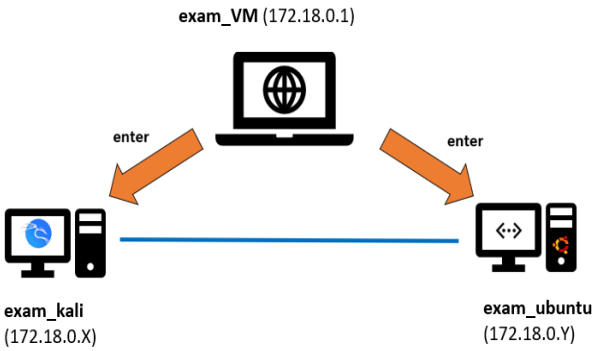
Computer Systems Security. Lab Exam. Ordinary Call 2022

Name and Surname:
UO:

Instructions:

The following diagram shows the infrastructure that is provided for this exam, consisting of **a VM identical to the one used in the labs** with 2 also similar containers. To access them, on the desktop of the virtual machine you will see an `exam.zip` compressed file **that you must unzip** with the password that will be said at the beginning of the exam. Inside it, you will see the `exam/lab_exam` folder. **The contents of this folder contain the laboratory of this exam, composed of:**



- A *Kali* container.
- An *Ubuntu* container with a copy of our school's website.

The containers are already built, so it **is enough to run `build_lab.sh` to start up the infrastructure** (**IT IS NOT NECESSARY to run `prepare_lab.sh`**). As in the labs, you can log into both with the provided `enter_kali.sh` and `enter_ubuntu.sh` scripts.

The IPs of the three components of our infrastructure are those that appear in the image, although X and Y can be 2 or 3 depending on the order in which the containers are started. **Remember that you can bring all the material you developed in the labs as an aid to do this exam.**

Delivery:

You must submit **a single PDF file** in the task enabled on the virtual campus with the **necessary screenshots and explanations** that show that you have done each exercise. It is recommended to open a *Word* file at the beginning of the exam and write screen captures and explanations as you do exercises, saving often to avoid losing information. You can then **save the file as PDF** for its delivery.

Exercises:

1) **On the web page hosted in the *Ubuntu* container, a clueless administrator has misused the standard mechanism that exists to prevent *Google* or other search engines to index certain pages of a website, pretending to hide a secret data with it. Locate that mechanism and what he is hiding (1 point).**

2) **Bonifacia (also known as `ssiuser`) intends to send a message to Anacleto expressing his eternal love, and wants to use the public key infrastructure to do it. Bonifacia has Anacleto's public key, but he is very ignorant, and sent his public key file encrypted with a symmetric key to her. You can find this encrypted public key file in the `/exam` folder, called `clave_pública_anacleto2022.asc.asc`. Despite his blind love, Bonifacia is aware of Anacleto's limitations and knows that he always uses the same *password* for everything, even to authenticate himself in the system. Demonstrate that you can implement the following procedure to send the love message (1.5 points)**
   - **Create, through `useradd`, an `anacleto` user in the VM whose hashed password is `'$1$marina22$FNBWK1hFyaa4U.BEdgTL71'`. You can find this hash inside the `/exam` folder, in the `hash_anacleto.txt` file. No parameter is necessary other than the password and the name of the user, but BE CAREFUL!, use single quotation marks (strong quotation marks), to supply the password.**
   - **Attack using `john` the system files as we did in the labs to find out the password of `anacleto` on the VM. To do this, use the file `diccionario.txt` in the `/exam` folder. `john` is installed both in the VM and in both containers, choose the one you want for it.**

- **Decrypt, with the password obtained in the previous step, the `anacleto` public key.**
- **Import the public key obtained from `anacleto` and encrypt Bonifacia's declaration of love (use a simple .txt file 😊) for the aforementioned secret agent.**

3) Using the help of the *Ubuntu* CIS guide if you deem it necessary (provided as additional material in the `exam` folder), perform the following manual security operations on the virtual machine (1.5 points)

a. **Change the SSH server settings so that:**
   i. **Graphical mode access is disabled.**
   ii. **The maximum number of authentication attempts is 6.**
   iii. **The maximum number of sessions is 3 or less.**
   iv. **The login with the `root` account is deactivated.**
b. **Assess the security level with `lynis`.**

4) In the `exam` folder of the virtual machine you have a ZIP file called `U_CAN_Ubuntu_18-04_LTS_V2R5_STIG_Ansible.zip`. It is an STIG applicable with *Ansible,* identical to the one seen in laboratories. The machine also has the `oscap` package and its *Oscap Workbench* GUI already installed. To complete the exercise you need to (1.5 points)::

- **Evaluate the current state of the VM with *Oscap Workbench* and the CIS profile, as we saw in lab classes and deliver a screenshot where the final score (percentage) obtained is seen. In the `exam` folder you have the file `scap-security-guide-0.1.59-oval-5.10.zip`, equal to the one we used in labs.**
- **Clone or take a *snapshot* of the VM, so that you can return to the initial state if necessary.**
- **Apply the security controls contained in the STIG ZIP, writing in the results report what you have done to do it. *Ansible* is already installed on the virtual machine.**
- **Reevaluate the VM with these controls applied using the same profile as above, and display the final new score (percentage).**

5) To do this exercise the virtual machine has `firewalld` and its GUI seen in labs already installed. Using these tools do the following (1.5 points):

- **Enter in the console of the `exam_ubuntu` container and verify that you can `ssh` to the virtual machine**
- **Open the firewall GUI and, in the `docker` zone, forbid the communication of the previous container with the virtual machine for that `ssh` service, only under IPv4, indicating in the results how you have done it.**
- **Do the previous step again, but checking now that the connection is not possible, and that the *firewall* is doing its job.**

6) On the `exam_ubuntu` container there is an *Apache* web server with `mod_security2` installed. This question only requires you to prove that the `mod_security2` engine works, by manually adding a new rule to the WAF that fires when the value of the "rick" parameter in a URL is a string containing the word "pickle". The WAF must return an error page with the HTTP status *406 Not Acceptable*. In addition, the incident must be recorded with the following message "I'm a pickle Rick!". Fire this rule on purpose (1.5 points)

7) The `ssiuser` user, unconsciously decides to activate the SUID bit to the `'date'` command, with the idea that it can be executed with privileges, even if they are not available. A bad guy, after studying how you can use common *Linux* commands to do Stranger Things, learns that the `date` command has the SUID bit enabled. Demonstrate how you could access the contents of the file where the passwords of system users are stored in these circumstances. In the `/exam` folder you have a file that can help you to do it (1.5 points)