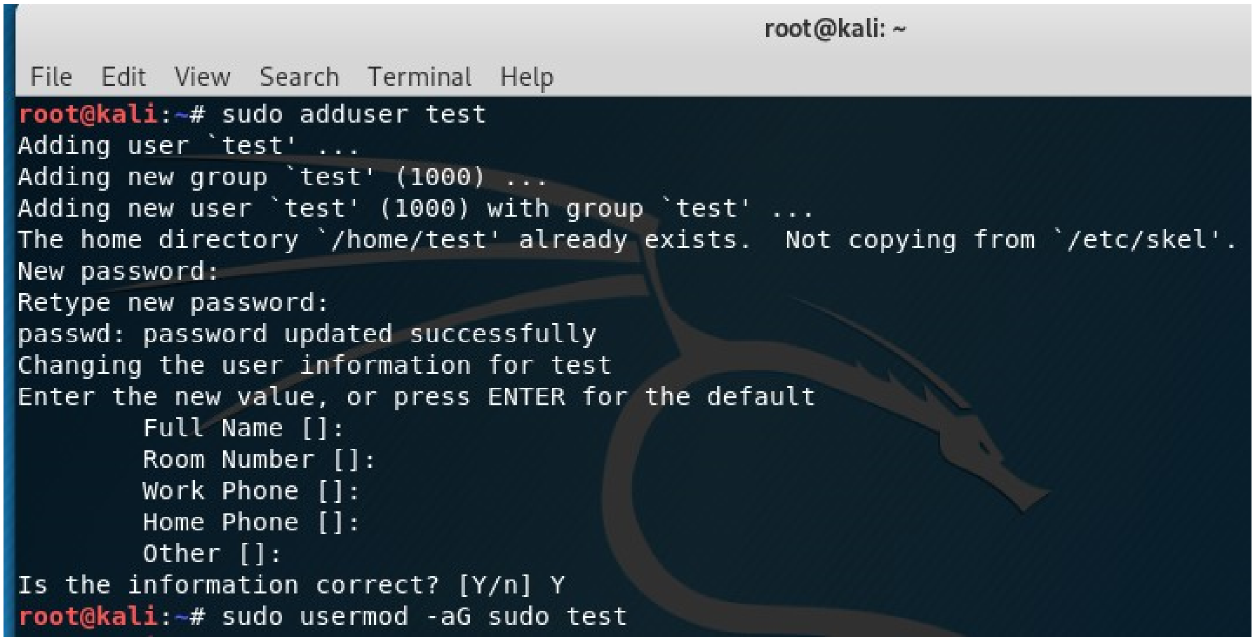
“Encrypting File with Generated Password” 7/24/23

This lab is powered by
Kali 2019 Virtual Machine

Task : Successfully encrypt a file with a password generated from our .py script we made

Step 1 : (a.)Make a user to use other then root
(b.)give sudo permissions

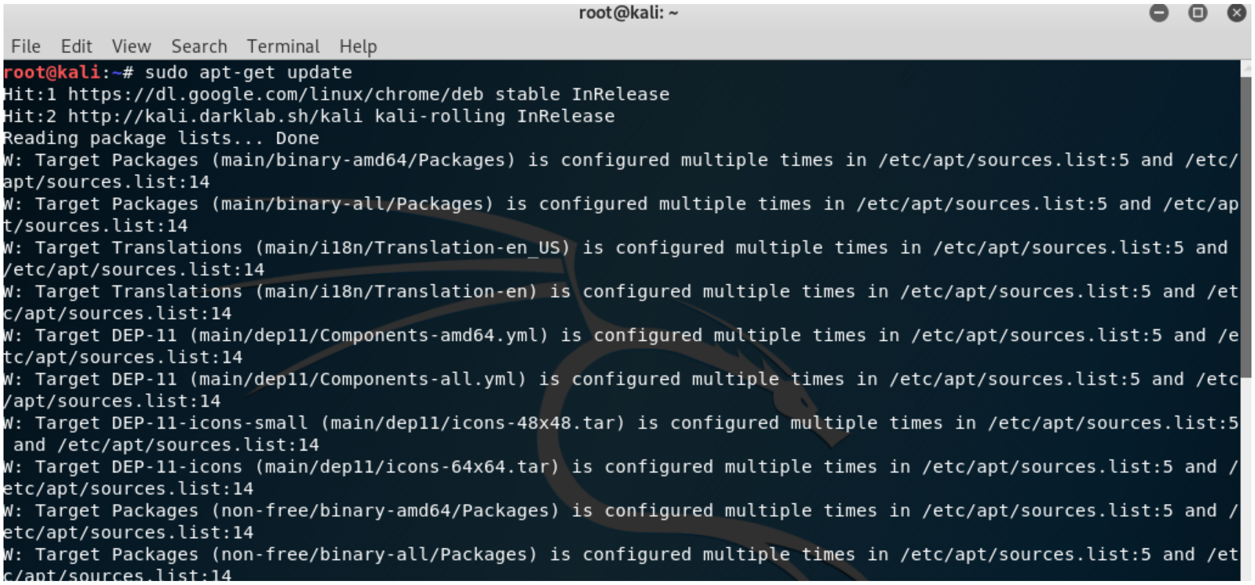


```
root@kali:~# sudo adduser test
Adding user `test' ...
Adding new group `test' (1000) ...
Adding new user `test' (1000) with group `test' ...
The home directory `/home/test' already exists.  Not copying from `/etc/skel'.
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for test
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] Y
root@kali:~# sudo usermod -aG sudo test
```

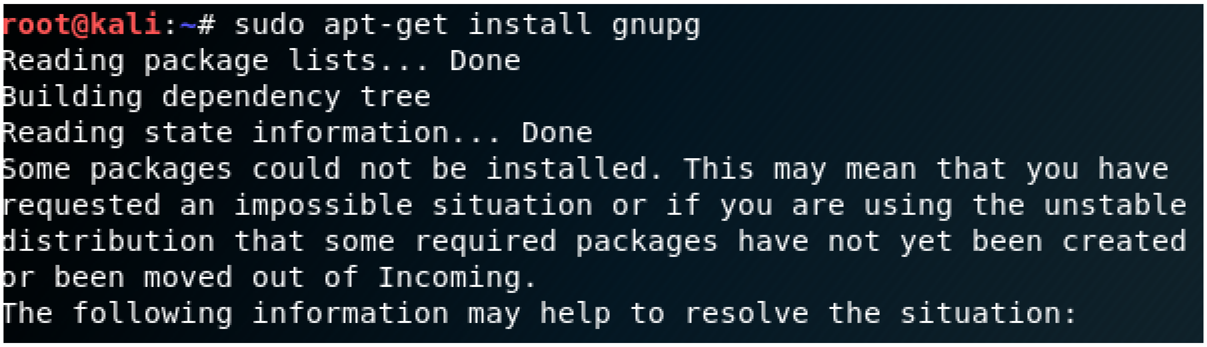
Step 2 : Install **gpg** (A utility that will allow us to encrypt and decrypt files securely

(a.) sudo apt-get update

(b.) sudo apt-get install gnupg

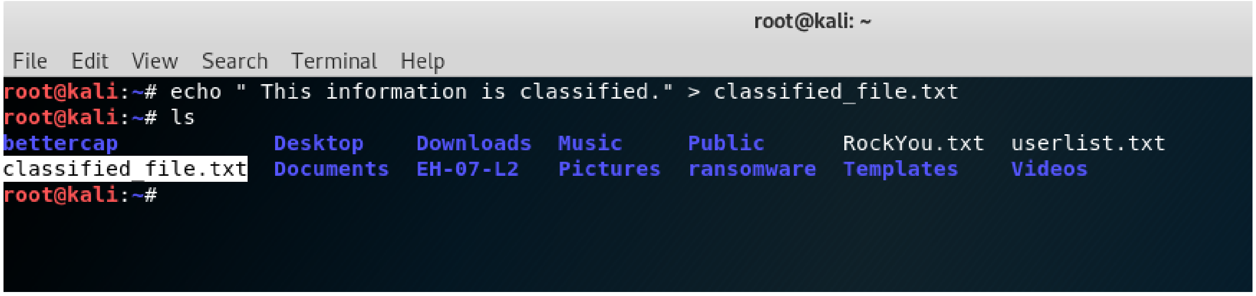


```
root@kali:~# sudo apt-get update
Hit:1 https://dl.google.com/linux/chrome/deb stable InRelease
Hit:2 http://kali.darklab.sh/kali kali-rolling InRelease
Reading package lists... Done
W: Target Packages (main/binary-amd64/Packages) is configured multiple times in /etc/apt/sources.list:5 and /etc/
apt/sources.list:14
W: Target Packages (main/binary-all/Packages) is configured multiple times in /etc/apt/sources.list:5 and /etc/ap
t/sources.list:14
W: Target Translations (main/i18n/Translation-en_US) is configured multiple times in /etc/apt/sources.list:5 and
/etc/apt/sources.list:14
W: Target Translations (main/i18n/Translation-en) is configured multiple times in /etc/apt/sources.list:5 and /et
c/apt/sources.list:14
W: Target DEP-11 (main/dep11/Components-amd64.yml) is configured multiple times in /etc/apt/sources.list:5 and /e
tc/apt/sources.list:14
W: Target DEP-11 (main/dep11/Components-all.yml) is configured multiple times in /etc/apt/sources.list:5 and /etc
/apt/sources.list:14
W: Target DEP-11-icons-small (main/dep11/icons-48x48.tar) is configured multiple times in /etc/apt/sources.list:5
 and /etc/apt/sources.list:14
W: Target DEP-11-icons (main/dep11/icons-64x64.tar) is configured multiple times in /etc/apt/sources.list:5 and /
etc/apt/sources.list:14
W: Target Packages (non-free/binary-amd64/Packages) is configured multiple times in /etc/apt/sources.list:5 and /
etc/apt/sources.list:14
W: Target Packages (non-free/binary-all/Packages) is configured multiple times in /etc/apt/sources.list:5 and /et
c/apt/sources.list:14
```



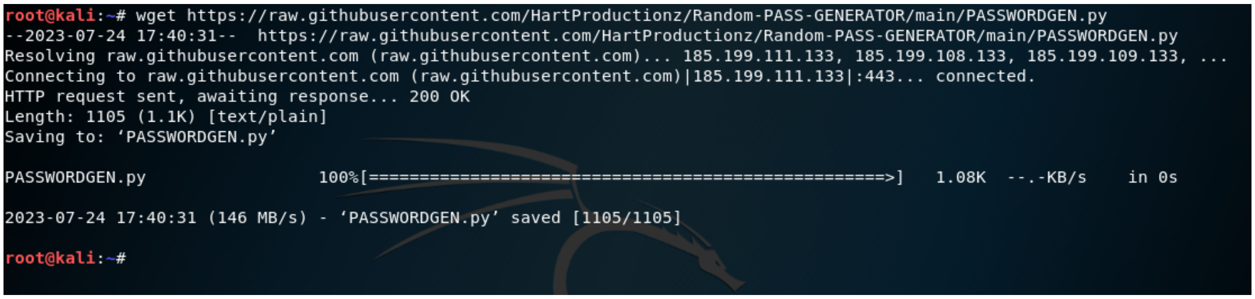
```
root@kali:~# sudo apt-get install gnupg
Reading package lists... Done
Building dependency tree
Reading state information... Done
Some packages could not be installed. This may mean that you have
requested an impossible situation or if you are using the unstable
distribution that some required packages have not yet been created
or been moved out of Incoming.
The following information may help to resolve the situation:
```

Step 3: Create the sensitive file with sensitive information in it



```
root@kali:~# echo " This information is classified." > classified_file.txt
root@kali:~# ls
bettercap            Desktop    Downloads  Music     Public      RockYou.txt  userlist.txt
classified_file.txt  Documents  EH-07-L2   Pictures  ransomware  Templates    Videos
root@kali:~#
```

Step 4 : Load the python Script onto the kali VM



```
root@kali:~# wget https://raw.githubusercontent.com/HartProductionz/Random-PASS-GENERATOR/main/PASSWORDGEN.py
--2023-07-24 17:40:31--  https://raw.githubusercontent.com/HartProductionz/Random-PASS-GENERATOR/main/PASSWORDGEN.py
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.111.133, 185.199.108.133, 185.199.109.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.111.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1105 (1.1K) [text/plain]
Saving to: ‘PASSWORDGEN.py’

PASSWORDGEN.py                 100%[===================================================>]   1.08K  --.-KB/s    in 0s

2023-07-24 17:40:31 (146 MB/s) - ‘PASSWORDGEN.py’ saved [1105/1105]

root@kali:~#
```

Step (5a) : Make your script an executable & run it

I Realized that I made an error, I made the script in windows and tried to transfer it over to Unix format which does not transfer well.

You have to download **dos2unix**

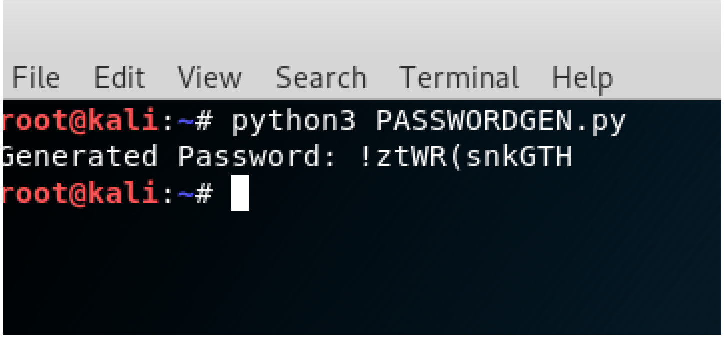
(a.) sudo-apt-get update

(b.)sudo apt-get install **dos2unix**

(c.) **dos2unix** PASSWORDGEN.py

Step (5b) : Get the 12 character unique password

(python3 PASSWORDGEN.py)



```
root@kali:~# python3 PASSWORDGEN.py
Generated Password: !ztWR(snkGTH
root@kali:~#
```

(!ztWR(snkGTH

Step 5: Encrypt the file using **gpg**

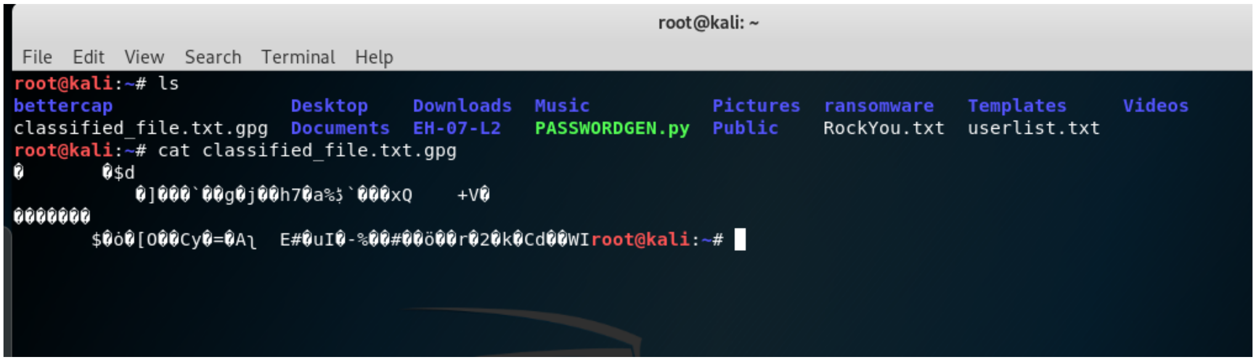
“gpg -c classified_file.txt

-then remove unencrypted copy with :

shred -u classified_file.txt (This will securely delete the file)

Step 6 : Results

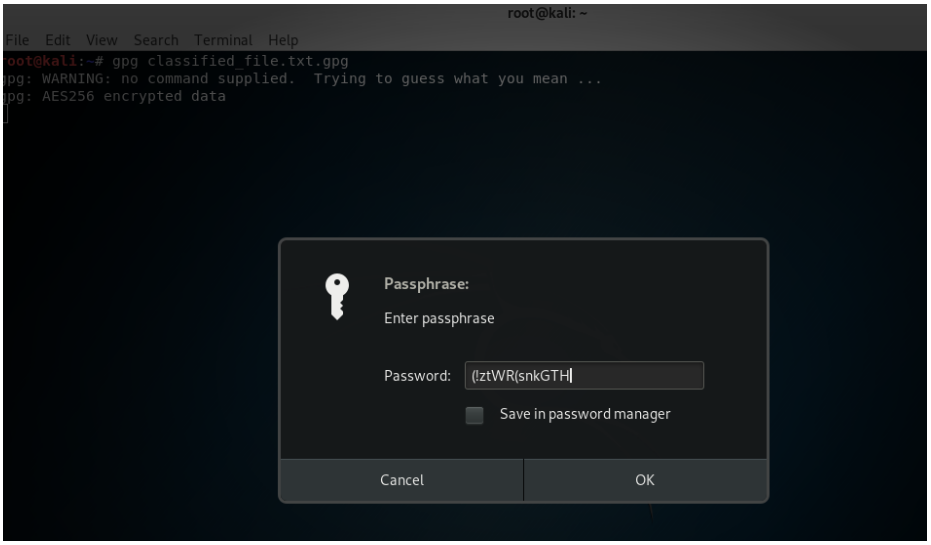
This is what happens if you try to **cat** the encrypted file



```
root@kali:~# ls
bettercap               Desktop    Downloads  Music          Pictures  ransomware  Templates     Videos
classified_file.txt.gpg Documents  EH-07-L2   PASSWORDGEN.py Public    RockYou.txt userlist.txt
root@kali:~# cat classified_file.txt.gpg
�       �$d
           �]���`��g�j��h7�a%ʒ`���xQ    +V�
�������
      $�ȯ�[O��Cy�=�Aʅ  E#�uI�-%��#��ö��r�2�k�Cd��WIroot@kali:~# 
```

To get the contents of the file, you have to run this command to bypass the key cache to ensure that the passphrase requires you to enter the password

gpg **- -no-symkey-cache** classified.txt.gpg



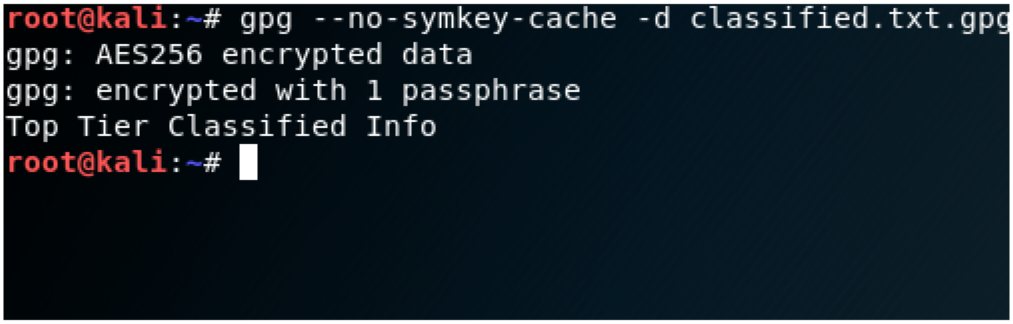
```
oot@kali:~# gpg classified_file.txt.gpg
pg: WARNING: no command supplied.  Trying to guess what you mean ...
pg: AES256 encrypted data
```

Passphrase:

Enter passphrase

Password: (!ztWR(snkGTH

□ Save in password manager



```
root@kali:~# gpg --no-symkey-cache -d classified.txt.gpg
gpg: AES256 encrypted data
gpg: encrypted with 1 passphrase
Top Tier Classified Info
root@kali:~# 
```

Conclusion : In this lab, we learned a new tool called gpg, which is used for secure encryption/decryption. But the important part is that we got the script to give us random generated high secure passwords, this is a key script to have if you get caught needing a random password.