Source: Bing Chat AI

LABORATORY 1. SYSTEM SETUP

DEPARTMENT OF COMPUTER SCIENCE. UNIVERSITY OF OVIEDO

Computer Security | 2023 – 2024 (V4.0 *"S-81 Isaac Peral"*)

Content

1
2
3
✍️

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: Perhaps 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).***

1

2

3

Exercise L1B1_LOGIN: Login and update the virtual machine

Activity description / Practical application

You need to update your OS software to prevent vulnerabilities

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?*

Additional information required to do it

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 this command.

```
sudo snap refresh
```

Exercise L1B1_USER: Create a proper user

Activity description / Practical application

You need create a user identity for a new system user

Expected results

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

Additional information required to do it

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.

🔍 ***NOTE: 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)***

Exercise L1B1_USERCHG: Properly change between users

Activity description / Practical application

You need to assume multiple user identities in the same work session for certain operations

Expected results

This activity will be complete when user change works without failure

Additional information required to do it

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.***

Block 2: Basic security operations

Exercise L1B2_SSH: Install an *OpenSSH* server

Activity description / Practical application

You need secure remote access to your machine

Expected results

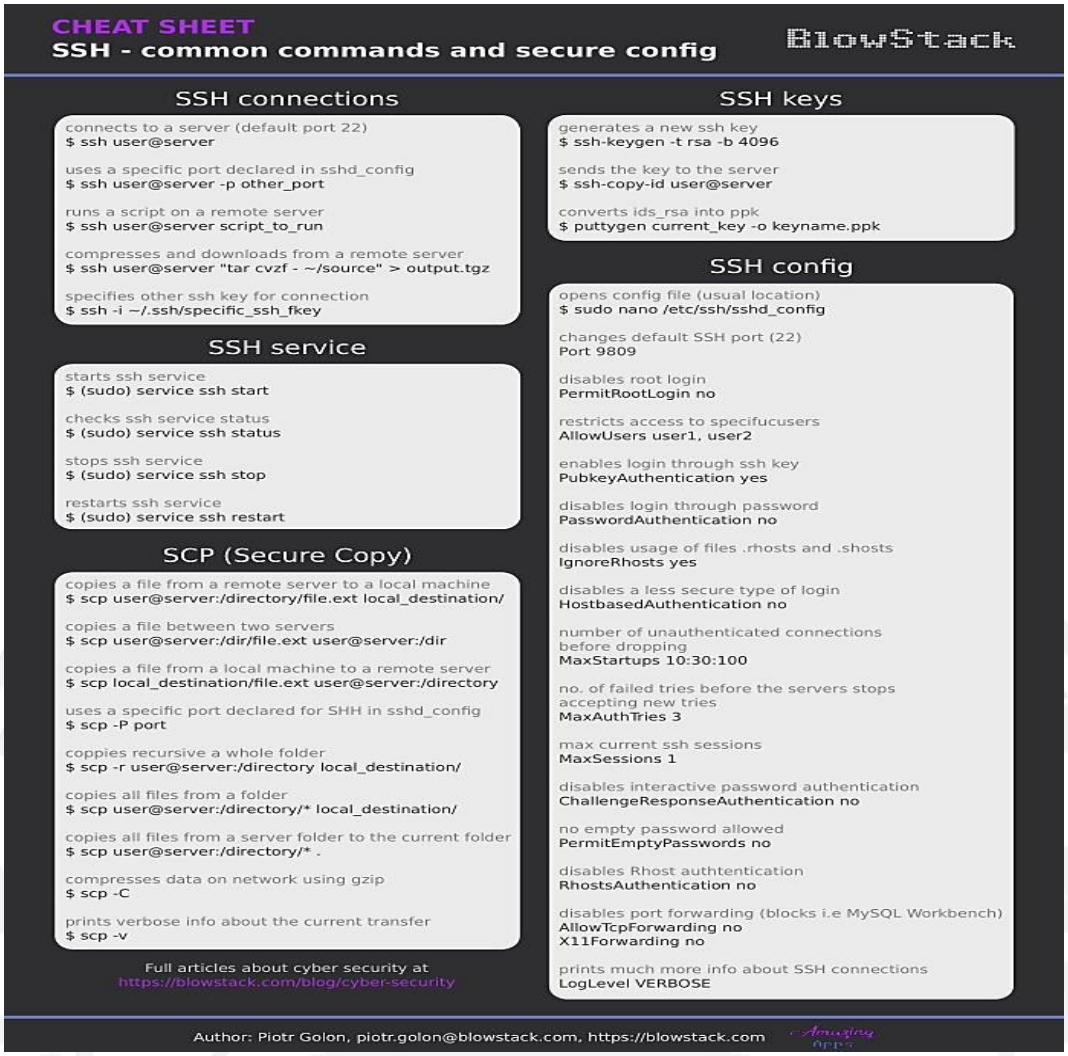
This activity will be complete when you are able to connect to your machine using SSH from your host.

Additional information required to do it

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. If you use the standard command-line client you can use this image as a command guide.



Exercise L1B2_UFW: Enable the firewall

Activity description / Practical application

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

Expected results

This activity will be complete when the firewall reports that it is active, and the proper SSH service port is allowed.

Additional information required to do it

`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**: *VirtualBox* 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, you can block the `ssh` port/service 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)

Use the following image as a guide to how to manage `ufw`. Then check the status of the *firewall* and that the ports have been opened successfully.

Cheatography

firewall Cheat Sheet

by fila2 via cheatography.com/56433/cs/14993/

UFW

sudo ufw status	Ver el estado
sudo ufw default {deny, allow} {incoming, outgoing}	Programar reglas por defecto de entrada y salida
sudo ufw allow <nombre de servicio>	Permitir tráfico por servicio
sudo ufw allow {in, out,_} <nº puerto >	Permitir tráfico por puerto
sudo ufw allow <nº puerto> {tcp, udp,_}	Permitir tráfico por puerto para un tipo de paquetes TCP/UDP...
sudo ufw app list	Listar apps posibles de configuración
sudo ufw allow from <dirección ip> to any app <nombre app>	Abrir un puerto para una app y unas direcciones ip determinadas
sudo ufw delete <servicio>	Borrar una regla
sudo ufw reset	Borrar todas las reglas
sudo ufw status numbered	Listar reglas de manera numerada
sudo ufw delete <nº regla>	Borrar regla por número
sudo ufw enable	Encender el firewall
sudo ufw disable	Desactivar el firewall
sudo ufw status verbose	Ver configuración completa

LOAD AVERAGE

top uptime htop w cat /proc/loadavg	Averiguar Load average
cat /proc/cpuinfo | grep "procesador"	Averiguar nº de procesadores
cat /proc/cpuinfo | grep "cores"	Averiguar nº de núcleos

Procesos

top	Muestra los procesos que están corriendo y la posibilidad de modificarlos. Muestra valores como el uso de CPU,memoria,swap y presenta una interfaz simple que cuenta con varias partes.

Procesos (cont)

htop	Similar a top, muestra información referida a ellos como el id del proceso y permite mandarles órdenes y matarlos
ps -aux	Muestra todos los procesos activos del sistema
ps -fax	Procesos activos en forma de árbol
lsof	Lista los archivos que están abriendo los procesos
netstat	Listado de las conexiones activas
netstat -puta	Muestra los programas que usan los puertos
sudo apt install iptraf-ng	Captura paquetes de red y nos da información
sudo apt install iftop	Visión continua e interactiva del tráfico de red
sudo apt install iotop	Monitorizar y visualizar en tiempo real las E/S del disco y procesos
sudo apt install bmon	Monitorización y depuración de red
sudo speedometer <-rx, -tx> <interfaz>	Monitoriza el tráfico de red o la velocidad/progreso de una transferencia de archivos

Almacenamiento

df	Información de uso de disco duro
du	Ver el uso de disco de un directorio. Devuelve los bloques de datos (por defecto de 1Kb cada uno) que consume cada objeto
du -{k, b, m,} <directorio >	Para especificar el tamaño de los bloques se usa b (bytes), k (kilobytes), m (megabytes)
du -ah <directorio >	mostrar los tamaños de carpetas y archivos
sudo apt install discus	Información almacenamiento del sistema

By **fila2**
cheatography.com/fila2/

Not published yet.
Last updated 6th March, 2018.
Page 1 of 1.

Exercise L1B2_LYNIS: Evaluate base VM security configuration: *Lynis*

Activity description / Practical application

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

Expected results

This activity will be complete when you can **obtain a *Lynis* score** of your system

Additional information required to do it

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 3.0.9 or higher)
- Evaluate the base system using *Lynis* using the appropriate option (https://cisofy.com/documentation/lynis/get-started/#first-run). It is recommended to redirect the output to a file to see the results. Results are best viewed (colors) using `more` over the file.

Exercise L1B2_NMAP: Basic enumeration with *nmap*

Activity description / Practical application

You need to locate remote “alive” machines in any network

Expected results

This activity will be complete when you can tell if a remote machine is "alive" or not using `nmap`. You can answer this question: *Why do you think `nmap` is a better choice than ping when it comes to locating "live" hosts and their characteristics?* (try pinging any hosts and see what happens)

Additional information required to do it

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 `nmap` 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)

```
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
```

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*

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

Block 3: Additional basic security operations

Exercise L1B3_PWD: Put a proper password

Activity description / Practical application

You need to change the password of any user with a proper and secure one

Expected results

This activity will be complete when your chosen password is **reported as very secure**

Additional information required to do it

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/

Exercise L1B3_MALSCAN: Enable *malware* / *rootkit* scanning software

Activity description / Practical application

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

Expected results

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

Additional information required to do it

It is important for a secure system to be able to detect and stop potential malware / rootkits that may be present in it. 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

Rootkits. 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>`

Exercise L1B3_LYNISREPORT: *Lynis* reports

Activity description / Practical application

You need to obtain a security report of the current hardening level or your machine

Expected results

This activity will be complete when you can obtain a *Lynis* HTML report of your VM, and you can correctly locate in the report the number that describe the system security level

Additional information required to do it

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.

Exercise L1B3_VIRUST: The *Virustotal* service

Activity description / Practical application

You need to scan suspicious URLs to see if they point to a known malicious website

Expected results

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

Additional information required to do it

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

Exercise L1B3_USERGSAFEB: *Google Safe Browsing*

Activity description / Practical application

You need to know if a website is identified as malicious by *Google*

Expected results

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

Additional information required to do it

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 its website here: https://transparencyreport.google.com/safe-browsing/search

Exercise L1B3_CVEHTTP: Checking known product vulnerabilities using HTTP headers

Activity description / Practical application

You need to know if a web page is running products/frameworks with known vulnerabilities using no special tools

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).

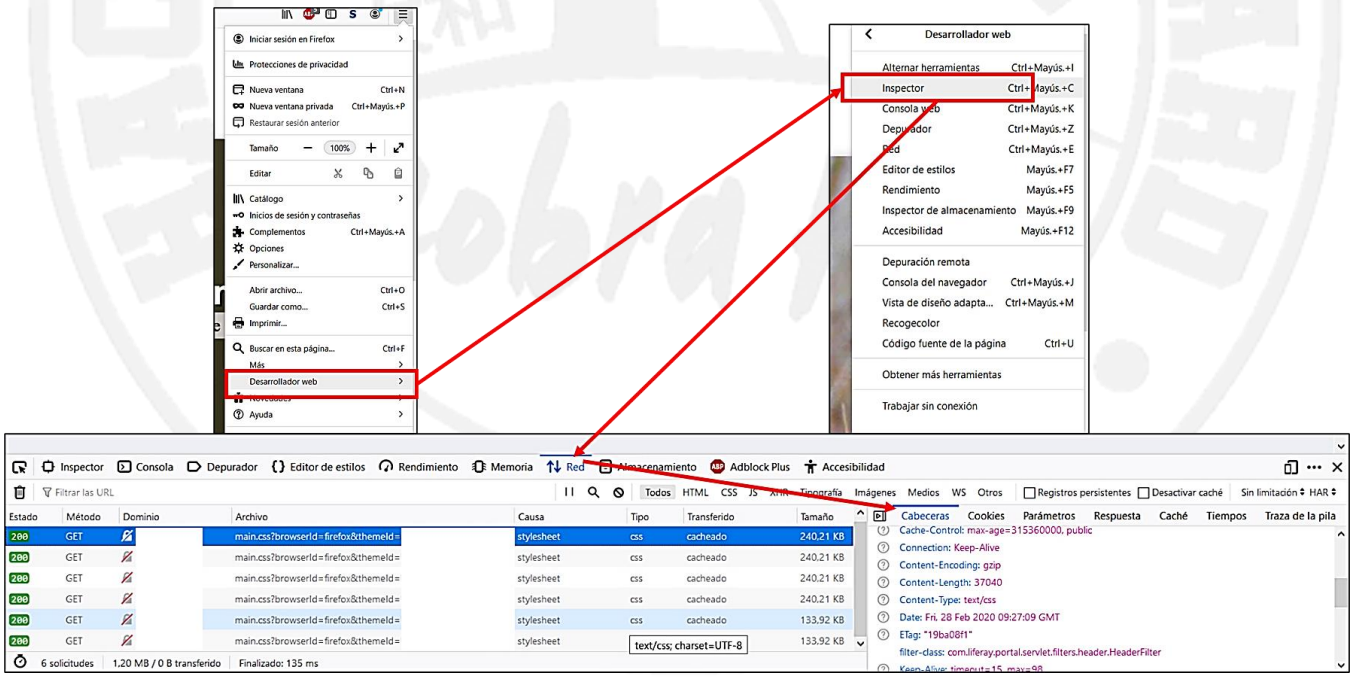
Additional information required to do it

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 versions of the software products that the web server is running, and this information is being provided by the server 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 excess of 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 web pages of major companies providing you this free information easily (although you never know! 😊).

Exercise L1B3_NEXTDNS: Using a secure DNS for your own protection: *NextDNS*

Activity description / Practical application

You can use the *NextDNS* service for free, even without creating an account, to be able to browse much more securely in a transparent way, to understand the enormous importance of the DNS service in your browsing.

Expected results

The goal of this exercise is to make you realize that **without a DNS server** that resolves your requests to IPs, **you wouldn't really be able to browse**. Therefore, if that resolution service **has "intelligence" that prevents browsing websites that are known to be harmful**, we could achieve a more secure website (even with a risk of censorship). *NextDNS* is this, but it also allows you to control that "intelligence". Once you try it, the idea is that you will be able to answer these questions:

- *Do you understand that by using NextDNS you are implementing a large number of security measures in a transparent way (without you being aware of it)?*
- *Which of the services provided by NextDNS do you think you might find most useful?*
- *Have you noticed any difference when browsing (errors, speed...) or any problem with a website?*
- *Based on your experience with the service, would you install it in your home? Would you do it on a virtual machine that you have for "safe browsing" along with other security measures?*

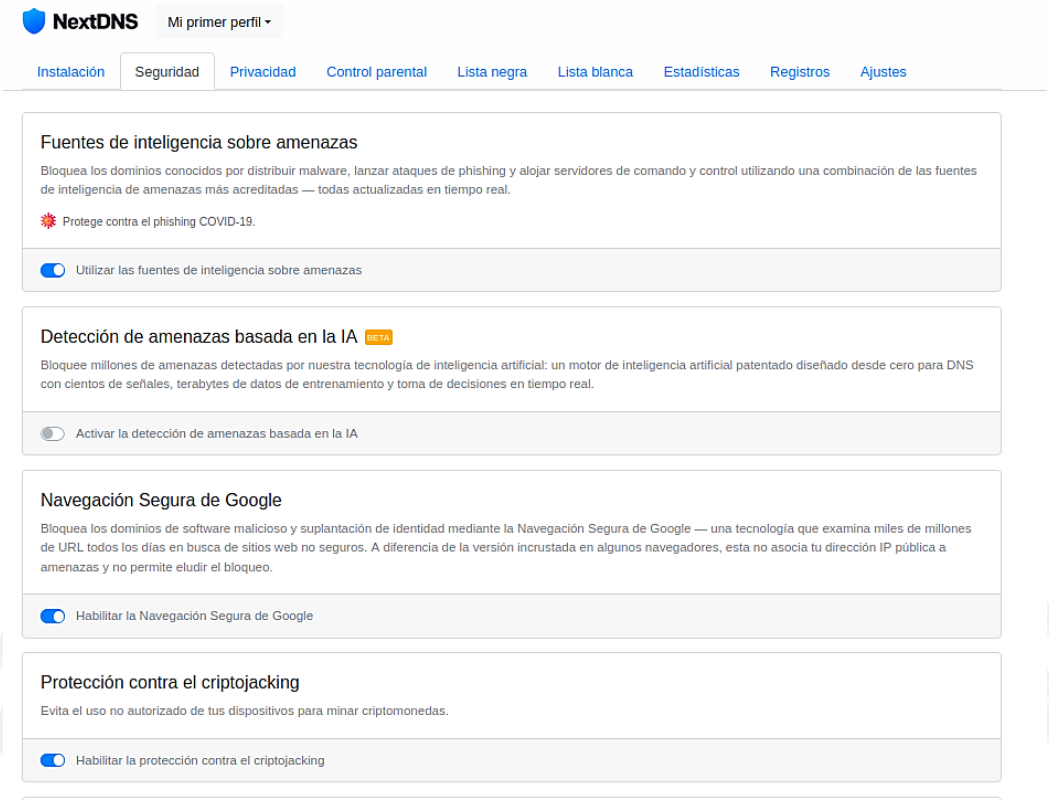
Additional information required to do it

The *NextDNS* (https://nextdns.io/) service is really a way to get rid of a lot of problems in a transparent way, and that can be used to have a much less trouble-prone browsing. Once configured in browsers or machines, the service will automatically block **any type of access, either direct or indirect** (because another page loads it) to domains that are considered threats, or that belong to advertising companies, that are not approved by this centralized DNS service.

The advantage of this service is that you don't even have to make an account to use it, and if you do, it's to save your particular settings. You also don't need to install anything if you don't want to. While the free service is limited to 300,000 requests per month, after which it will continue to serve you web pages, but you won't have protection, that volume of requests is enough for normal browsing by any user with basic knowledge.

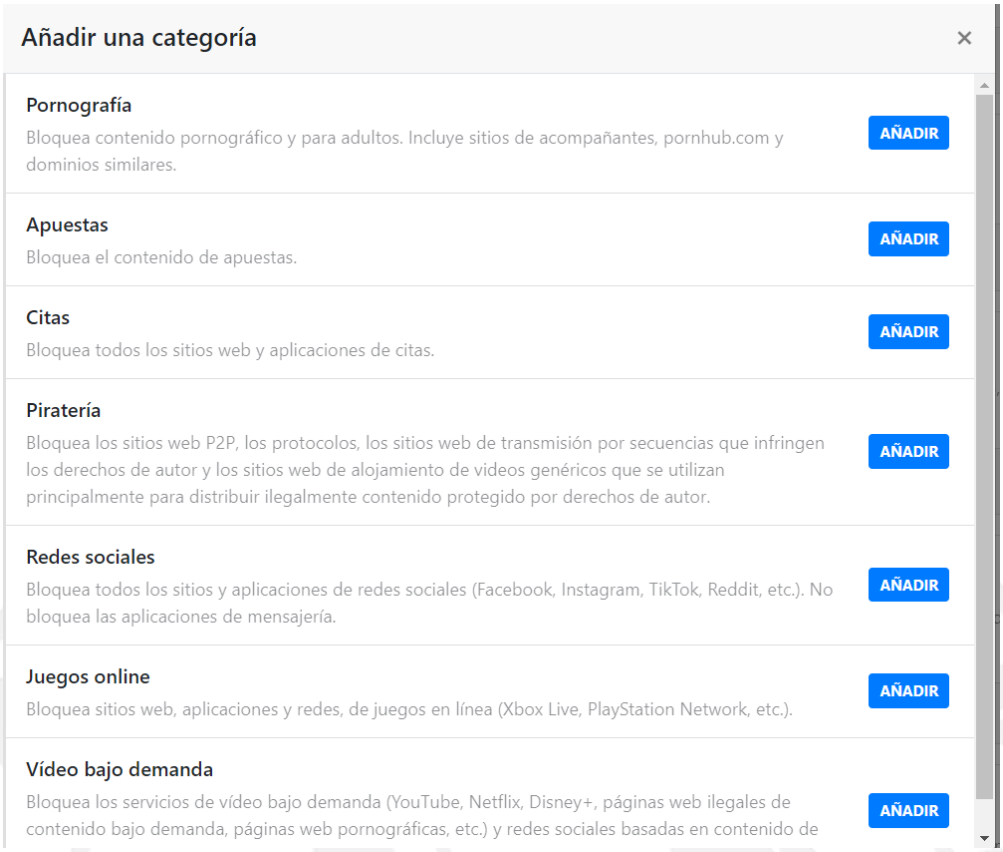
To use this service, we have to go to its website and use the "***Try it now***" option on their home page, after which we will go to a URL that looks like this: `https://my.nextdns.io/<random code of letters and numbers>/setup`. In that URL, you can configure the security options of the service and see the installation options on your machine if you finally want to use them.

As you can see, in the first ***Security*** tab we have a series of measures active by default, and it is recommended to activate all of them initially to see how it behaves with our usual browsing:

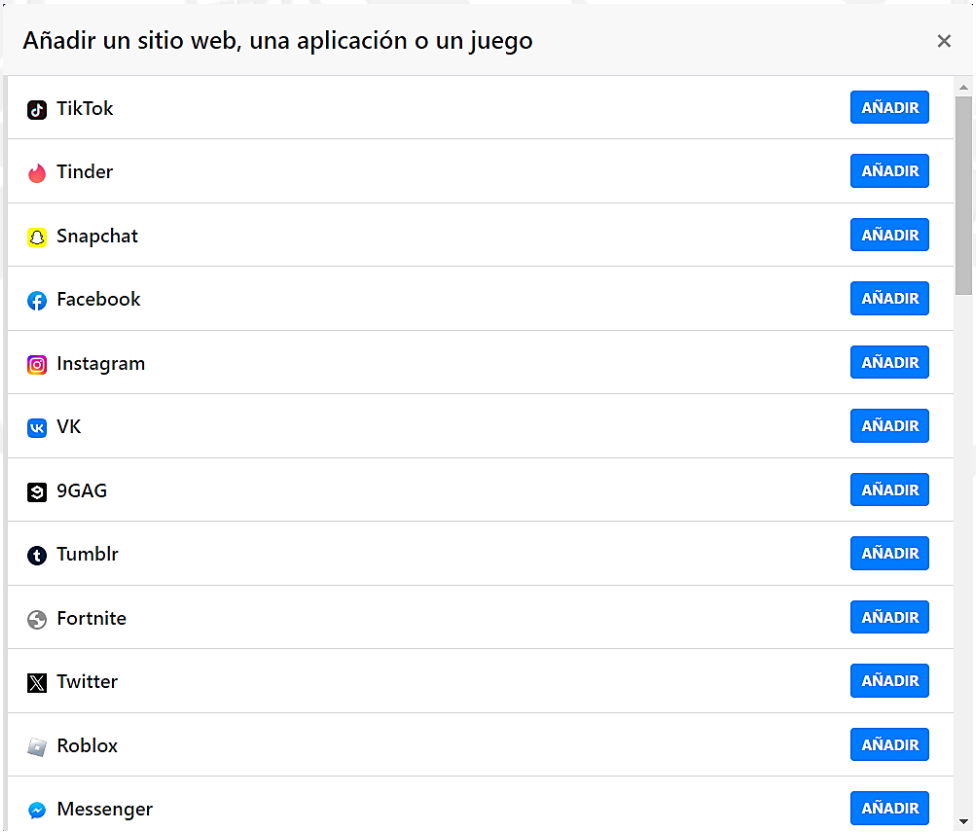


There are a large number of measures that we can activate, so it is recommended to go through the entire section and activate all of them initially. Keep in mind that each measurement has an explanation of what exactly it is. Even if we don't understand it, it is recommended to activate them, and only deactivate them if there is a problem in our usual use. The exercise consists of **going through all the options** *that NextDNS* gives us and activating all the ones that we think are useful for our particular use case:

- The option **to block TLDs** is used to block access to pages whose domain ends in a specific prefix (`.es`, `.com`, etc.). You can find a full list here: https://en.wikipedia.org/wiki/List_of_Internet_top-level_domains.
- In the **privacy** section, we can find a very useful feature: **block lists**. They are basically lists of domains that the service will not resolve because they belong to ads, malicious pages, or intrusive or uninteresting services that have been deemed harmful in some way. Although the service comes with one already loaded, we can improve it by using lists that are available over the Internet created for *NextDNS*. You can find some of them at this link: https://github.com/hagezi/dns-blocklists.
- *NextDNS* also has a **parental control** section where you can limit your search and Internet usage to the times and classified domains you want. If you're going to use this option, keep in mind that it implies that the web pages you navigate **to must be classified by *NextDNS* within the category to which they belong**, or specify it (and not all of them do). The **list of site categories** used by *NextDNS* is very eloquent:



- The **access limitation to applications and games** consists of selecting some from a list that the service has and that we want to prohibit.

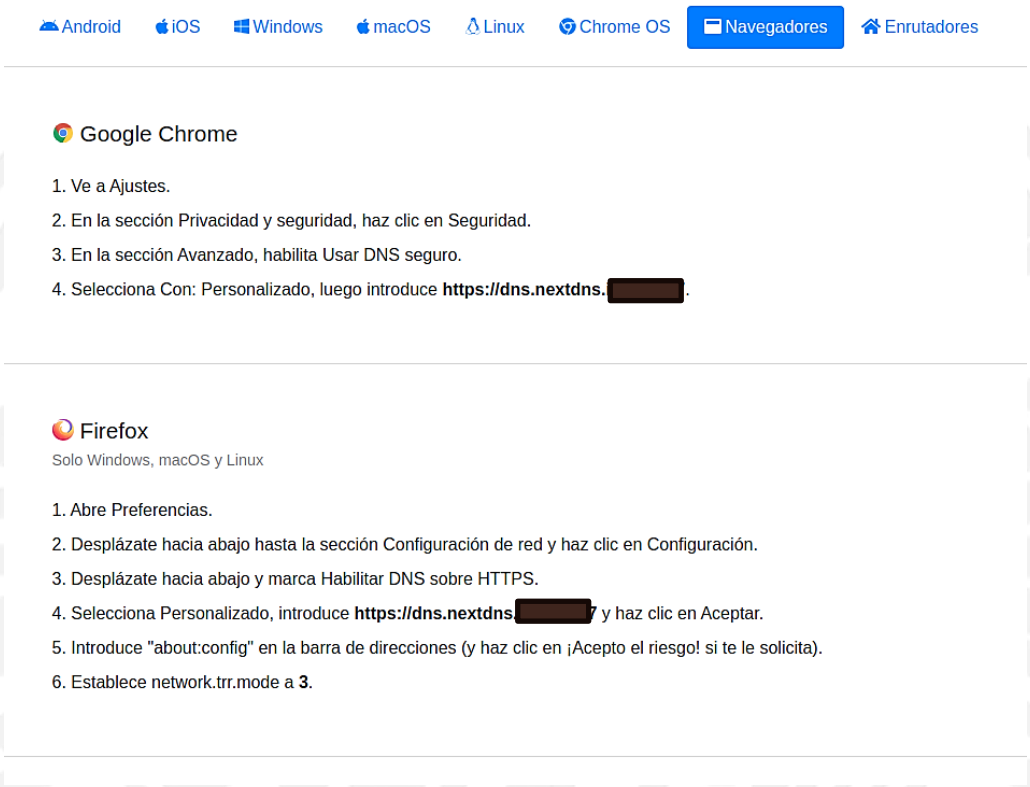


- **The rest of the service** are block lists of specific sites that we want, lists to allow sites regardless of whether they fit into any restrictions that we have set, and statistics of blocking and use of the service so that we have control of it.

- In the **installation** section, we have instructions to launch the service on any operating system (PC or mobile), specific browsers (if we don't want to do it for the entire operating system) or even on routers, but only specific models that allow the execution of the commands they indicate.

Keep in mind that **a new configuration is generated** (different block of numbers and letters in the URL) every time you enter the service with the button to test it that we have seen. **Write down your identifier** so you can continue to configure the service later, and keep in mind that settings in browsers or OSs use that identifier to distinguish your settings from others.

If you want to start with something simple to test how it works, it's best to **configure one of your browsers** with only the settings you've put in and see how it goes.



✍️ 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 can be brought to lab exams.

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 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 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 understand how to configure a machine or browser to use *NextDNS* and all the protection features it provides	
	**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	