Lab 01 - Introduction

Objectives

- Simple CTF tasks
- Introduction to basic security-related tools
- Simple program compiling tools
- Basics of networking related monitoring tools

Preparation

You will solve this lab inside a virtual machine on openstack [https://cloud-controller.grid.pub.ro]:

1. Log in with your LDAP credentials
2. Create a RSA key pair on *fep.grid.pub.ro* (ssh LDAP-USERNAME@fep.grid.pub.ro; ssh-keygen -t rsa -b 4096 -C "your_email@example.com")
3. Make sure you have the public key configured on openstack in Project → Compute → Key Pairs
    - Passphrase authentication will be disabled on the SSH server
    - The key that you provide will be automatically registered in *.ssh/authorized_keys*
4. Create a new Instance with the following parameters:
    - Flavor: m1.small
    - Instance Boot Source: Boot from image
    - Image Name: ISC 2020
    - Key Pair: the one you just added
        - **NOTE:** if you have *only one* key pair configured, it will be chosen by default; otherwise, make sure you don't skip this step
5. From the fep console (ssh) login into the newly created VM (ssh student@VM-IP)

Recording tutorial

Due to the fact that we have to work remote, please make sure that you record your screen while working. Here is how.

The VM does not have asciinema install, please run the following command to install it.

```
root@host:/home/student# apt-get install asciinema
```

```
# start the recording after you ssh into the machine
root@host:/home/student# asciinema rec lab01_mihai.cast
[...]
# !!!IMPORTANT before you start working echo your name in the terminal!!!
root@host:/home/student# echo "Mihai Chiroiu's terminal!"

# stop recording
root@host:/home/student# exit
asciinema: recording finished
asciinema: asciicast saved to lab01_mihai.cast

# upload the recording
ASCIINEMA_API_URL=https://asciinema.cs.pub.ro asciinema upload lab01_mihai.cast
```

When you finish your work, submit the details on the form [https://forms.office.com/r/GZzRJVqQuy]. Double check to see if all is good (form responses [https://ctipub-my.sharepoint.com/:x:/g/personal/mihai_chiroiu_upb_ro/Ee4pZApRKA5Iq9JgR2r652QB0FVL4J9EFtTBtva3jX-1Lw?e=ylQ7lu])

CTF local tasks

Download the task archive for this section. Each exercise will have a corresponding folder.

01. [10p]B64 encoding

- The flag is in b64.txt. It should look something like this: **FLAG{...}**.
  - **Hint:** python3, base64

02. [10p]EXIF

- The flag is hidden somewhere within this image. Remember its format.
  - **Hint:** it's not steganography; don't look at the pixels

03. [10p]From Manchester with love

- Remember RL? Remember Manchester [https://en.wikipedia.org/wiki/Manchester_code#Encoding]?

04. [10p]Corrupted file

- The header seems to be damaged…

Up for more?

- CTFlearn [https://ctflearn.com]
- OverTheWire [https://overthewire.org/wargames/]
- Cryptopals Challenges [https://cryptopals.com/]
- PicoCTF [https://picoctf.com/]

OS Management

05. [10p]Web server & console browser

- Install and configure **apache2** and **links**. Use the latter to connect to http://localhost [http://localhost]
  - **Hint:** use the distro specific package manager.

06. [10p]Disk space & usage

- Display the disk space usage for each individual directory (. and .. excluded) in the first two hierarchical levels of */usr/include/* in a human readable format
  - **Hint:** find, du
- Sort the list in ascending order, by size

Program compilation tools

07. [10p]Program compilation

- Download the following program [https://curl.haxx.se/libcurl/c/simple.html] and compile it using **gcc**.
- What is the program intended for?
- Modify the program such that it connects to "http://localhost [http://localhost]" (i.e. your local apache server) and prints the response (apache's default HTML test page) to stdout, just like standard curl.
    - **Hint:** you need to install libcurl's development libraries.
    - **Hint:** you need some flags for the compiler to know where libcurl is installed (see library's documentation [https://curl.haxx.se/libcurl/c/libcurl-tutorial.html])

08. [10p]Static compilation

- Statically compile the program (but keep a copy of the old, shared executable).
    - **Hint:** curl-config --static-libs
    - **Hint:** Note that you'll need even more development libraries: libidn11-dev librtmp-dev libssl-dev libidn11-dev librtmp-dev libssl-dev libcrypto++-dev libkrb5-dev libldap2-dev libnghttp2-dev libpsl-dev
    - **Hint:** Getting a pthread-related linker error? Try *-lpthread* at the end of the gcc command!
- Check the size difference. What does it mean?
    - **Hint:** ldd
- Uninstall libcurl and see which of the executables successfully run now!
- Reinstall curl again if you need it ;)

Networking related tools

09. [10p]Traffic sniffing

- Use the tcpdump suite to save all the traffic from interface ens3/eth0 to a file.
    - **Hint:** Tcpdump may complain that it has no privileges to write the log file. Use "-Z student" (man!) to reacquire them.

10. [10p]Logging & Auditing

- Write an **iptables** rule that logs all the traffic generated by curl.

Feedback

11. [10p]Feedback

Please take a minute to fill in the feedback form [https://forms.gle/BugCwG6GNkdq5DTg7] for this lab.