Lab 5b:

Password Cracking Lab

This lab uses the **Ubuntu 20.04** virtual machine (VM) as an OVA file **cscelab.ova** on Canvas. The credentials are as follows:

Username: sec-lab

Password: untccdc

**NOTE 1:** For your lab report, you need to reply questions which are marked in red.

**NOTE 2:** Make sure that your hostname is your EUID with a ‘lab’ suffix and for your default user, use your EUID. For example, assuming your EUID is ab1234, the computer name (hostname) would be ab1234lab and the user would be ab1234.

**NOTE 3:** When it is required to “attach a screenshot”, feel free to attach several screenshots if needed. If the screenshot requires your terminal and commands, you need to make sure that your hostname and username are clearly visible.

**NOTE 4:** When you are running a command, you have to be extra careful. Especially when you are applying an attack, if you provide a wrong command or parameter, you may cause a bigger harm than expected. Therefore, you are responsible for your commands.

# Objectives

The objective of this lab is to apply both brute-force and dictionary attacks using a famous password cracker tool called Hydra.

Our scenario will be as follows: You are an attacker who has access to the computer and you are interested in finding the password of the other user(s). You observed that there is SSH login only; so, you will try with brute-force attack and also the dictionary attack.

Hydra supports many protocols, including (but not limited to) FTP, HTTP, HTTPS, MySQL, MSSQL, Oracle, Cisco, IMAP, VNC, and many more. One should note however that this type of attack may be detected due to multiple login attempts.

We can use Hydra through the terminal or the GUI application (xHydra); however, the terminal gives us more flexibility and control. Therefore, the lab will contain only terminal based work from now on.

# Brute-Force attack

First of all, we will try brute-force attack, which requires trying all possible combinations. As you can imagine, this is a time-consuming process, especially if you don’t have any clues on the password.

Our victim will be the user: test. Assuming that we don’t know its password, we will be trying all the possibilities. As this is not feasible for a short amount of time, instead we will do an educated guess: i.e., the password for the user test has only three lower case letters.

Hydra runs as follows: hydra <parameters> <IP> <PROTOCOL>. Since we are applying attack to another user on the same computer, we will use 127.0.0.1 for IP and for the PROTOCOL, we will use SSH. You can find all the possible parameters for Hydra through ‘man hydra’, but we are going to run using multiple threads (-t <#THREADS> and we can use 4), declare the username (-l <USERNAME> -- in our case, the username is test), and we will ask Hydra to generate all the combinations for us through -x parameter (-x MIN:MAX:LETTERS format where min and max number of letters are given and then the type of the letters are provided. For our case, we know that it is using three letters only so both min and max will be 3. And, we know that it only has lower-case letters, we will use ‘a’. Hence, our input will be 3:3:a. If we had upper case only, then it would be 3:3:A or we had lower and upper case three characters 3:3:aA. We could even include numbers, such as 3:3:aA1). We are also interested in the execution time, so we are going to note how long it takes to try using time command. In conclusion, our command will be as follows:

**time hydra -t 4 -l test -x 3:3:a 127.0.0.1 ssh**

*When it finds the password, you can exit the execution by pressing CTRL-C.*

**Q1: Provide a screenshot of your running code.**

**Q2: How long did it take to run?**

**Q3: What is the found password?**

# Dictionary attack

Not always we are this lucky to get the clues for the password. However, there are other options. For example, when we search a bit, we can see that there are some commonly used passwords (they are found through studies after breaches occur). You can find multiple lists online. Now, we will try a commonly used password list instead of brute-force attack (-P <password\_list>). Hence, following is our command now:

**time hydra -t 4 -l test -P /opt/hydra/common\_passwd.txt 127.0.0.1 ssh**

**Q4: Provide a screenshot of your running code.**

**Q5: How long did it take to run?**

**Q6: A bit of research time: Search the web and find the most commonly used 10 passwords. What are they? What is common for them?**