legitimate requests. This is very useful feature of Nikto. For instance, Google's crawlers will often visit websites, and are one of the least suspicious entities in a webserver's visit or error logs. We can use this to our advantage by using the following command to mimic a Google bot:

```
$ nikto -h target.com -useragent "Googlebot (compatible; Googlebot/2.1;
+http://www.google.com/bot.html)"
```

This command will set Google crawler as the user agent issuing the requests. You can use any custom HTTP header and user agents that you want however, for this lab we have kept it simple and used Google crawler as an example.