Hash auth

Description: Ok ok ok. They have learned there lesson. Instead of using confusing authentication functions there have switched to crypto functions. At least they think they have switched to crypto functions. Anyway - please help us again and hack this piece of code!
Level: Level 2
Type: Coding
Link: *https://ctf.securityvalley.org/dashboard*

So what we have is a `.py` file:

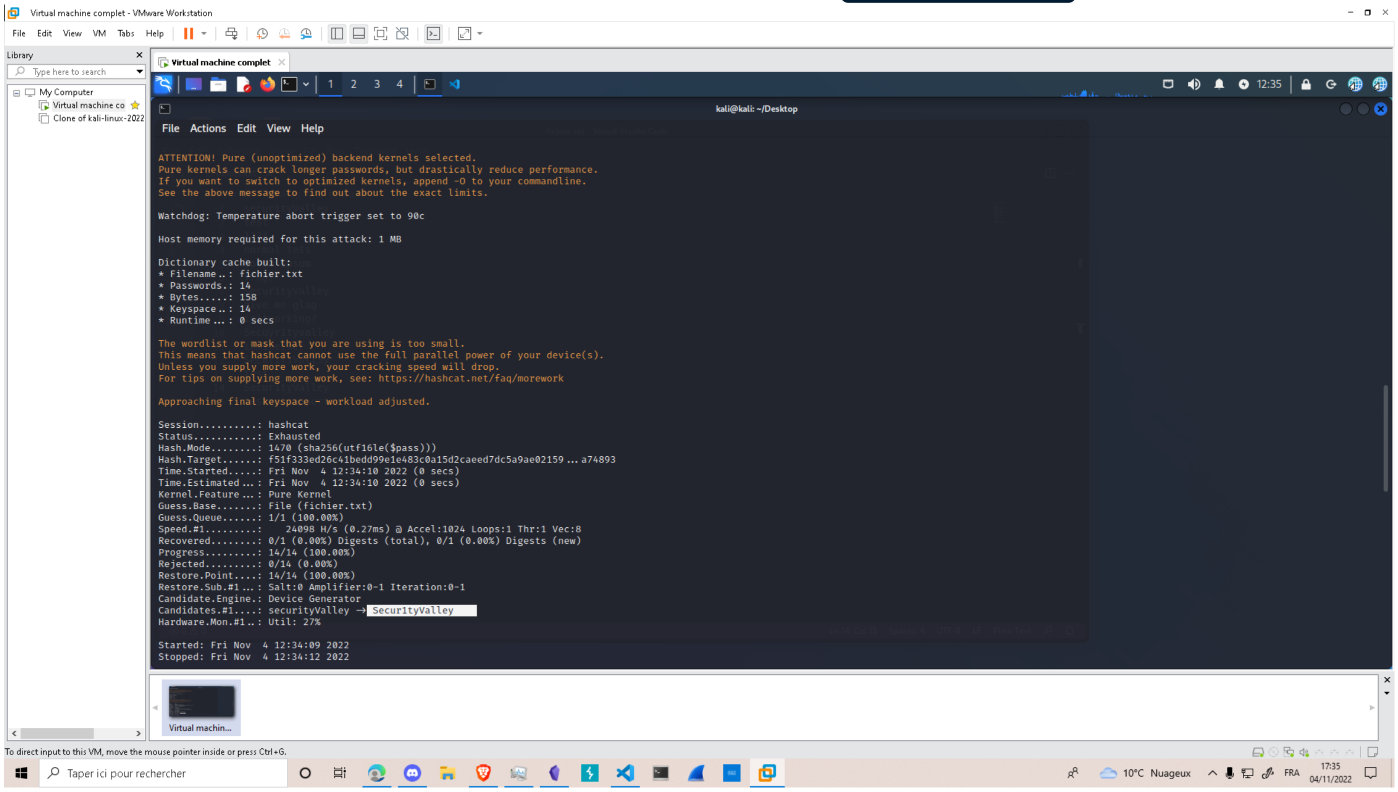
```
from hashlib import sha256

import sys



def validate_password(password):

    # be creative. it has something to do with SecurityValley ;-)

    if sha256(password.encode("utf-8")).hexdigest() ==
"f51f333ed26c41bedd99e1e483c0a15d2caeed7dc5a9ae02159f196799a74893":

        return True



    return False



def print_banner(payload):
    print("that was great !!!")

    print("run the following command to get the flag.")

    print("curl -X POST http://ctf.securityvalley.org:7777/api/v1/validate -H 'Content-Type: application/json' -d
'{\"pass\": \""+payload+"\"}'")



if __name__ == "__main__":

    print("let's do more python ;-)")



    password = input("please enter password: ")

    if validate_password(password):

        print_banner(password)

        sys.exit()

    print("wrong!")
```

We can see that their is a hash: `f51f333ed26c41bedd99e1e483c0a15d2caeed7dc5a9ae02159f196799a74893`, but it gives us nothing with **hash cracker**.
They give us a hint: "be creative. it has something to do with SecurityValley ;-)",so we can try to make a `"wordlist"` file with some varient of "SecurityValley" like this:

```
securityValley
SecurityValley
securityvalley
Securit1Valley
secur1tyValley
Securt1valley
```

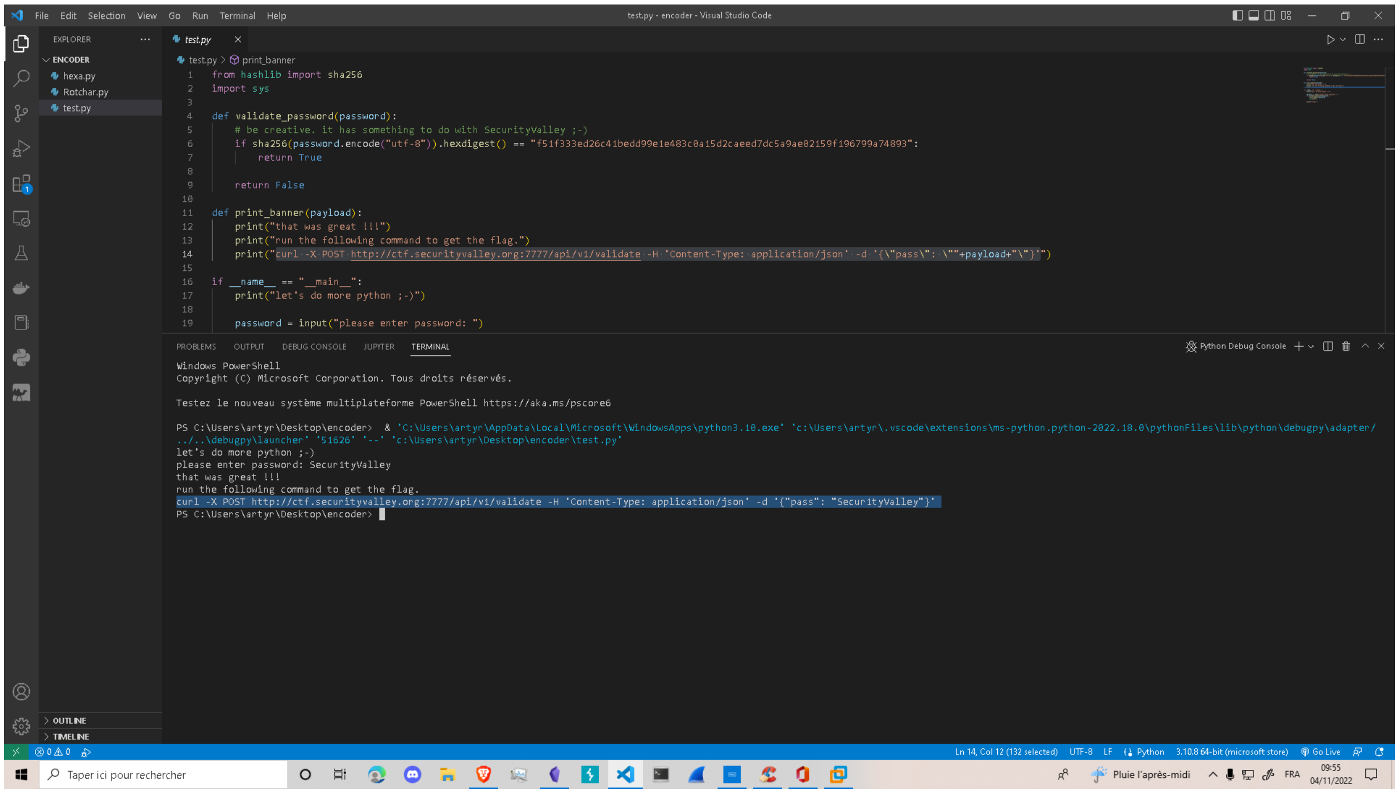
and use **Hashcat** to crack the hash and get the password: by using this command:
`hashcat -a 0 -m 1470 f51f333ed26c41bedd99e1e483c0a15d2caeed7dc5a9ae02159f196799a74893`

`wordlist.txt -o --show`
to crack the ashes with some word of our `wordlist.txt` file,and we can see that the first one to come is `Secur1tyValley`:



```
ATTENTION! Pure (unoptimized) backend kernels selected.
Pure kernels can crack longer passwords, but drastically reduce performance.
If you want to switch to optimized kernels, append -O to your commandline.
See the above message to find out about the exact limits.

Watchdog: Temperature abort trigger set to 90c

Host memory required for this attack: 1 MB

Dictionary cache built:
* Filename..: fichier.txt
* Passwords.: 14
* Bytes.....: 158
* Keyspace..: 14
* Runtime...: 0 secs

The wordlist or mask that you are using is too small.
This means that hashcat cannot use the full parallel power of your device(s).
Unless you supply more work, your cracking speed will drop.
For tips on supplying more work, see: https://hashcat.net/faq/morework

Approaching final keyspace - workload adjusted.

Session..........: hashcat
Status...........: Exhausted
Hash.Mode........: 1470 (sha256(utf16le($pass)))
Hash.Target......: f51f333ed26c41bedd99e1e483c0a15d2caeed7dc5a9ae02159...a74893
Time.Started.....: Fri Nov  4 12:34:10 2022 (0 secs)
Time.Estimated...: Fri Nov  4 12:34:10 2022 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (fichier.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:    24098 H/s (0.27ms) @ Accel:1024 Loops:1 Thr:1 Vec:8
Recovered........: 0/1 (0.00%) Digests (total), 0/1 (0.00%) Digests (new)
Progress.........: 14/14 (100.00%)
Rejected.........: 0/14 (0.00%)
Restore.Point....: 14/14 (100.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1....: securityValley -> Secur1tyValley
Hardware.Mon.#1..: Util: 27%

Started: Fri Nov  4 12:34:09 2022
Stopped: Fri Nov  4 12:34:12 2022
```

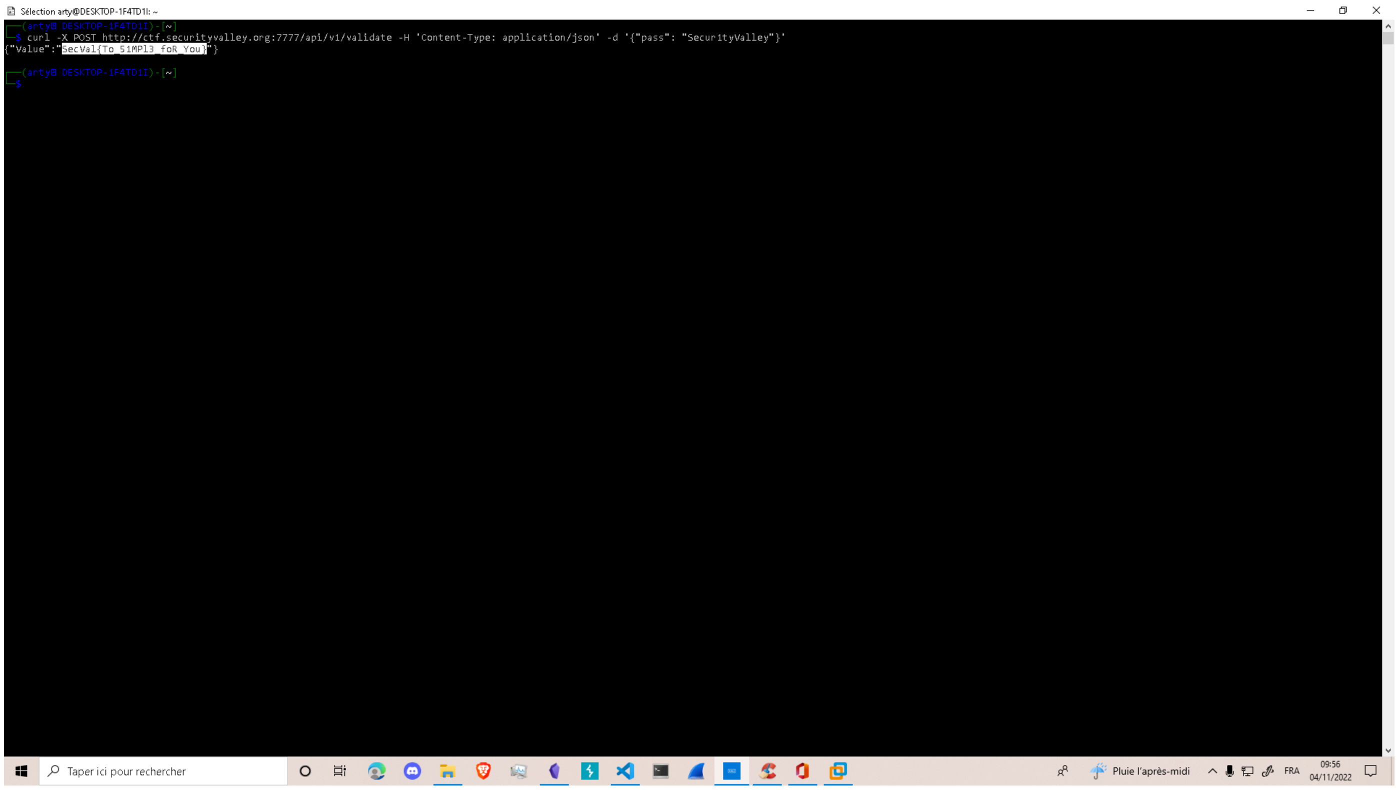
And then when we enter `Secur1tyValley` in the python script,it give us the command to run to get the flag:



```python
from hashlib import sha256
import sys

def validate_password(password):
    # be creative. it has something to do with SecurityValley ;-)
    if sha256(password.encode("utf-8")).hexdigest() == "f51f333ed26c41bedd99e1e483c0a15d2caeed7dc5a9ae02159f196799a74893":
        return True

    return False

def print_banner(payload):
    print("that was great !!!")
    print("run the following command to get the flag.")
    print("curl -X POST http://ctf.securityvalley.org:7777/api/v1/validate -H 'Content-Type: application/json' -d '{\"pass\": \""+payload+"\"}'")

if __name__ == "__main__":
    print("let's do more python ;-)")

    password = input("please enter password: ")
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Tous droits réservés.

Testez le nouveau système multiplateforme PowerShell https://aka.ms/pscore6

PS C:\Users\artyr\Desktop\encoder>  & 'C:\Users\artyr\AppData\Local\Microsoft\WindowsApps\python3.10.exe' 'c:\Users\artyr\.vscode\extensions\ms-python.python-2022.18.0\pythonFiles\lib\python\debugpy\adapter/../..\debugpy\launcher' '51626' '--' 'c:\Users\artyr\Desktop\encoder\test.py'
let's do more python ;-)
please enter password: Secur1tyValley
that was great !!!
run the following command to get the flag.
curl -X POST http://ctf.securityvalley.org:7777/api/v1/validate -H 'Content-Type: application/json' -d '{"pass": "Secur1tyValley"}'
PS C:\Users\artyr\Desktop\encoder>
```

And when we run: `curl -X POST *http://ctf.securityvalley.org:7777/api/v1/validate* -H 'Content-Type: application/json' -d '{"pass": "Secur1tyValley"}'
It gives us the flag !



Sélection arty@DESKTOP-1F4TD1I: ~

```
┌──(arty㉿ DESKTOP-1F4TD1I)-[~]
└─$ curl -X POST http://ctf.securityvalley.org:7777/api/v1/validate -H 'Content-Type: application/json' -d '{"pass": "Secur1tyValley"}'
{"Value":"SecVal{To_51MPl3_foR_You}"}

┌──(arty㉿ DESKTOP-1F4TD1I)-[~]
└─$
```

The flag is: **SecVal{To_51MPl3_foR_You}**