Committed

> Oh no, not again! One of our developers accidentally committed some sensitive code to our GitHub repository. Well, at least, that is what they told us... the problem is, we don't remember what or where! Can you track down what we accidentally committed?
>
> Access this challenge by deploying the machine attached to this task by pressing the green "Start Machine" button. You will need to use the in-browser view to complete this room. Don't see anything? Press the "Show Split Screen" button at the top of the page.
>
> The files you need are located in **/home/ubuntu/commited** on the VM attached to this task.

Enumeration

Transfer the file to the local machine for deeper analysis

```
┌──(kali㉿kali)-[~/TryHackMe/Committed]
└─$ wget http://10.10.155.51:8000/commited.zip
--2023-07-19 02:32:56--  http://10.10.155.51:8000/commited.zip
Connecting to 10.10.155.51:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 34993 (34K) [application/zip]
Saving to: ‘commited.zip’

commited.zip                 100%[============================================>]  34.17K   184KB/s    in 0.2s

2023-07-19 02:32:56 (184 KB/s) - ‘commited.zip’ saved [34993/34993]

┌──(kali㉿kali)-[~/TryHackMe/Committed]
└─$ ls -l
total 36
-rw-r--r-- 1 kali kali 34993 Mar 11  2022 commited.zip
```

Unzip the file

```
┌──(kali㉿kali)-[~/TryHackMe/Committed]
└─$ ls -l
total 40
drwxrwxr-x 3 kali kali  4096 Feb 13  2022 commited
-rw-r--r-- 1 kali kali 34993 Mar 11  2022 commited.zip

┌──(kali㉿kali)-[~/TryHackMe/Committed]
└─$ cd commited

┌──(kali㉿kali)-[~/TryHackMe/Committed/commited]
```

```
└─$ ls
main.py  Readme.md
```

Readme.md

```
# Commited
---

## About the Project

Commited is our project created to manage our databases, Commited will bring help our database management team by
 simplfying database management by using our python scripts.

## Project Status

Completed.

## Team

Our development team consists of finest developers and we work simultaneously using our cool version control metho
dology. We are the BEST.
```

main.py

```
import mysql.connector

def create_db():
    mydb = mysql.connector.connect(
    host="localhost",
    user="", # Username Goes Here
    password="" # Password Goes Here
    )

    mycursor = mydb.cursor()

    mycursor.execute("CREATE DATABASE commited")


def create_tables():
    mydb = mysql.connector.connect(
    host="localhost",
    user="", #username Goes here
    password="", #password Goes here
    database="commited"
    )

    mycursor = mydb.cursor()

    mycursor.execute("CREATE TABLE customers (name VARCHAR(255), address VARCHAR(255))")


def populate_tables():
    mydb = mysql.connector.connect(
    host="localhost",
    user="",
    password="",
    database="commited"
    )

    mycursor = mydb.cursor()

    sql = "INSERT INTO customers (name, address) VALUES (%s, %s)"
    val = ("John", "Highway 21")
    mycursor.execute(sql, val)

    mydb.commit()

    print(mycursor.rowcount, "record inserted.")


create_db()
create_tables()
populate_tables()
```

Exploit

As the file **main.py**, the **user** and **password** are required at each step (function) such as `create_db()`, `create_tables()`, `populate_tables()`. Therefore, while interacting with the server/git repository, the developers might get through these processes and they might entered the above sensitive information.

A tool extractor.sh could help to recover incomplete git repositories

```
┌──(kali㉿kali)-[~/TryHackMe/Committed]
└─$ wget https://raw.githubusercontent.com/internetwache/GitTools/master/Extractor/extractor.sh
--2023-07-19 03:00:09--  https://raw.githubusercontent.com/internetwache/GitTools/master/Extractor/extractor.sh
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.110.133, 185.199.109.133, 185.199.108.1
33, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.110.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2594 (2.5K) [text/plain]
Saving to: ‘extractor.sh’

extractor.sh                 100%[==============================================>]   2.53K  --.-KB/s    in 0s

2023-07-19 03:00:10 (55.7 MB/s) - ‘extractor.sh’ saved [2594/2594]
```

Use the following command format to execute the tool:

```
bash extractor.sh <TARGET_DIRECTORY> <DESTINATION_DIRECTORY>
```

Note: The `<TARGET_DIRECTORY>` which contains a `.git` directory

```
┌──(kali㉿kali)-[~/TryHackMe/Committed]
└─$ ls -la commited
total 20
drwxrwxr-x 3 kali kali 4096 Jul 19 02:41 .
drwxr-xr-x 4 kali kali 4096 Jul 19 03:00 ..
drwxrwxr-x 8 kali kali 4096 Feb 13  2022 .git
-rw-rw-r-- 1 kali kali  982 Feb 13  2022 main.py
-rw-rw-r-- 1 kali kali  393 Feb 13  2022 Readme.md
```

```
┌──(kali㉿kali)-[~/TryHackMe/Committed]
└─$ bash extractor.sh commited commited_extract
```

The extract directory would be like this:

```
┌──(kali㉿kali)-[~/TryHackMe/Committed/commited_extract]
└─$ tree
.
├── 0-4e16af9349ed8eaa4a29decd82a7f1f9886a32db
│   ├── commit-meta.txt
│   ├── main.py
│   ├── Note
│   └── Readme.md
├── 1-b0eda7db60a1cb0aea86f053816a1bfb7e2d6c67
│   ├── commit-meta.txt
│   ├── main.py
│   └── Readme.md
├── 2-441daaaa600aef8021f273c8c66404d5283ed83e
│   ├── commit-meta.txt
│   ├── main.py
│   └── Readme.md
├── 3-3a8cc16f919b8ac43651d68dceacbb28ebb9b625
│   ├── commit-meta.txt
│   ├── main.py
│   ├── Note
│   └── Readme.md
├── 4-28c36211be8187d4be04530e340206b856198a84
│   ├── commit-meta.txt
│   ├── main.py
│   └── Readme.md
├── 5-c56c470a2a9dfb5cfbd54cd614a9fdb1644412b5
│   ├── commit-meta.txt
│   ├── main.py
│   ├── Note
│   └── Readme.md
├── 6-9ecdc566de145f5c13da74673fa3432773692502
│   ├── commit-meta.txt
│   ├── main.py
│   └── Readme.md
├── 7-6e1ea88319ae84175bfe953b7791ec695e1ca004
│   ├── commit-meta.txt
│   ├── main.py
│   ├── Note
│   └── Readme.md
└── 8-26bcf1aa99094bf2fb4c9685b528a55838698fbe
    ├── commit-meta.txt
    ├── main.py
    └── Readme.md

10 directories, 31 files
```

For loop through all the files inside each directory to find the flag → Use `grep` with flag `-R` (--dereference-recursive)

```
┌──(kali㉿kali)-[~/TryHackMe/Committed/commited_extract]
└─$ grep -R "flag"
3-3a8cc16f919b8ac43651d68dceacbb28ebb9b625/main.py:    password="flag{HIDDEN}" # Password Goes Here
3-3a8cc16f919b8ac43651d68dceacbb28ebb9b625/main.py:    password="flag{HIDDEN}", #password Goes here
3-3a8cc16f919b8ac43651d68dceacbb28ebb9b625/main.py:    password="flag{HIDDEN}",
```