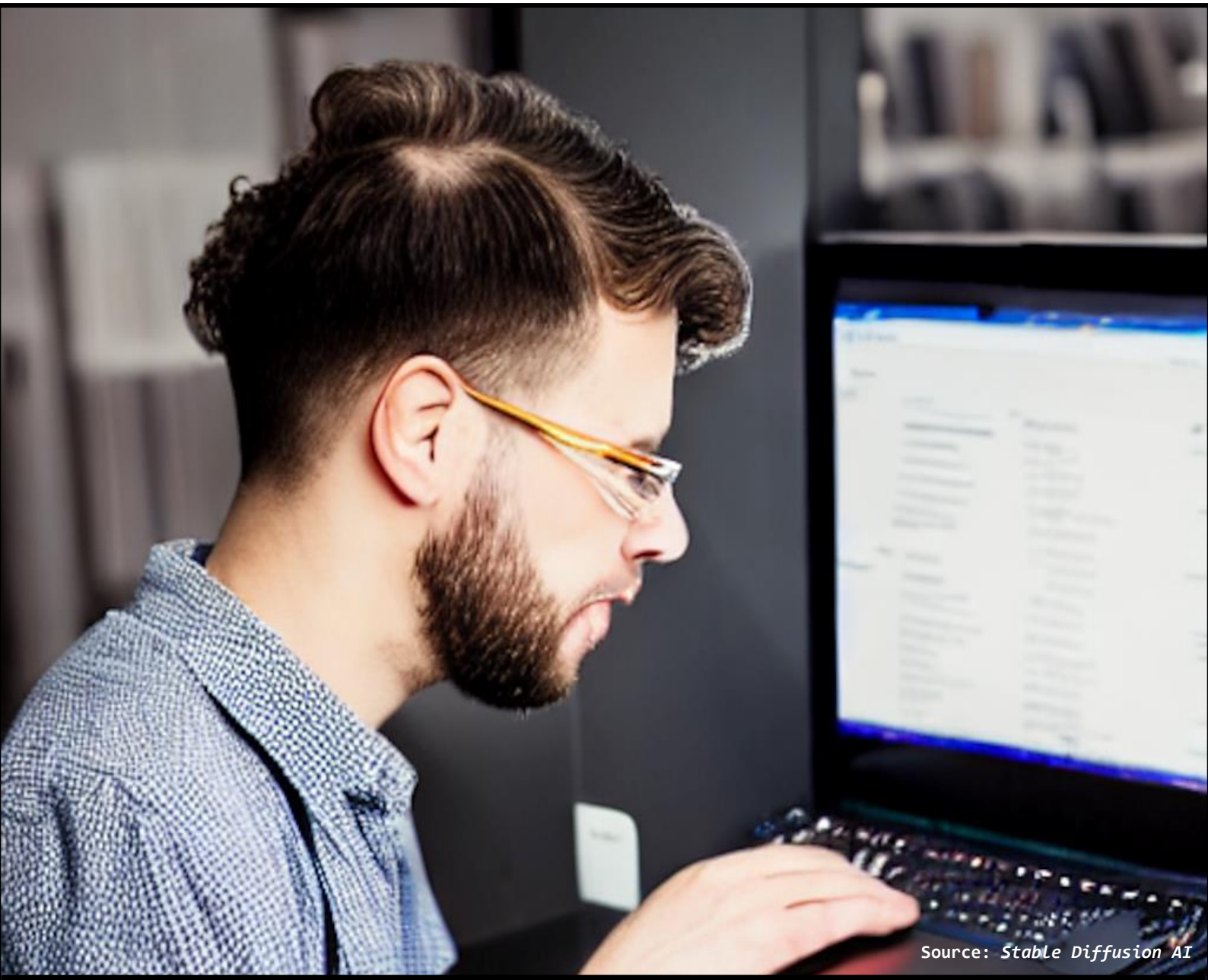


LABORATORY 1. SYSTEM SETUP

DEPARTMENT OF COMPUTER SCIENCE. UNIVERSITY OF OVIEDO

Computer Security | 2022 – 2023 (V2.7 *"S-81 Isaac Peral"*)

CONTENT

BLOCK 1: VIRTUAL MACHINE SETUP

Prerequisites

Make sure you have a working virtual machine before starting this laboratory. Please refer to the file "*Lab 0A. Creating your VM for the Course*" for instructions about what options you have for deploying a suitable *Ubuntu* virtual machine for the course (Automated via *Vagrant*, preconstructed .ova image, or manual installation) and ensure it is up and running previously to start. **This is a prerequisite to start the labs of the course, so be sure to fulfill it.**

NOTE: Maybe you will experience a *VirtualBox* error regarding the Host-only adapter when you first boot the VM. If this is your case, remove this network interface from the machine properties in *VirtualBox* and run it again. This usually happens in the school labs due to permission problems, not at home. Remember that you can still access the VM through SSH doing `ssh localhost:2222` (Option 1 or Option 2 only).

Login and update the virtual machine

Practical application: *You need to update your OS software to prevent vulnerabilities*

Once deployed, you must update your virtual machine. You can do this as follows.

```
sudo apt update
sudo apt full-upgrade
```

To potentially save disk space, you can also (optionally) run

```
sudo apt autoremove
sudo apt autoclean
```

Ubuntu systems also use the `snap` package manager to install programs, and you should also check the software installed with this package manager for updates with

```
sudo snap refresh
```

Expected Results: This activity will be complete when your OS is updated. If you choose the Vagrant VM building method, it is possible that this command does not update anything if you just built the VM. Why do you think this is happening?

Create a proper user

NOTE: If you choose the Vagrant automated VM option and updated your username in the file as indicated, this exercise will be already done, and this just describes the procedure that was followed. Our future work will be ALWAYS performed using a user account whose name is your UO. The goal is that no matter what you do, the VM should have two users (your UO being one of them)

Practical application: *You need create a user identity for a new employee*

A new user `uoxxxxxx` must be created and introduced in the `sudo` group (to promote to `root` if needed for privileged operations such as installing software). You can follow this: https://www.digitalocean.com/community/tutorials/how-to-create-a-sudo-user-on-ubuntu-quickstart

- Create the user: `sudo adduser <uoxxxxxx>`
- Set and confirm the new user's password at the prompt
- Set the new user's information. It is fine to accept the defaults to leave all this information blank.
- Use the `usermod` command to add the user to the `sudo` group: `sudo usermod -aG sudo <uoxxxxxx>`. By default, on *Ubuntu*, members of the `sudo` group have `sudo` privileges.

Expected Results: This activity will be complete when the new user is operational and can use `sudo`.

Properly change between users

Practical application: *You need to assume multiple user identities in the same work session for certain operations*

Now that you have two users you must practice how to change between them. This is done through the `su` command, but please specify the `-` parameter (e. g.: `su - ssiuser`) to start the shell as a login shell with an environment like a real login. **Otherwise, some programs may not work properly in the future! NOTE:** Changing between users in the command line (even with `su -`) may prevent the GUI to start with the new user. You may have to do a proper log out. **NOTE**: VMs deployed with *Vagrant* already have two sudoer users, `vagrant` and the one you specified in the `username.txt` file.

Expected Results: This activity will be complete when user change works without failure.

BLOCK 2: BASIC SECURITY OPERATIONS

Install an OpenSSH server

Practical application: *You need secure remote access to your machine*

Install this secure remote connection server and check that it works properly from your host machine. You can follow this tutorial: https://linuxize.com/post/how-to-enable-ssh-on-ubuntu-18-04/ or read the course documentation about *VirtualBox* we provide. **Please make sure that your host can connect to your VM before you continue to the other steps. NOTE: *Vagrant* and the preconstructed VM already have *OpenSSH* installed, so you only must ensure you can connect through SSH to the VM.** You can use any SSH client you prefer.

Expected Results: This activity will be complete when you are able to connect to your machine using ssh from your host.

Enable the firewall

Practical application: *You need to block any incoming connections to any service you do not authorize explicitly (safest policy)*

`ufw` (*Uncomplicated Firewall*) is a convenient front-end for the traditional Linux firewall `iptables` that facilitates its management. The goal of this part of the lab is twofold:

1. Enable the firewall
2. Allow only those ports that will have working services. It is important to remember a couple of things
   - All `ufw` commands you will need are in official manual: https://help.ubuntu.com/community/UFW
   - If you are using SSH to access your machine **you MUST FIRST open the SSH port BEFORE enabling the firewall**, or you will lock yourself out of the system via SSH. **NOTE**: Virtual machine consoles do not use SSH, so you can use them to recover remote access. If you only use the virtual machine console to access the machine, the `ssh` port/service may be disabled/unused for security reasons.
   - Do not forget to open the corresponding port every time you install a service. For example, when you install a web server you must allow the `http` and `https` (if used) services (equivalent to port 80 and 443)

Therefore, the following procedure must be followed:

- Type first: `sudo ufw allow ssh`
- And then: `sudo ufw enable`
- Check that the open ports are correct with `sudo ufw status`

Expected Results: This activity will be complete when the firewall reports that it is active and the SSH port is allowed.

Evaluate base VM security configuration: *Lynis*

Practical application: *You need to quickly evaluate the security level of your OS with a score from 0 to 100, and obtain hardening advice*

NOTE: *Vagrant* and the preconstructed VM already have the latest version of *Lynis* installed, so the installation process is not necessary, and you only need to execute it. If you manually created the VM, you should know that the *Lynis* package that come with *Ubuntu* is not the last stable version available, so we must follow a couple of steps to use this latest version properly:

- Download the software vendor (*CISofy*) key to use the contents of their repository

```
sudo wget -O - https://packages.cisofy.com/keys/cisofy-software-public.key | sudo apt-key add -
```

- Install support for secure (HTTPS) downloads from `apt` repositories (necessary to load *CISofy* contents).

```
sudo apt install apt-transport-https
```

- Add the *Lynis* repository to the OS

```
echo "deb https://packages.cisofy.com/community/lynis/deb/ stable main" | sudo tee
/etc/apt/sources.list.d/cisofy-lynis.list
```

- Update the Ubuntu package database to pick the newly added packages: `sudo apt update`
- Install the software (`apt install lynis`) and check the version (`lynis show version`) (should be 3.0.6 or higher)
- Evaluate the base system using *Lynis* using the appropriate option. It is recommended to redirect the output to a file to see the results. Results are best viewed (colors) using `more` over the file.

Expected Results: This activity will be complete when you can obtain a Lynis score of your system.

Basic enumeration with Nmap

Practical application: *You need to locate remote "alive" machines in any of your networks*

Perform a basic ping scan over a host you choose to check if it is "alive" with `nmap` (your own host machine, `scanme.nmap.org`...). You can use this cheatsheet to look for the appropriate option. **NOTE**: If for some reason it is not already installed, you can install it with `sudo apt install nmap`

Nmap 7.80 Cheatsheet series (ingenieriainformatica.uniovi.es)

Part 1: Reconnaissance (Basic)

https://nmap.org/

GENERAL USAGE

```
nmap [Scan Type(s)] [Options] {target specification}
```

NOTES

Target specifications can be host names, IP addresses, ranges, networks, etc. (scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.0-255.1-254)

Use these options to Locate "alive machines" (sometimes only returns that, sometimes they also return some port / service information)

TARGET SPECIFICATION

-iL <inputfilename>: Input from from file a list of hosts/networks

-iR <num hosts>: Choose random targets

--exclude <host1[,host2][,host3],...>: Exclude hosts/networks

--excludefile <exclude_file>: Exclude list from file

RECONOISSANCE EXAMPLES

nmap -sn scanme.nmap.org

sudo nmap -PS22,80 scanme.nmap.org

sudo nmap --traceroute scanme.nmap.org

```
operario@kali:~$ nmap -sn 192.168.20.10
Starting Nmap 7.80 ( https://nmap.org ) at 2020-09-21 18:38 CEST
Nmap scan report for 192.168.20.10
Host is up (0.00085s latency).
Nmap done: 1 IP address (1 host up) scanned in 13.01 seconds
```

```
operario@kali:~$ sudo nmap -PS22,80 192.168.20.10
Starting Nmap 7.80 ( https://nmap.org ) at 2020-09-21 18:42 CEST
Nmap scan report for 192.168.20.10
Host is up (0.00012s latency).
Not shown: 999 closed ports
PORT   STATE SERVICE
80/tcp open  http
MAC Address: 08:00:27:67:7A:EF (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 13.30 seconds
```

HOST DISCOVERY OPTIONS (WAYS TO CHECK "ALIVE" MACHINES)

--dns-servers <serv1[,serv2],...>: Specify custom DNS servers

-n/-R: Never do DNS resolution/Always resolve [default: sometimes]

-PE/PP/PM: ICMP echo, timestamp, and netmask request discovery probes

-Pn: Treat all provided hosts as online -- skip host discovery

-PO[protocol list]: IP Protocol Ping

-PS/PA/PU/PY[portlist]: TCP SYN/ACK, UDP or SCTP discovery to given ports

-sL: List Scan - simply list targets to scan

-sn: Ping Scan - disable port scan

--system-dns: Use OS's DNS resolver

--traceroute: Trace hop path to each host

Now perform a `ping` and the same `nmap` scan to www.asturias.es and see the differences. *Why do you think this is happening?*

Expected Results: This activity will be complete when you can tell if a remote machine is "alive" or not using `nmap`.

BLOCK 3: ADDITIONAL BASIC SECURITY OPERATIONS

Put a proper password

Practical application: *You need to change the password of any user with a proper and secure one*

Ensure that your user password is secure. You must change your UO VM user password for something that passes a minimum level of security strength test to understand how strong passwords are created. You can use tools like these to evaluate candidate passwords:

- http://www.passwordmeter.com
- https://password.kaspersky.com

Change the password using `passwd`: https://www.linuxtechi.com/10-passwd-command-examples-in-linux/

Expected Results: This activity will be complete when your chosen password is reported as very secure.

Enable malware / *rootkit* scanning software

Practical application: *You need the ability of locating malware or rootkits on-demand in any of your files*

It is important for a secure system to be able to detect and stop potential malware / rootkits that may be present in the system. To do that, you should install the following tools

Rootkits. Chkrootkit

- Install: `sudo apt install chkrootkit`
- Run the program (requires `root` privileges) and redirect the output to a file. Check the file to see if something has been detected

Rookits. Rkhunter

- Install the application with `sudo apt install rkhunter`. The application is prepared to send alerts via email, but we need an email server to do that. As we will not install one in our test lab, we configure it to send only local emails, leaving the rest of the options as default.
- Run and send the output to a file, checking that everything is correct: `sudo rkhunter -c` (**NOTE**: It may take a while to complete)

Malware. ClamAV

- Install the software: `sudo apt install clamav`
- Perform a signature update of the antimalware software: `sudo freshclam`
- **NOTE**: If signature update complains about the log file locked for writing, do: `sudo service clamav-freshclam stop` and then do the signature update again. **If the remote server appears to be down, skip the update**.
- Scan a directory: `sudo clamscan -r -i <DIRECTORY>`

Expected Results: This activity will be complete when you can successfully run the mentioned tools.

Lynis reports

Practical application: *You need to obtain a security report of the current hardening level or your machine*

Lynis outputs its results to the standard output, using colors. However, not all viewers are able to display these colors. There is a trick to output these results to HTML, preserving formats and colors and making them much easier to view:

- Install the `kbtin` package, which includes the `ansi2html` tool.
- Run `sudo lynis audit system | ansi2html > report.html` and check the output with any browser.

Expected Results: This activity will be complete when you can obtain a Lynis HTML report of your VM.

The *Virustotal* service

Practical application: *You need to scan suspicious URLs to see if they point to a malicious website*

Virustotal is an online service (https://www.virustotal.com/gui/home/upload) that scans any file you upload for known malware using more than 50 different antivirus engines. It also scans URLs to detect the presence of known malware (https://www.virustotal.com/gui/home/url). It can be used to

- Increase your certainty about the security of an unknown executable file or document you are considering opening.
- Try a file from your machine to see if it is infected (that may indicate that a larger infection is currently happening in the system).
- Increase the certainty that an URL you are going to visit is not hiding potential malware.

To perform this activity, you need to upload and scan an executable you choose with the service. Additionally, you also need to inspect any URL you choose.

Expected Results: This activity will be complete when you can obtain reports of an URL or executable from Virustotal.

Google Safe Browsing

Practical application: *You need to know if a website is known as malicious*

Google Safe Browsing scans billions of URLs to locate unsafe websites. It discovers thousands of new unsafe sites daily, many of which are legitimate websites that have been hacked. When it detects unsafe websites, it displays warnings in *Google Search* and in web browsers that use this technology. Additionally, you can manually search an URL to see if it's dangerous to visit a website here: https://transparencyreport.google.com/safe-browsing/search

Expected Results: This activity will be complete when you can obtain a report of any URL you choose from Google Safe Browsing.

Checking known product vulnerabilities using HTTP headers

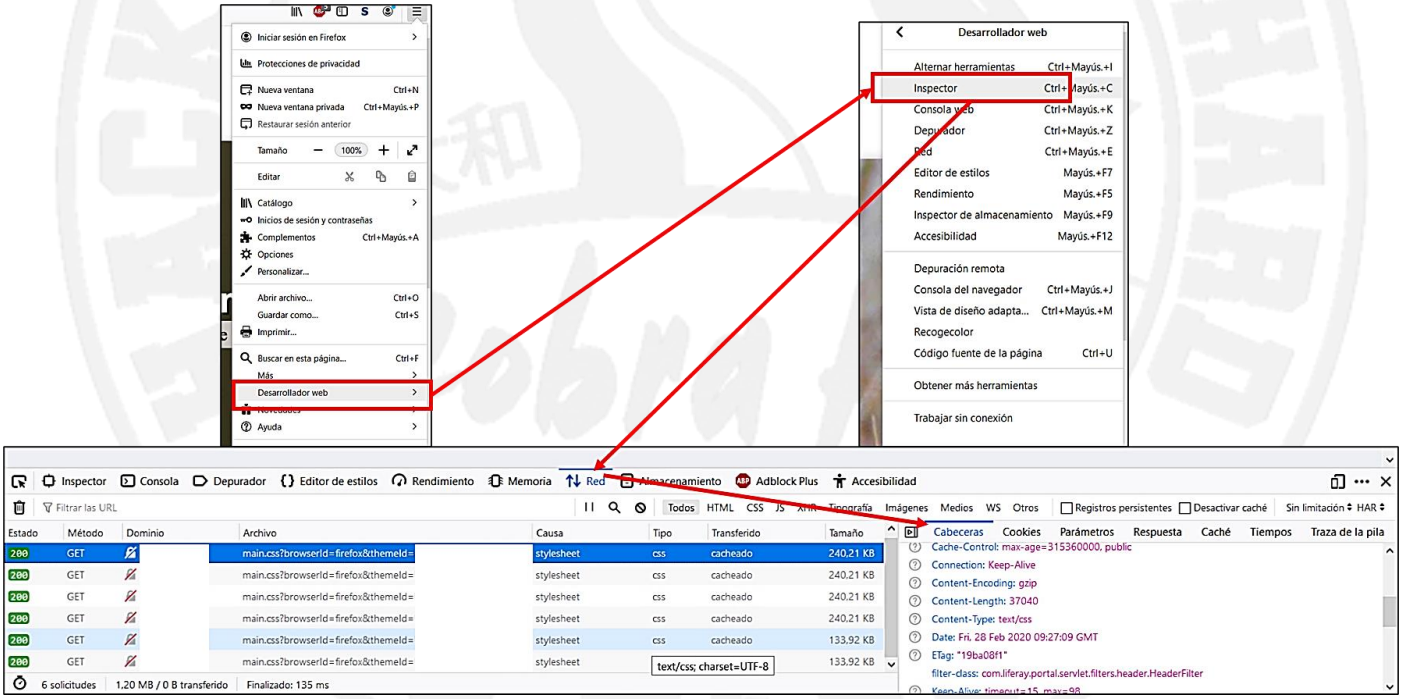
Practical application: *You need to know if any remote web page is running products/frameworks with known vulnerabilities*

Whenever we connect to a website, apart from the content (HTML) we also receive additional information that belongs to the HTTP protocol: the HTTP headers. These headers contain protocol information but, sometimes, they also contain excessive information that we can take advantage of very easily: **product type and version information**. This information may seem trivial, but it can be used to locate the known vulnerabilities of the version of the product that the web server is kindly providing us for free 😊.

You do not even need special software to do this, as the own browser is able to show you this information. The following diagram shows how to do this in Firefox, but other browsers have this option too.

Once you can do that, you only must locate a web page that provides this excessive information to the browser (not all do) and check the information against CVE databases like the ones shown in theory:

- https://cve.mitre.org/
- https://www.cvedetails.com/



This can be a serious security vulnerability, so do not expect major web pages providing you this free information easily (although you never know! 😊).

Expected Results: This activity will be complete when you locate the CVE list of known vulnerabilities of any product you like (preferable from HTTP headers, but any product if you are out of luck finding an "indiscrete" web page).

Tracing package routes

Practical application: *You need to know the routes of the packages you send over the network*

`tcptraceroute` is a tool that shows you the path followed by the packets of a connection (if the network provides this information, if not, package "jumps" will be shown as `*`). Install this tool (`sudo apt install tcptraceroute`) and check how the packets travel through the network to reach any destination you choose. To complete this activity, you should identify which parts belong to your home or laboratory and which parts are on an external network. This will help you to understand better how network connections are performed.

Expected Results: This activity will be complete when you are able to see the package route of any request (if the network blocks this information, it will be considered complete anyway)

See DNS in action: dig

Practical application: *You can ask your DNS server for certain information about any remote domain name*

The `dig` command can be used to consult DNS servers for various information and inspect how the DNS system work, illustrating the theory contents. Using this tutorial: https://www.hostinger.es/tutoriales/comando-dig-linux/ use the `dig` tool to perform these three operations:

1. See the IP corresponding to a DNS name
2. See the DNS name corresponding to an IP (**reverse search**)
3. Trace the different DNS servers that participate in a DNS name resolution

NOTE: If the execution of `dig +trace` does not output more information, try to change the DNS IP address of `/etc/resolv.conf` to `8.8.8.8`.

Expected Results: This activity will be complete when you are able to guess the `dig` options that gives you the three types of information listed.

BADGES AND SELF-ASSESSMENT

NOTE: You have a version of this badge table in an editable format available in the *Virtual Campus*. You can use this file to easily create a document in the format you want and take extended notes of your activities to create a log of what you did. Remember that this material is key to keep track of your self-evaluation.

Badge Level	Unlocked when	Unlocked?
	You can deploy virtual machines in *VirtualBox*	
	You can update an *Ubuntu* system	
	You can create a user on an *Ubuntu* system and put it into the `sudo` group	
	You can change a user password	
	You can check if a password is secure	
	You can answer this question: *What are the advantages of having a strong, non-default, not-known password?*	
	You can properly change between users in an *Ubuntu* system	
	You can install an operational *OpenSSH* server	
	You can activate the `ufw` firewall and restrict all ports but the `ssh` one	
	You understand why `nmap` is a better way of locating alive hosts than `ping`	
	You can trace the package route to any remote system, and identify if the nodes they travel are at your home or in other networks (if the network allows it)	
	You can evaluate if there is an operational connection between your host and your base virtual machine	
	You can enable and use *malware* and *rootkit* scanning software	
	You can enable the signature update and run the *ClamAV* antivirus	
	You can enable the latest version of *Lynis* and evaluate the current security configuration of your base system	
	You can create security reports on an Ubuntu machine with *Lynis* in HTML	
	You can answer this question: *does the Lynis score of the machine improve once you install malware and rootkit detection software?*	

	Because of the previous one, you know the general procedure to add external repositories of software to *Ubuntu* from third party vendors.	
	You can answer this question: According to the Lynis report, *do you think that a base updated Ubuntu installation is secure enough? Why?*	
	You can perform a `ping` scan over a remote server	
	You can analyze an executable with *Virustotal* and interpret its results	
	You can analyze a URL with *Virustotal* and *Google Safe Browsing* and interpret its results	
	You can check known vulnerabilities of any product by inspecting their CVEs. Particularly, you can do this from the information that a web server might provide about their used software products	
	You can answer the following question: *what you should expect about the general state of the security from a webpage that provides product types and versions in their HTTP headers*	
	You can use `dig` to see IP <-> DNS conversions and to trace the DNS hierarchy of DNS servers that are used to resolve a name.	
	**I can do this all day**: You can deploy a base virtual machine, evaluate its security, and perform some initial security-related operations that allow you to better understand the theory concepts	