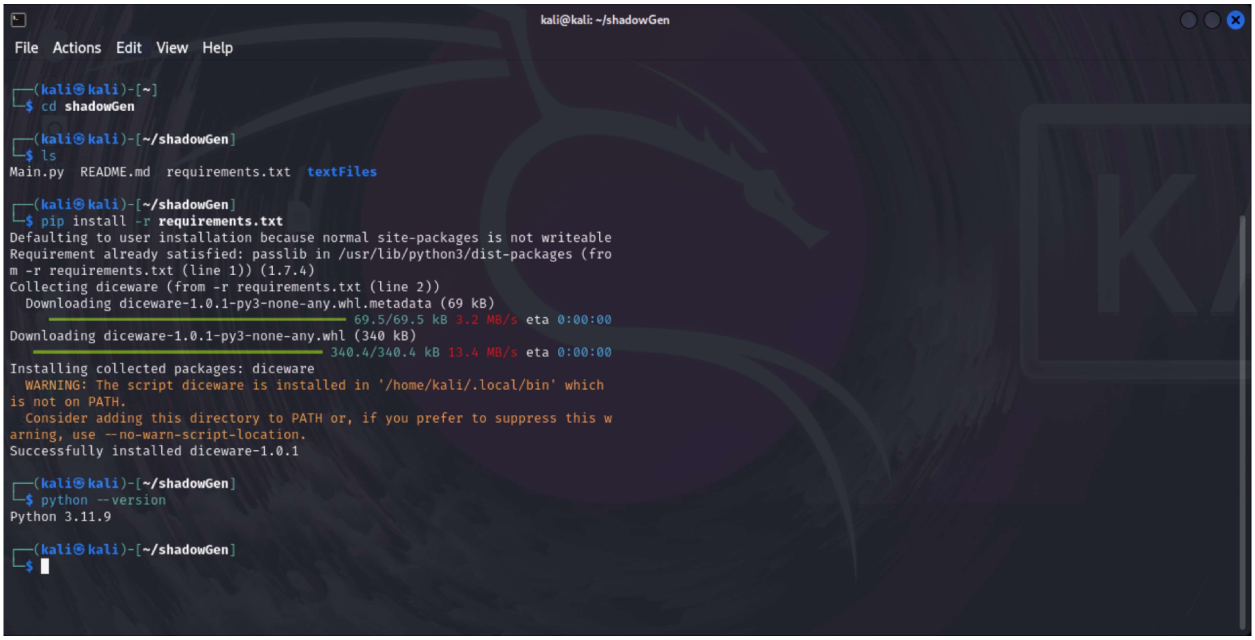
# ShadowGen Lab

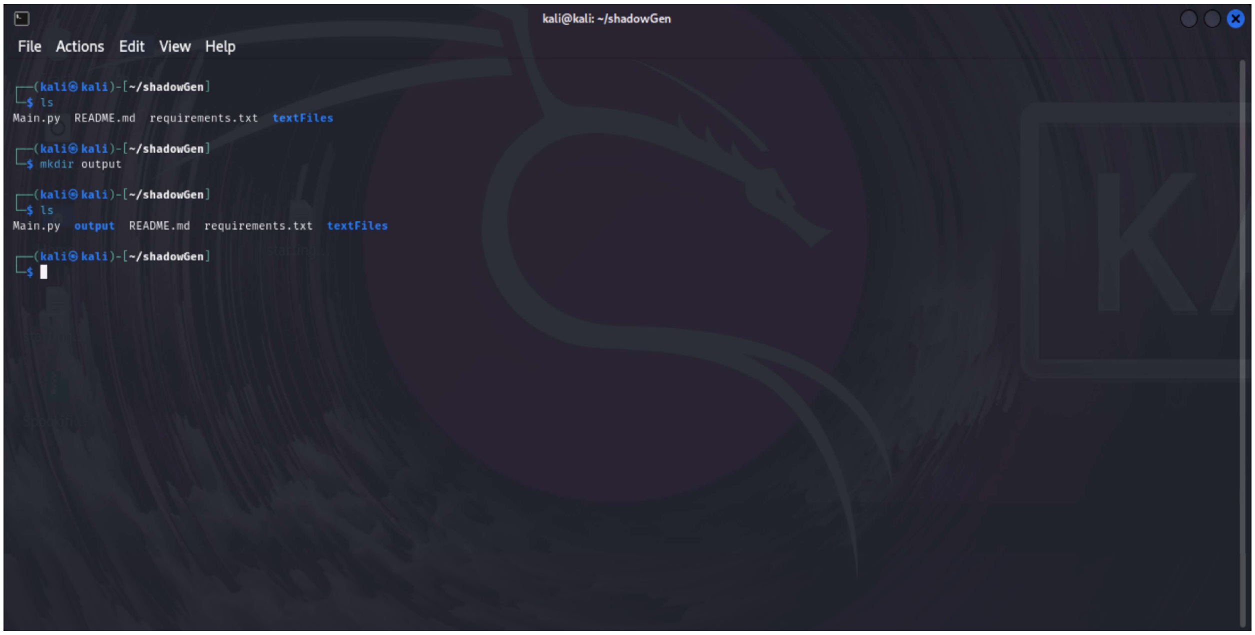
## Installation

For this lab, I used a Kali Linux VirtualBox .iso image: kali-linux-2024.3-virtualbox-amd64.

1. **Command**: git clone https://github.com/ethan-snyder/shadowGen.git
2. **Command**: cd ShadowGen
3. Next, we will install the requirements, using requirements.txt
    a. **Command**: pip install -r requirements.txt
4. After that, you need to ensure we have a Python version ≥Python 3.7. If your python version is less than 3.7, you may need to install a newer version for this lab.
    a. **Command**: python –version
5. Lastly, you will need to create an output directory.
    a. **Command**: mkdir output



```
┌──(kali㉿kali)-[~]
└─$ cd shadowGen

┌──(kali㉿kali)-[~/shadowGen]
└─$ ls
Main.py  README.md  requirements.txt  textFiles

┌──(kali㉿kali)-[~/shadowGen]
└─$ pip install -r requirements.txt
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: passlib in /usr/lib/python3/dist-packages (fro
m -r requirements.txt (line 1)) (1.7.4)
Collecting diceware (from -r requirements.txt (line 2))
  Downloading diceware-1.0.1-py3-none-any.whl.metadata (69 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 69.5/69.5 kB 3.2 MB/s eta 0:00:00
Downloading diceware-1.0.1-py3-none-any.whl (340 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 340.4/340.4 kB 13.4 MB/s eta 0:00:00
Installing collected packages: diceware
  WARNING: The script diceware is installed in '/home/kali/.local/bin' which
is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this w
arning, use --no-warn-script-location.
Successfully installed diceware-1.0.1

┌──(kali㉿kali)-[~/shadowGen]
└─$ python --version
Python 3.11.9

┌──(kali㉿kali)-[~/shadowGen]
└─$ █
```

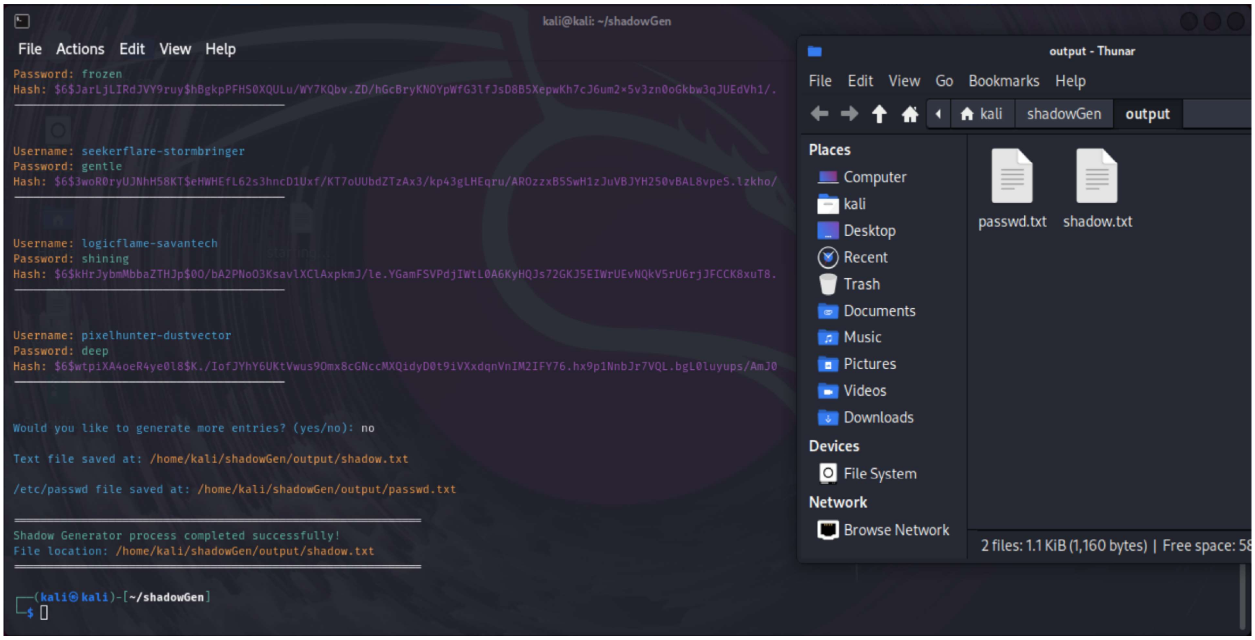


```
┌──(kali㉿kali)-[~/shadowGen]
└─$ ls
Main.py  README.md  requirements.txt  textFiles

┌──(kali㉿kali)-[~/shadowGen]
└─$ mkdir output

┌──(kali㉿kali)-[~/shadowGen]
└─$ ls
Main.py  output  README.md  requirements.txt  textFiles

┌──(kali㉿kali)-[~/shadowGen]
└─$ █
```

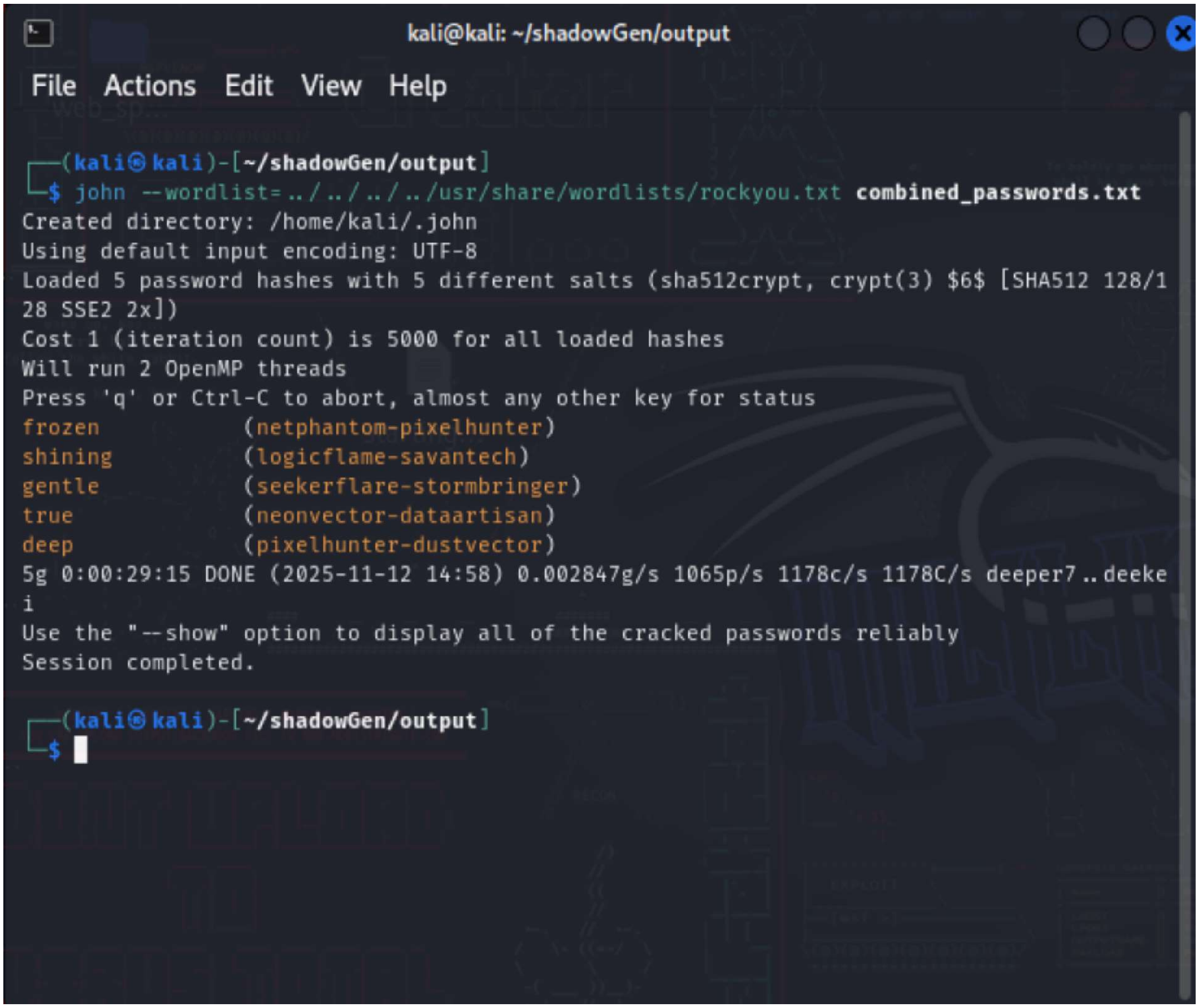
## Creating Credentials & Ripping

Now that shadowGen has successfully been cloned, it is time to generate some example shadow & passwd files.

1. Command: python Main.py
2. Now that shadowGen is running, you can simply follow the prompts until you have generated enough passwords for your purposes. After you are done, Main.py will close, and the output folder will open. For this instance, I generated 5 passwords each only being 1 word. This will obviously take less time to rip. However, I don't have a super computer so we will have to sacrifice some realism for practicality.
3. Here is where the lab starts to get *really interesting*. Now that everything has been generated, we first need to format the output with john the ripper.
    a. Command: unshadow passwd.txt shadow.txt combined_passwords.txt
4. Finally, we are ready to rip. You can use any hash ripper or word list of your choice. I opted to use John the Ripper as my ripper, and rockyou.txt as my wordlist (you can't beat the classics). If you are using Kali Linux, I recommend that you first unzip rockyou.txt so that you can use this word list first, as this is one of the most prolific wordlists for offensive security specialists.
    a. Command: john –wordlist=[pathToWordlist] combined_passwords.txt
5. And now we wait.

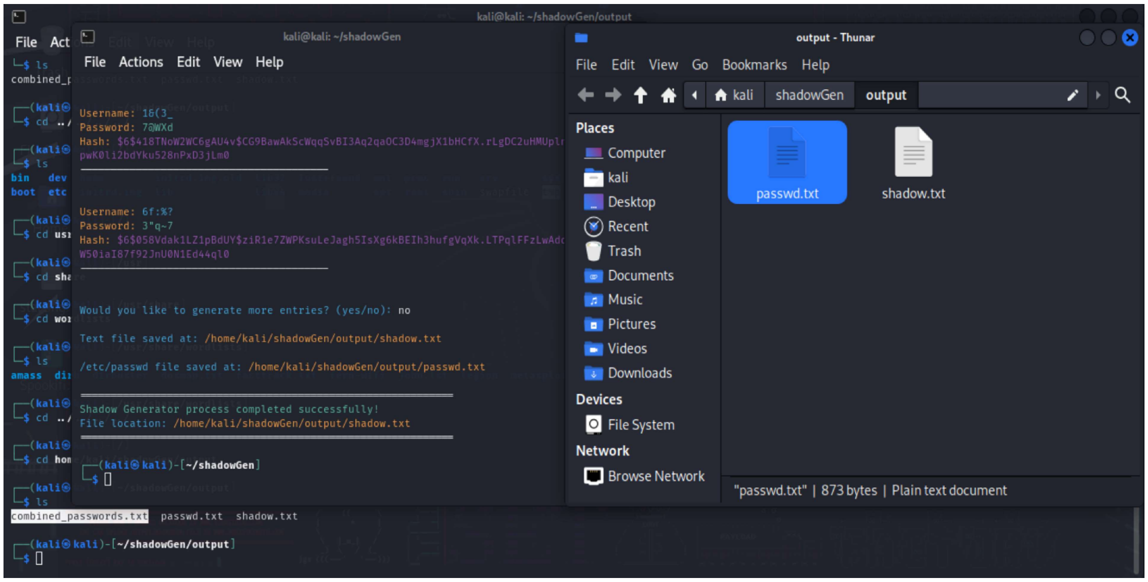


```
Password: frozen
Hash: $6$JarLjLIRdJVY9ruy$hBgkpPFHS0XQULu/WY7KQbv.ZD/hGcBryKNOYpWfG3lfJsD8B5XepwKh7cJ6um2×5v3zn0oGkbw3qJUEdVh1/.
────────────────────────────────────


Username: seekerflare-stormbringer
Password: gentle
Hash: $6$3woR0ryUJNhH58KT$eHWHEfL62s3hncD1Uxf/KT7oUUbdZTzAx3/kp43gLHEqru/AROzzxB5SwH1zJuVBJYH250vBAL8vpeS.lzkho/
────────────────────────────────────


Username: logicflame-savantech
Password: shining
Hash: $6$kHrJybmMbbaZTHJp$0O/bA2PNoO3KsavlXClAxpkmJ/le.YGamFSVPdjIWtL0A6KyHQJs72GKJ5EIWrUEvNQkV5rU6rjJFCCK8xuT8.
────────────────────────────────────


Username: pixelhunter-dustvector
Password: deep
Hash: $6$wtpiXA4oeR4ye0l8$K./IofJYhY6UKtVwus9Omx8cGNccMXQidyD0t9iVXxdqnVnIM2IFY76.hx9p1NnbJr7VQL.bgL0luyups/AmJ0
────────────────────────────────────


Would you like to generate more entries? (yes/no): no

Text file saved at: /home/kali/shadowGen/output/shadow.txt

/etc/passwd file saved at: /home/kali/shadowGen/output/passwd.txt

════════════════════════════════════════════════════════
Shadow Generator process completed successfully!
File location: /home/kali/shadowGen/output/shadow.txt
════════════════════════════════════════════════════════

┌──(kali㉿kali)-[~/shadowGen]
└─$ ▯
```



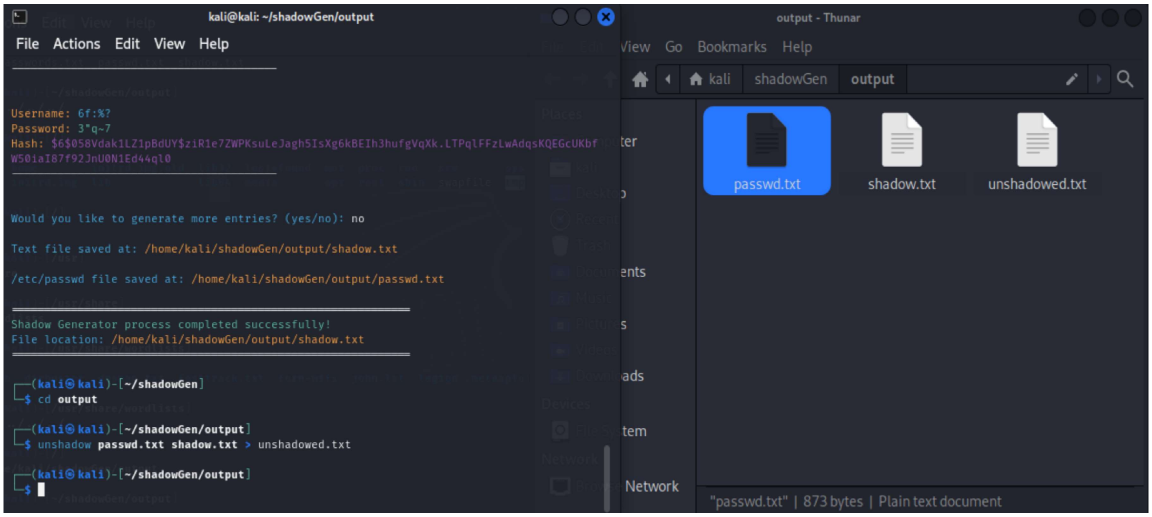
```
┌──(kali㉿kali)-[~/shadowGen/output]
└─$ john --wordlist=../../../../usr/share/wordlists/rockyou.txt combined_passwords.txt
Created directory: /home/kali/.john
Using default input encoding: UTF-8
Loaded 5 password hashes with 5 different salts (sha512crypt, crypt(3) $6$ [SHA512 128/1
28 SSE2 2x])
Cost 1 (iteration count) is 5000 for all loaded hashes
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
frozen           (netphantom-pixelhunter)
shining          (logicflame-savantech)
gentle           (seekerflare-stormbringer)
true             (neonvector-dataartisan)
deep             (pixelhunter-dustvector)
5g 0:00:29:15 DONE (2025-11-12 14:58) 0.002847g/s 1065p/s 1178c/s 1178C/s deeper7..deeke
i
Use the "--show" option to display all of the cracked passwords reliably
Session completed.

┌──(kali㉿kali)-[~/shadowGen/output]
└─$ █
```

Below, I have included an example with a couple of passwords that are either 4 or 5 characters long. Even after almost 48 minutes of cracking, at 1251 C/s (cracks per second), not a single hash has been cracked. This is the most important thing you should reremember from this lab. Passwords that contain words alone are insecure. Assuming a more realistic scenario with an 8-character password, that is $94^8$ or 6,095,689,385,410,816 possible passwords! Assuming 100,000 C/s, it would still take 60,956,893,854 seconds, or ~3,864 years. Personally, I don't have that long, but to each their own!



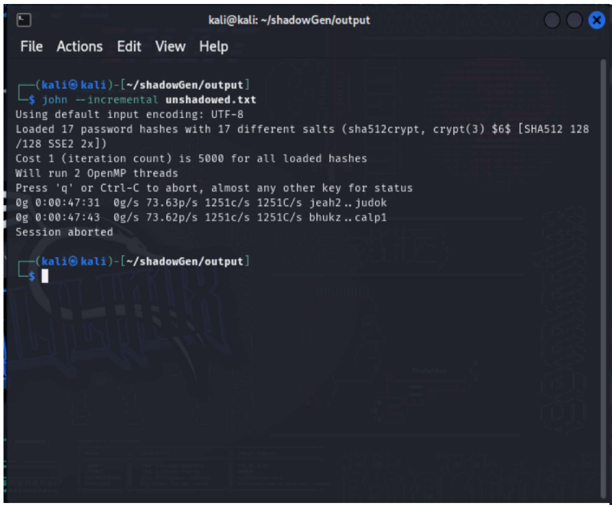
```
└─$ ls
combined_p

┌──(kali㉿
└─$ cd ../

┌──(kali㉿
└─$ ls
bin   dev
boot  etc

┌──(kali㉿
└─$ cd usr

┌──(kali㉿
└─$ cd sha

┌──(kali㉿
└─$ cd wor

┌──(kali㉿
└─$ ls
amass  dir

┌──(kali㉿
└─$ cd ../

┌──(kali㉿
└─$ cd hom

┌──(kali㉿
└─$ ls
combined_passwords.txt  passwd.txt  shadow.txt

┌──(kali㉿kali)-[~/shadowGen/output]
└─$ ▯
```

```
Username: 1б(3_
Password: 7@WXd
Hash: $6$418TNoW2WC6gAU4v$CG9BawAkScWqqSvBI3Aq2qaOC3D4mgjX1bHCfX.rLgDC2uHMUplr
pwK0li2bdYku528nPxD3jLm0
────────────────────────────────────────


Username: 6f:%?
Password: 3"q~7
Hash: $6$058Vdak1LZ1pBdUY$ziR1e7ZWPKsuLeJagh5IsXg6kBEIh3hufgVqXk.LTPqlFFzLwAdc
W50iaI87f92JnU0N1Ed44ql0
────────────────────────────────────────


Would you like to generate more entries? (yes/no): no

Text file saved at: /home/kali/shadowGen/output/shadow.txt

/etc/passwd file saved at: /home/kali/shadowGen/output/passwd.txt

════════════════════════════════════════════════════════════
Shadow Generator process completed successfully!
File location: /home/kali/shadowGen/output/shadow.txt
════════════════════════════════════════════════════════════

┌──(kali㉿kali)-[~/shadowGen]
└─$ ▯
```

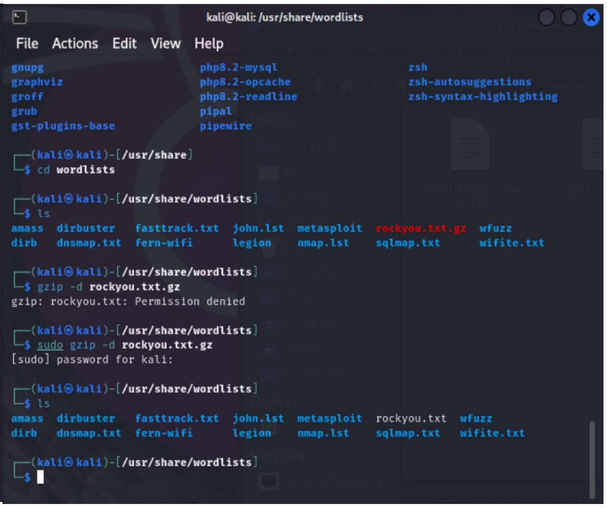


```
Username: 6f:%?
Password: 3"q~7
Hash: $6$058Vdak1LZ1pBdUY$ziR1e7ZWPKsuLeJagh5IsXg6kBEIh3hufgVqXk.LTPqlFFzLwAdqsKQEGcUKbf
W50iaI87f92JnU0N1Ed44ql0
────────────────────────────────────────


Would you like to generate more entries? (yes/no): no

Text file saved at: /home/kali/shadowGen/output/shadow.txt

/etc/passwd file saved at: /home/kali/shadowGen/output/passwd.txt

════════════════════════════════════════════════════════════
Shadow Generator process completed successfully!
File location: /home/kali/shadowGen/output/shadow.txt
════════════════════════════════════════════════════════════

┌──(kali㉿kali)-[~/shadowGen]
└─$ cd output

┌──(kali㉿kali)-[~/shadowGen/output]
└─$ unshadow passwd.txt shadow.txt > unshadowed.txt

┌──(kali㉿kali)-[~/shadowGen/output]
└─$ █
```

kali@kali: ~/shadowGen/output
File Actions Edit View Help
output - Thunar
View Go Bookmarks Help
kali shadowGen output
passwd.txt
shadow.txt
unshadowed.txt
"passwd.txt" | 873 bytes | Plain text document



```
┌──(kali㉿kali)-[~/shadowGen/output]
└─$ john --incremental unshadowed.txt
Using default input encoding: UTF-8
Loaded 17 password hashes with 17 different salts (sha512crypt, crypt(3) $6$ [SHA512 128
/128 SSE2 2x])
Cost 1 (iteration count) is 5000 for all loaded hashes
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
0g 0:00:47:31  0g/s 73.63p/s 1251c/s 1251C/s jeah2..judok
0g 0:00:47:43  0g/s 73.62p/s 1251c/s 1251C/s bhukz..calp1
Session aborted

┌──(kali㉿kali)-[~/shadowGen/output]
└─$ █
```



```
gnupg                   php8.2-mysql              zsh
graphviz                php8.2-opcache            zsh-autosuggestions
groff                   php8.2-readline           zsh-syntax-highlighting
grub                    pipal
gst-plugins-base        pipewire

┌──(kali㉿kali)-[/usr/share]
└─$ cd wordlists

┌──(kali㉿kali)-[/usr/share/wordlists]
└─$ ls
amass  dirbuster  fasttrack.txt  john.lst  metasploit  rockyou.txt.gz  wfuzz
dirb   dnsmap.txt  fern-wifi     legion    nmap.lst    sqlmap.txt      wifite.txt

┌──(kali㉿kali)-[/usr/share/wordlists]
└─$ gzip -d rockyou.txt.gz
gzip: rockyou.txt: Permission denied

┌──(kali㉿kali)-[/usr/share/wordlists]
└─$ sudo gzip -d rockyou.txt.gz
[sudo] password for kali:

┌──(kali㉿kali)-[/usr/share/wordlists]
└─$ ls
amass  dirbuster  fasttrack.txt  john.lst  metasploit  rockyou.txt  wfuzz
dirb   dnsmap.txt  fern-wifi     legion    nmap.lst    sqlmap.txt   wifite.txt

┌──(kali㉿kali)-[/usr/share/wordlists]
└─$ █
```

## Takeaways

This lab helps to teach the basics of hash cracking using John the Ripper. Most importantly, however, it illustrates that passwords should obey the following rules if they wish to be practically secure:

1. Should contain no words
2. Use lower-case and upper-case alphanumeric characters and symbols
3. Should contain many characters, realistically, no less than 8

## Resources

https://github.com/ethan-snyder/shadowGen

https://www.openwall.com/john/doc/

https://pegasustechnologies.com/password-security-2025/