FIRST TASK- User Creation: Start by creating a new user in the Linux system. Remember to assign the user a password for subsequent login purposes.

sudo adduser newuser # Replace 'newuser' with the desired username

SECOND TASK- User Switch: Once the new user has been created, switch your session to that user.

su - newuser # Switch to the newly created user

THIRD TASK- Folder and File Creation: As the new user, create a new directory. Create a file and write a simple message inside this directory. Your message could be something like "Hello from [your username]!". Remember to save the file before proceeding.

Code is

mkdir ~/newdirectory

echo "Hello from newuser!" > ~/newdirectory/message.txt

HOWEVER…

Once signed into new user it doesnt let me make the new directory…

Check Current User's Permissions: Before attempting to create a new directory, ensure that the current user (newuser) has the necessary permissions to create directories in the desired location. You can do this by checking the permissions of the parent directory:

ls -ld /home/\*newuser\* # asterisk meaning sub for name of new user

Grant Permissions: If the current user doesn't have write permissions to the parent directory, you can grant them using the chmod command:

sudo chmod u+w /path/to/parent/directory

Once again mine didn’t let me even do that…

sudo chmod u+w /home/dominic

[sudo] password for dominic:

Sorry, try again.

[sudo] password for dominic:

dominic is not in the sudoers file. This incident will be reported.

(this was my error)

A computer screen shot of a program

Description automatically generated

To resolve this issue, you'll need to add the user dominic to the sudoers file, which grants them permission to execute commands with superuser privileges.

Log in as a user with administrative privileges.

Edit the sudoers file using the visudo command, which ensures that the syntax is correct:

sudo visudo

In the sudoers file, locate the line that grants sudo privileges to users

# User privilege specification

Below this line, add the following line to grant sudo privileges to the user dominic:

dominic ALL=(ALL:ALL) ALL

Save and close the sudoers file.

This will allow the new user to read, write and execute

A screenshot of a computer

Description automatically generated

Folder and File Creation(continued)

We go back into the new user so

su dominic

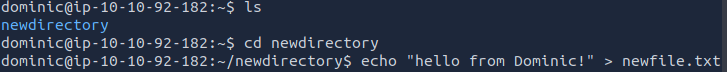
mkdir newdirectory # Create a new directory

cd newdirectory # Navigate to the newly created directory

echo "Hello from newuser!" > newfile.txt # Create a new file with a message

This command will create a file named newfile.txt with the message "Hello from newuser!" inside the newdirectory.

A blue background with white text

Description automatically generated

Now we switch back to original user

exit

I went back and did it all over to clean it up a bit still stumbled but here we are

A computer screen shot of a program

Description automatically generated

End of Basic User and File Management

File System Flags

In this challenge, you will navigate a complex file system environment to find hidden flags. These flags might be buried within folders or in front of your eyes.

Remember, your ability to locate and retrieve these flags will demonstrate your understanding and competence in navigating and manipulating the Linux file systems, an important skill in cybersecurity. There are four file system flags - one of them is find\_flag.txt All of the flags in this CTF will have the same structure - {T415\_15\_a\_te5t}; keep an eye out for them

This is the first flag I found

A computer screen shot of a computer screen

Description automatically generated

Flag #2

A computer screen shot of a program

Description automatically generated

Flag #3

(Dont go in run away)

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

Flag #4

(Y0u\_G0T\_1t}

A blue background with white letters

Description automatically generated

Next task

Your task is to find flags hidden within the webpages. For instance, a flag might be hidden in the HTML comments or in a hidden page.

find /\* -iname "Flag\*" 2>/dev/null A screen shot of a computer

Description automatically generated

find: Initiates the search command.

/: Specifies the starting directory for the search (in this case, the root directory).

-iname "Flag\*": Specifies the search criteria. -iname means the search is case-insensitive. "Flag\*" specifies that the search should match any file or directory name starting with "Flag".

2>/dev/null: Redirects any error messages (stderr) to /dev/null, effectively discarding them.

cat /srv/ftp/flag.txt

A computer screen shot of white text

Description automatically generated

Password not what I though. Gotta try to crack this password.

Want to find all the files that have curly brackets since that’s how the main format is laid out for flags.

grep -r: Recursively search for files in the current directory and its subdirectories.

'\{.\*\}': This is the regular expression pattern we're searching for.

It matches { followed by any number of characters (denoted by .\*) followed by }.

We use backslashes (\) to escape the curly braces because they're special characters in regular expressions.

\*: Specifies the files to search in.

A screen shot of a computer code

Description automatically generated

Searched our target box IP address and came across this guy-

{STUDENT\_CTF\_Web}

A close-up of a green sign

Description automatically generated

Going through the hidden files and came across this guy

Got this one too

A screenshot of a computer program

Description automatically generated

{H1d3\_1n\_pl41n\_s1gh7}A screen shot of a computer

Description automatically generated

Found this one hiding out 

Knocking them out



More depth

A computer screen shot of a program

Description automatically generated

Went online to input the html and see what we got.

A screenshot of a computer

Description automatically generated

Had to decode A computer screen shot of a computer code

Description automatically generated

Another one

A screenshot of a computer

Description automatically generated

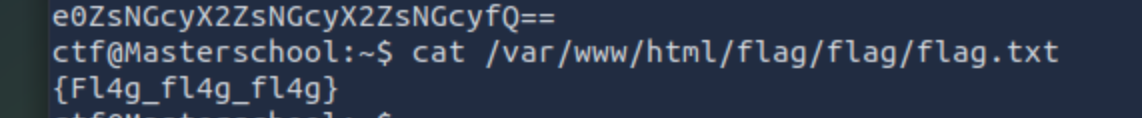
Another one. Had to crack this from base64

{Zl4g2\_fl4g2\_fl4g2}



Another one

{Fl4g\_fl4g\_fl4g}



Hash CrackingA computer screen shot of a computer program

Description automatically generated

Hash1- 53e06b5830ae3f4d7ebbf0baab22a2d1

Hash2- a6938e05ec33e356ff4b9aa961fe1e51138b4758

Hash3 - a15c292682ac51a76b7f25ec341707fc8967025d007a52c0fa8e565dfe2f7a5bca162e6b2fe8cd8f75c62192604f66df73d1a4028299f03c07fbc2dc6650b029

Hash4- d1d0f39e3be116c81453d7af22c3623ec555d007cdf77a9813e9647dfcc2cfaa

Hash5 - b9c86725a1c15a6af0e7b595b25b8d3a

Time to get crackin

sudo scp -r ctf@10.10.63.207:./hash\_to\_crack ./Desktop

had to bring the files back to where john the ripper and hashcat worked. Didn’t have permissions to install them on the CTF machine.

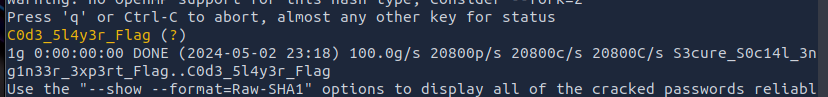
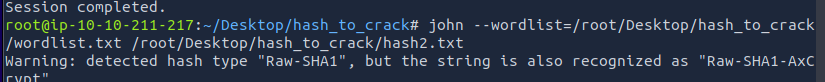
This was used to find what type of hash they are. Necessary when using cracking tools like hashcat or john the ripper.

A screenshot of a computer

Description automatically generated

Hash1 (cracked)- C0d3\_0b5cur3r\_Flag A computer screen with white text

Description automatically generated

Hash2 (cracked)- C0d3\_5l4y3r\_Flag

Hash3 (cracked)- H4ck3r\_FlagA screen shot of a computer code

Description automatically generated

Hash4 (cracked)- L0ck\_FlagA screenshot of a computer program

Description automatically generated

Hash5 (cracked) - S3cur1ty\_Flag

A screen shot of a computer

Description automatically generated

NMAP scan

A computer screen with white text and green text

Description automatically generated

FTP (Port 21/tcp):

Service: vsftpd 3.0.3

Vulnerability: Older versions of vsftpd may have known security vulnerabilities that could be exploited by attackers to gain unauthorized access or execute arbitrary code.

Recommendation: Update vsftpd to the latest version, apply patches, and enforce strong authentication mechanisms to prevent unauthorized access. Regularly monitor FTP logs for suspicious activities and implement intrusion detection systems (IDS) to detect and respond to potential attacks.

SSH (Port 22/tcp):

Service: OpenSSH 8.2p1 Ubuntu 4ubuntu0.5

Vulnerability: OpenSSH vulnerabilities could allow attackers to bypass authentication, conduct brute-force attacks, or exploit vulnerabilities in cryptographic protocols.

Recommendation: Keep OpenSSH up to date with the latest security patches, disable SSH protocol 1, enforce strong password policies, implement multi-factor authentication (MFA), and use tools like Fail2ban to mitigate brute-force attacks.

SMTP (Port 25/tcp):

Service: Postfix smtpd

Vulnerability: Misconfigured SMTP servers may be vulnerable to email spoofing, spamming, or unauthorized relay, which could lead to abuse of email services or phishing attacks.

Recommendation: Review Postfix configuration to ensure that it's not an open relay, implement SPF (Sender Policy Framework), DKIM (DomainKeys Identified Mail), and DMARC (Domain-based Message Authentication, Reporting, and Conformance) to prevent email spoofing and abuse. Monitor SMTP logs for suspicious activities and regularly audit email server configurations.

DNS (Port 53/tcp):

Service: ISC BIND 9.16.1-Ubuntu

Vulnerability: BIND vulnerabilities could lead to DNS cache poisoning, denial-of-service (DoS) attacks, or remote code execution.

Recommendation: Keep BIND updated, apply security patches promptly, configure BIND securely by limiting recursive queries, implementing access controls, and enabling DNSSEC (Domain Name System Security Extensions) to prevent DNS spoofing attacks. Regularly monitor DNS logs for unusual activities and consider using DNS firewall solutions for added protection.

HTTP (Port 80/tcp):

Service: Apache httpd 2.4.41 (Ubuntu)

Vulnerability: Outdated versions of Apache HTTP Server may have known vulnerabilities such as remote code execution, directory traversal, or HTTP header injection.

Recommendation: Keep Apache updated, apply security patches, disable unnecessary modules, configure secure SSL/TLS settings, implement web application firewalls (WAFs), and conduct regular security audits and vulnerability scans of web applications hosted on Apache servers.

POP3 (Port 110/tcp) and IMAP (Port 143/tcp):

Services: Dovecot pop3d and Dovecot imapd

Vulnerability: Vulnerabilities in Dovecot may allow attackers to access users' email accounts, intercept sensitive information, or perform denial-of-service attacks.

Recommendation: Keep Dovecot updated, apply security patches, enforce strong password policies, enable encryption for POP3S and IMAPS, implement intrusion detection and prevention systems (IDPS), and regularly audit email server configurations and logs for suspicious activities.

IMAPS (Port 993/tcp) and POP3S (Port 995/tcp):

Services: Encrypted versions of IMAP and POP3

Recommendation: Ensure that SSL/TLS certificates are up to date and properly configured, disable weak encryption algorithms, and implement secure cipher suites to protect sensitive email communications from interception and eavesdropping.