V E R Z E O

MAJOR PROJECT ASSIGNMENT

BY: PRERNA SIDANA

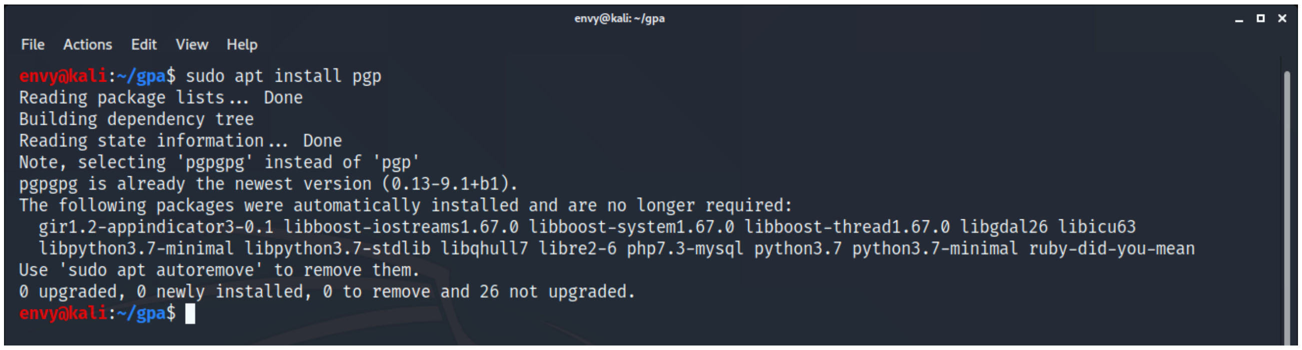
GROUP NO. – CS06B2

JUNE BATCH 2020

MENTOR: ANIMESH ROY

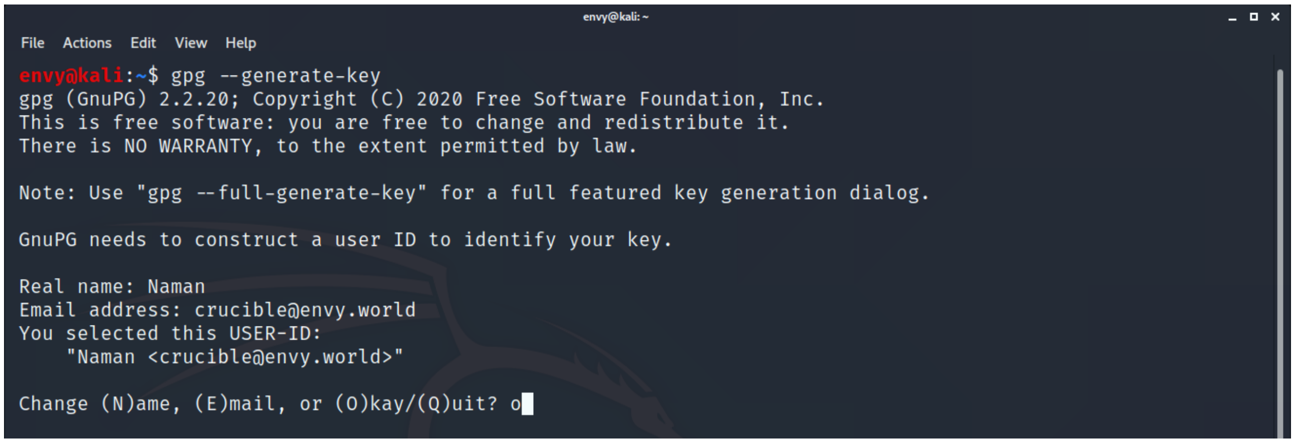
Activity 1: PGP Encrypted Mail

1. Get the PGP package using the terminal by using the command ‘sudo apt install pgp’.



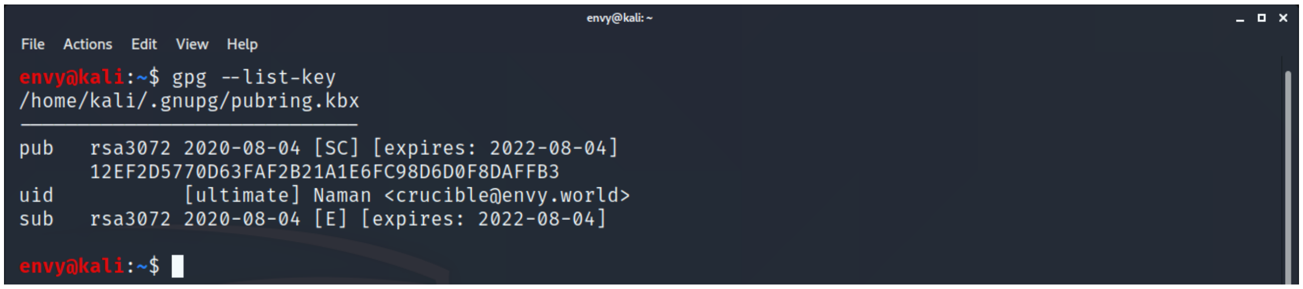
```
envy@kali:~/gpa$ sudo apt install pgp
Reading package lists... Done
Building dependency tree
Reading state information... Done
Note, selecting 'pgpgpg' instead of 'pgp'
pgpgpg is already the newest version (0.13-9.1+b1).
The following packages were automatically installed and are no longer required:
  gir1.2-appindicator3-0.1 libboost-iostreams1.67.0 libboost-system1.67.0 libboost-thread1.67.0 libgdal26 libicu63
  libpython3.7-minimal libpython3.7-stdlib libqhull7 libre2-6 php7.3-mysql python3.7 python3.7-minimal ruby-did-you-mean
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 26 not upgraded.
envy@kali:~/gpa$
```

2. Generate a key pair using the command ‘gpg –generate-key’ & Real name & email address and then supplying the system with raw data moving the mouse pointer on the screen to generate the key.



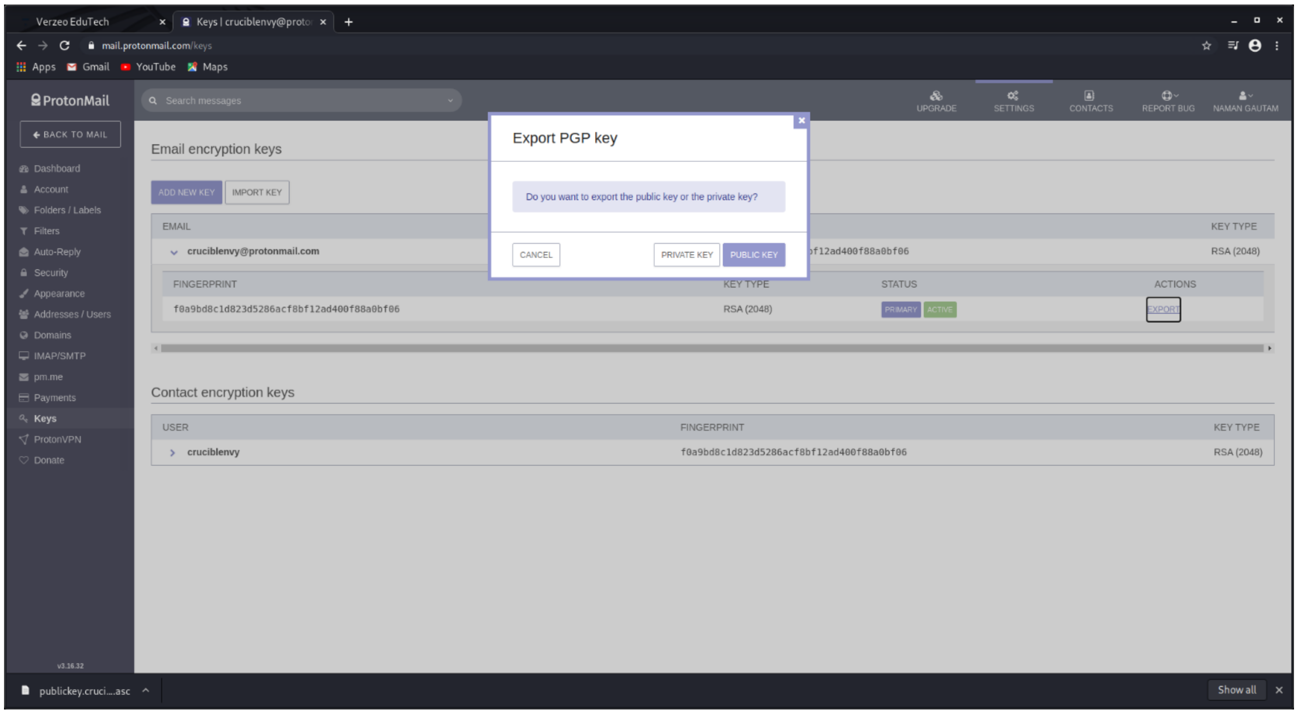
```
envy@kali:~$ gpg --generate-key
gpg (GnuPG) 2.2.20; Copyright (C) 2020 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Note: Use "gpg --full-generate-key" for a full featured key generation dialog.

GnuPG needs to construct a user ID to identify your key.

Real name: Naman
Email address: crucible@envy.world
You selected this USER-ID:
    "Naman <crucible@envy.world>"

Change (N)ame, (E)mail, or (O)kay/(Q)uit? o
```

3. List all the keys in the system by using the command ‘gpg –list-key’. This will display the public keys with the name and email id details.



```
envy@kali:~$ gpg --list-key
/home/kali/.gnupg/pubring.kbx
-----------------------------
pub   rsa3072 2020-08-04 [SC] [expires: 2022-08-04]
      12EF2D5770D63FAF2B21A1E6FC98D6D0F8DAFFB3
uid           [ultimate] Naman <crucible@envy.world>
sub   rsa3072 2020-08-04 [E] [expires: 2022-08-04]

envy@kali:~$
```

4. We will be using ProtonMail for this exercise as it provides a public and private key beforehand. After creating an account, export the public key from the settings of the mailbox.


Export PGP key

Do you want to export the public key or the private key?

CANCEL PRIVATE KEY PUBLIC KEY

Email encryption keys

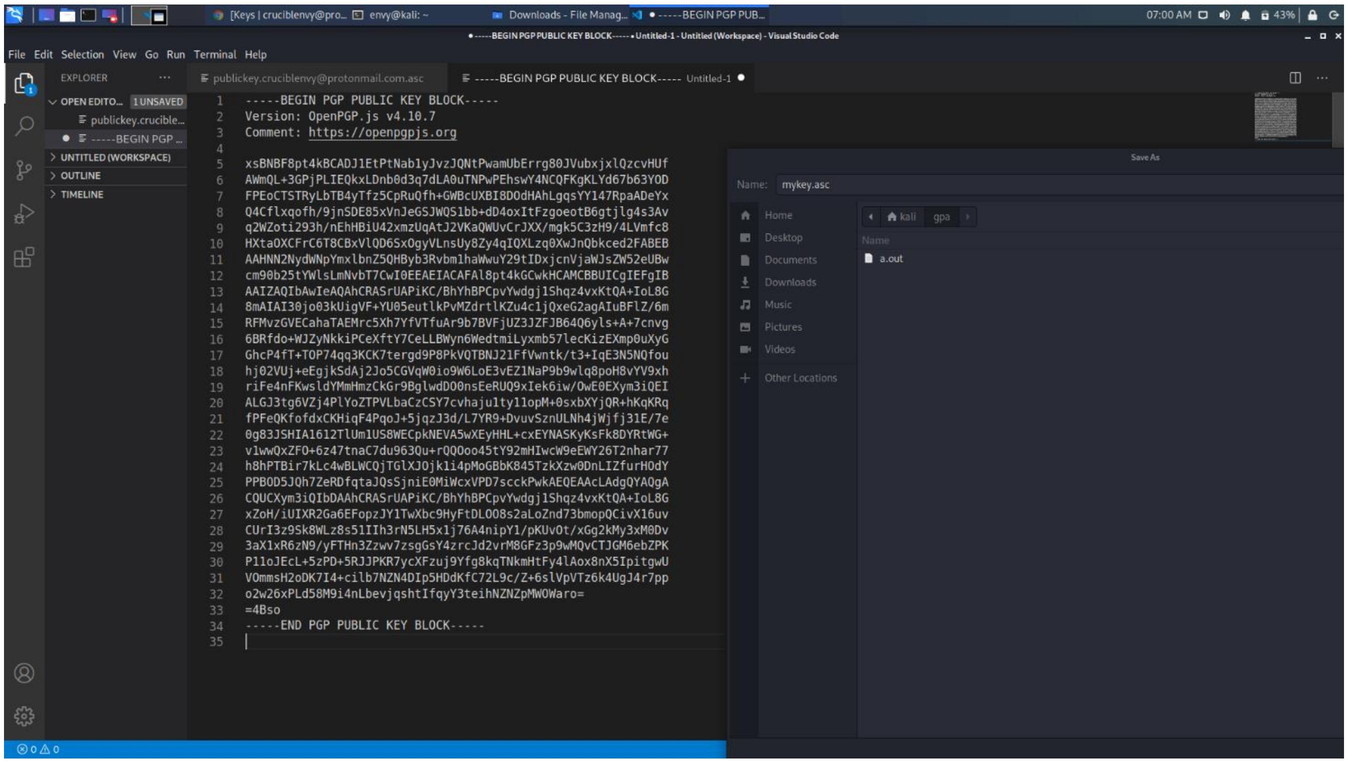
EMAIL		KEY TYPE
cruciblenvy@protonmail.com	f12ad400f88a0bf06	RSA (2048)

FINGERPRINT	KEY TYPE	STATUS	ACTIONS
f0a9bd8c1d823d5286acf8bf12ad400f88a0bf06	RSA (2048)	PRIMARY ACTIVE	EXPORT

Contact encryption keys

USER	FINGERPRINT	KEY TYPE
cruciblenvy	f0a9bd8c1d823d5286acf8bf12ad400f88a0bf06	RSA (2048)

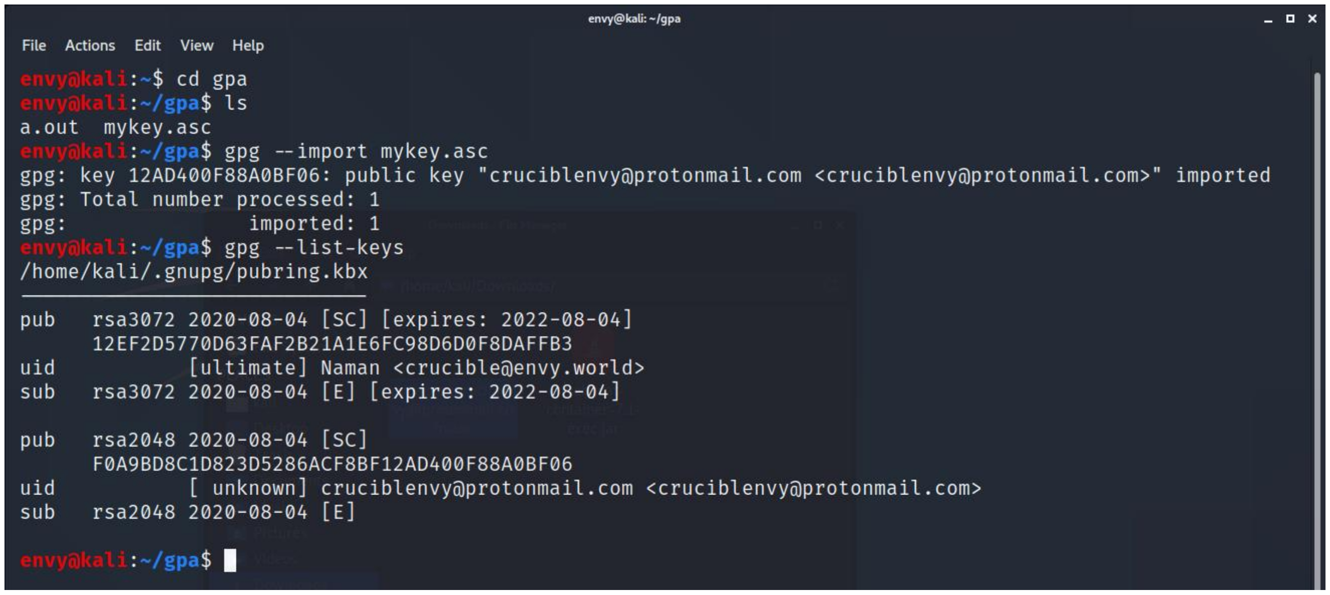
5. Open the key file downloaded from ProtonMail and save it as mykey.asc for easier access to the file.


```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: OpenPGP.js v4.10.7
Comment: https://openpgpjs.org

xsBNBF8pt4kBCADJ1EtPtNab1yJvzJQNtPwamUbErrg80JVubxjxlQzcvHUf
AWmQL+3GPjPLIEQkxLDnb0d3q7dLA0uTNPwPEhswY4NCQFKgKLYd67b63YOD
FPEoCTSTRyLbTB4yTfz5CpRuQfh+GWBcUXBI8DOdHAhLgqsYY147RpaADeYx
Q4Cflxqofh/9jnSDE85xVnJeGSJWQS1bb+dD4oxItFzgoeotB6gtjlg4s3Av
q2WZoti293h/nEhHBiU42xmzUqAtJ2VKaQWUvCrJXX/mgk5C3zH9/4LVmfc8
HXtaOXCFrC6T8CBxVlQD6SxOgyVLnsUy8Zy4qIQXLzq0XwJnQbkced2FABEB
AAHNN2NydWNpYmxlbnZ5QHByb3Rvbm1haWwuY29tIDxjcnVjaWJsZW52eUBw
cm90b25tYWlsLmNvbT7CwI0EEAEIACAFAl8pt4kGCwkHCAMCBBUICgIEFgIB
AAIZAQIbAwIeAQAhCRASrUAPiKC/BhYhBPCpvYwdgj1Shqz4vxKtQA+IoL8G
8mAIAI30jo03kUigVF+YU05eutlkPvMZdrtlKZu4c1jQxeG2agAIuBFlZ/6m
RFMvzGVECahaTAEMrc5Xh7YfVTfuAr9b7BVFjUZ3JZFJB64Q6yls+A+7cnvg
6BRfdo+WJZyNkkiPCeXftY7CeLLBWyn6WedtmiLyxmb57lecKizEXmp0uXyG
GhcP4fT+TOP74qq3KCK7tergd9P8PkVQTBNJ21FfVwntk/t3+IqE3N5NQfou
hj02VUj+eEgjkSdAj2Jo5CGVqW0io9W6LoE3vEZ1NaP9b9wlq8poH8vYV9xh
riFe4nFKwsldYMmHmzCkGr9BglwdDO0nsEeRUQ9xIek6iw/OwE0EXym3iQEI
ALGJ3tg6VZj4PlYoZTPVLbaCzCSY7cvhaju1ty1lopM+0sxbXYjQR+hKqKRq
fPFeQKfofdxCKHiqF4PqoJ+5jqzJ3d/L7YR9+DvuvSznULNh4jWjfj31E/7e
0g83JSHIA1612TlUm1US8WECpkNEVA5wXEyHHL+cxEYNASKyKsFk8DYRtWG+
v1wwQxZFO+6z47tnaC7du963Qu+rQQOoo45tY92mHIwcW9eEWY26T2nhar77
h8hPTBir7kLc4wBLWCQjTGlXJOjk1i4pMoGBbK845TzkXzw0DnLIZfurHOdY
PPBOD5JQh7ZeRDfqtaJQsSjniE0MiWcxVPD7scckPwkAEQEAAcLAdgQYAQgA
CQUCXym3iQIbDAAhCRASrUAPiKC/BhYhBPCpvYwdgj1Shqz4vxKtQA+IoL8G
xZoH/iUIXR2Ga6EFopzJY1TwXbc9HyFtDLOO8s2aLoZnd73bmopQCivX16uv
CUrI3z9Sk8WLz8s51IIh3rN5LH5x1j76A4nipY1/pKUvOt/xGg2kMy3xM0Dv
3aX1xR6zN9/yFTHn3Zzwv7zsgGsY4zrcJd2vrM8GFz3p9wMQvCTJGM6ebZPK
P11oJEcL+5zPD+5RJJPKR7ycXFzuj9Yfg8kqTNkmHtFy4lAox8nX5IpitgwU
VOmmsH2oDK7I4+cilb7NZN4DIp5HDdKfC72L9c/Z+6slVpVTz6k4UgJ4r7pp
o2w26xPLd58M9i4nLbevjqshtIfqyY3teihNZNZpMWOWaro=
=4Bso
-----END PGP PUBLIC KEY BLOCK-----
```

Save As

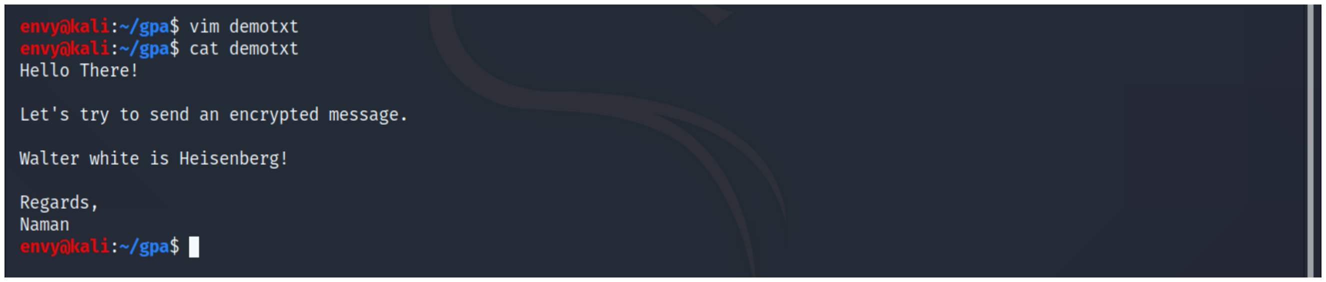
Name: mykey.asc

6. Navigate to the folder where you saved the file 'mykey.asc' and use the command 'gpg –import mykey.asc.' You can also use the 'gpg –list-keys' again to list the keys and see if the new key has been added to the database.



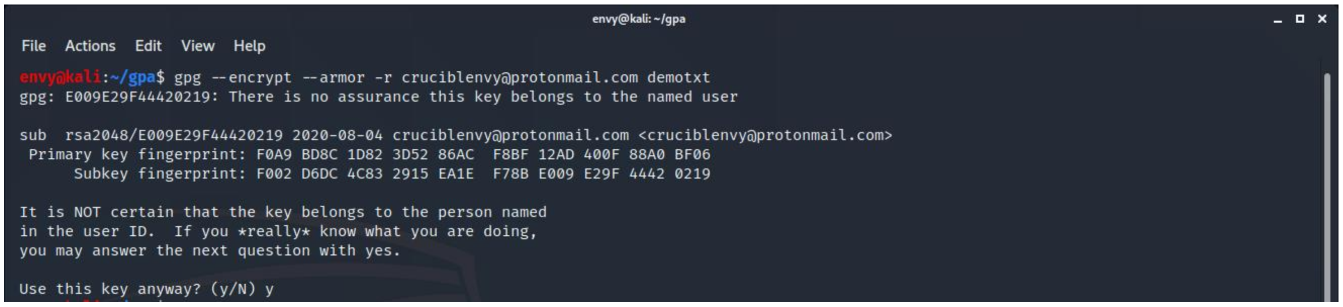
```
envy@kali:~$ cd gpa
envy@kali:~/gpa$ ls
a.out  mykey.asc
envy@kali:~/gpa$ gpg --import mykey.asc
gpg: key 12AD400F88A0BF06: public key "cruciblenvy@protonmail.com <cruciblenvy@protonmail.com>" imported
gpg: Total number processed: 1
gpg:               imported: 1
envy@kali:~/gpa$ gpg --list-keys
/home/kali/.gnupg/pubring.kbx
-----------------------------
pub   rsa3072 2020-08-04 [SC] [expires: 2022-08-04]
      12EF2D5770D63FAF2B21A1E6FC98D6D0F8DAFFB3
uid           [ultimate] Naman <crucible@envy.world>
sub   rsa3072 2020-08-04 [E] [expires: 2022-08-04]

pub   rsa2048 2020-08-04 [SC]
      F0A9BD8C1D823D5286ACF8BF12AD400F88A0BF06
uid           [ unknown] cruciblenvy@protonmail.com <cruciblenvy@protonmail.com>
sub   rsa2048 2020-08-04 [E]

envy@kali:~/gpa$ 
```

7. Use the vim command to create a file named 'demotxt' and enter the message which you want to send as the email. Confirm if the file has been saved properly by using the cat command.

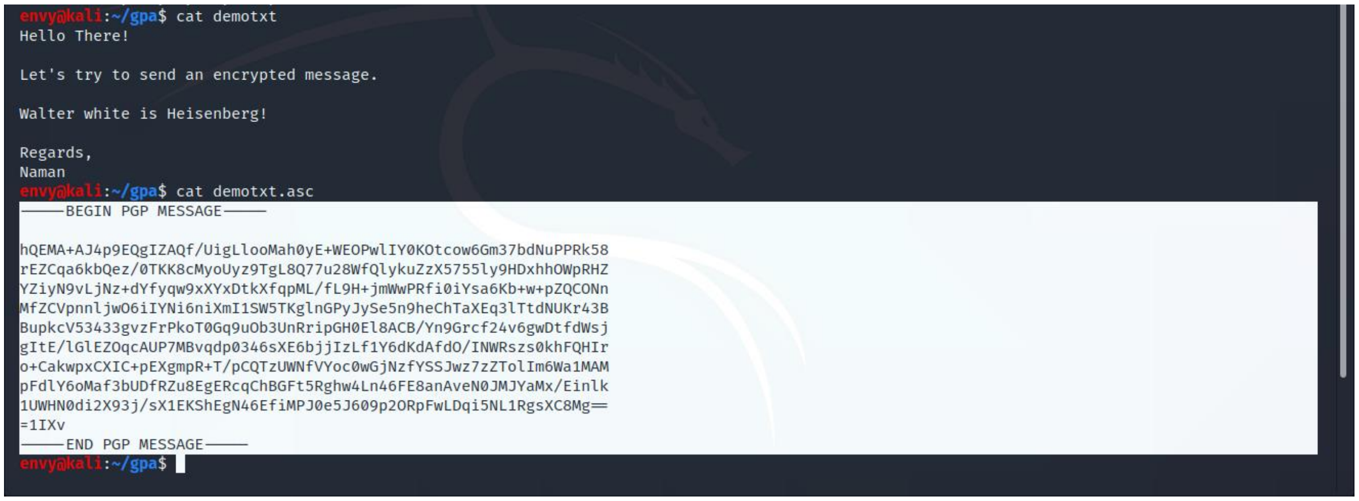


```
envy@kali:~/gpa$ vim demotxt
envy@kali:~/gpa$ cat demotxt
Hello There!

Let's try to send an encrypted message.

Walter white is Heisenberg!

Regards,
Naman
envy@kali:~/gpa$ 
```

8. We will encrypt the ‘demotxt’ file now using the public key of our ProtonMail email address by using the command ‘gpg –encrypt –armor -r sample@xyz.com demotxt’. A new file ‘demotxt.asc’ will be created. Now read the file using the cat command and copy the contents of the file.

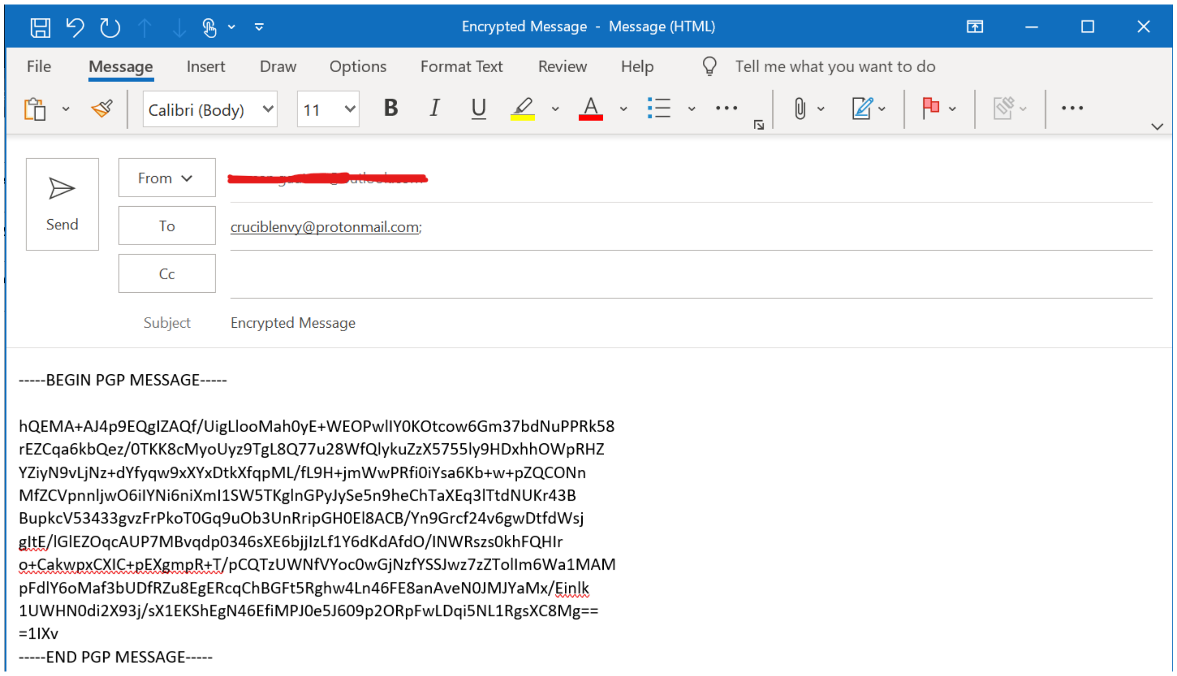


```
envy@kali:~/gpa$ gpg --encrypt --armor -r cruciblenvy@protonmail.com demotxt
gpg: E009E29F44420219: There is no assurance this key belongs to the named user

sub  rsa2048/E009E29F44420219 2020-08-04 cruciblenvy@protonmail.com <cruciblenvy@protonmail.com>
 Primary key fingerprint: F0A9 BD8C 1D82 3D52 86AC  F8BF 12AD 400F 88A0 BF06
      Subkey fingerprint: F002 D6DC 4C83 2915 EA1E  F78B E009 E29F 4442 0219

It is NOT certain that the key belongs to the person named
in the user ID.  If you *really* know what you are doing,
you may answer the next question with yes.

Use this key anyway? (y/N) y
```



```
envy@kali:~/gpa$ cat demotxt
Hello There!

Let's try to send an encrypted message.

Walter white is Heisenberg!

Regards,
Naman
envy@kali:~/gpa$ cat demotxt.asc
-----BEGIN PGP MESSAGE-----

hQEMA+AJ4p9EQgIZAQf/UigLlooMah0yE+WEOPwlIY0KOtcow6Gm37bdNuPPRk58
rEZCqa6kbQez/0TKK8cMyoUyz9TgL8Q77u28WfQlykuZzX5755ly9HDxhhOWpRHZ
YZiyN9vLjNz+dYfyqw9xXYxDtkXfqpML/fL9H+jmWwPRfi0iYsa6Kb+w+pZQCONn
MfZCVpnnljwO6iIYNi6niXmI1SW5TKglnGPyJySe5n9heChTaXEq3lTtdNUKr43B
BupkcV53433gvzFrPkoT0Gq9uOb3UnRripGH0El8ACB/Yn9Grcf24v6gwDtfdWsj
gItE/lGlEZOqcAUP7MBvqdp0346sXE6bjjIzLf1Y6dKdAfdO/INWRszs0khFQHIr
o+CakwpxCXIC+pEXgmpR+T/pCQTzUWNfVYoc0wGjNzfYSSJwz7zZTolIm6Wa1MAM
pFdlY6oMaf3bUDfRZu8EgERcqChBGFt5Rghw4Ln46FE8anAveN0JMJYaMx/Einlk
1UWHN0di2X93j/sX1EKShEgN46EfiMPJ0e5J609p2ORpFwLDqi5NL1RgsXC8Mg==
=1IXv
-----END PGP MESSAGE-----
envy@kali:~/gpa$
```

9. Now we will paste the encrypted text into the message body, and we can send it from any email address to the recipient whose public key we had used for encryption.

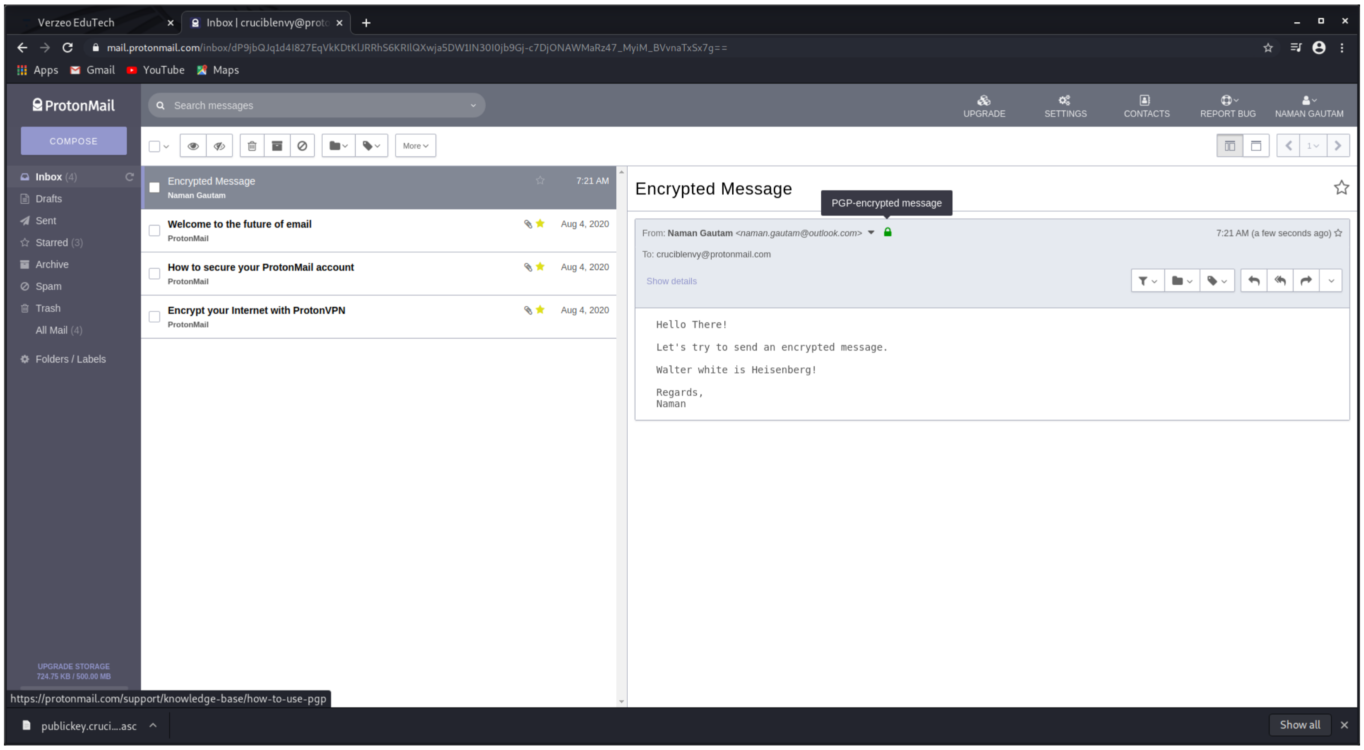


Encrypted Message - Message (HTML)
File Message Insert Draw Options Format Text Review Help Tell me what you want to do
Calibri (Body) 11
Send

From [redacted]
To cruciblenvy@protonmail.com;
Cc
Subject Encrypted Message

```
-----BEGIN PGP MESSAGE-----

hQEMA+AJ4p9EQgIZAQf/UigLlooMah0yE+WEOPwlIY0KOtcow6Gm37bdNuPPRk58
rEZCqa6kbQez/0TKK8cMyoUyz9TgL8Q77u28WfQlykuZzX5755ly9HDxhhOWpRHZ
YZiyN9vLjNz+dYfyqw9xXYxDtkXfqpML/fL9H+jmWwPRfi0iYsa6Kb+w+pZQCONn
MfZCVpnnljwO6iIYNi6niXmI1SW5TKglnGPyJySe5n9heChTaXEq3lTtdNUKr43B
BupkcV53433gvzFrPkoT0Gq9uOb3UnRripGH0El8ACB/Yn9Grcf24v6gwDtfdWsj
gItE/lGlEZOqcAUP7MBvqdp0346sXE6bjjIzLf1Y6dKdAfdO/INWRszs0khFQHIr
o+CakwpxCXIC+pEXgmpR+T/pCQTzUWNfVYoc0wGjNzfYSSJwz7zZTolIm6Wa1MAM
pFdlY6oMaf3bUDfRZu8EgERcqChBGFt5Rghw4Ln46FE8anAveN0JMJYaMx/Einlk
1UWHN0di2X93j/sX1EKShEgN46EfiMPJ0e5J609p2ORpFwLDqi5NL1RgsXC8Mg==
=1IXv
-----END PGP MESSAGE-----
```

10. We can see in our mailbox that we have received the mail. As soon as we open it and read its contents, we can see that the text we encrypted is displayed as plaintext into the mail.



Encrypted Message	Naman Gautam	7:21 AM
Welcome to the future of email	ProtonMail	Aug 4, 2020
How to secure your ProtonMail account	ProtonMail	Aug 4, 2020
Encrypt your Internet with ProtonVPN	ProtonMail	Aug 4, 2020

Encrypted Message

From: Naman Gautam <naman.gautam@outlook.com> 7:21 AM (a few seconds ago)

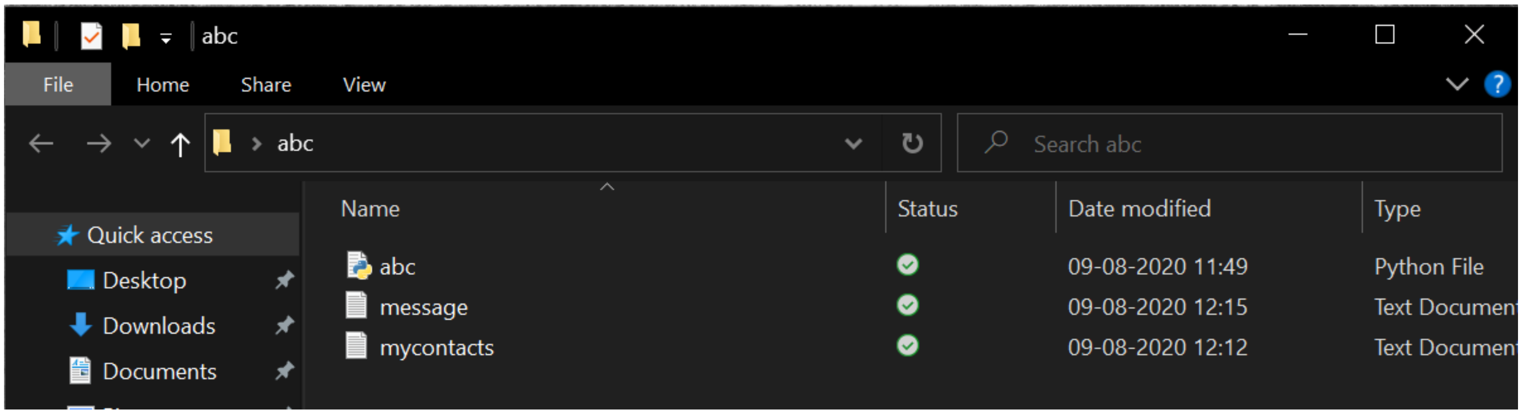
To: cruciblenvy@protonmail.com

```
Hello There!

Let's try to send an encrypted message.

Walter white is Heisenberg!

Regards,
Naman
```

This is how we send an encrypted PGP mail.

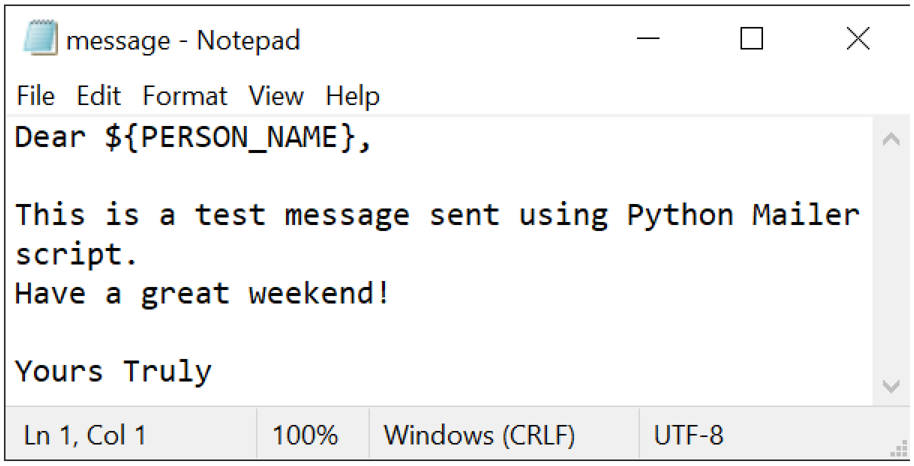
Activity 2 : Python Mailer Script

1. Create a folder in which there should be a script(abc.py), a message file(message.txt) & a contacts file.



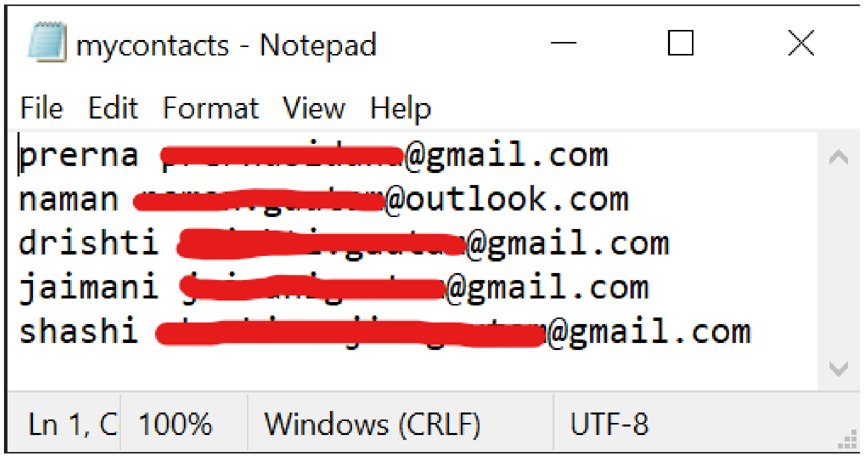
Name	Status	Date modified	Type
abc	✓	09-08-2020 11:49	Python File
message	✓	09-08-2020 12:15	Text Documen
mycontacts	✓	09-08-2020 12:12	Text Documen

message.txt - This file will contain the email message which has to be sent to the recipient.



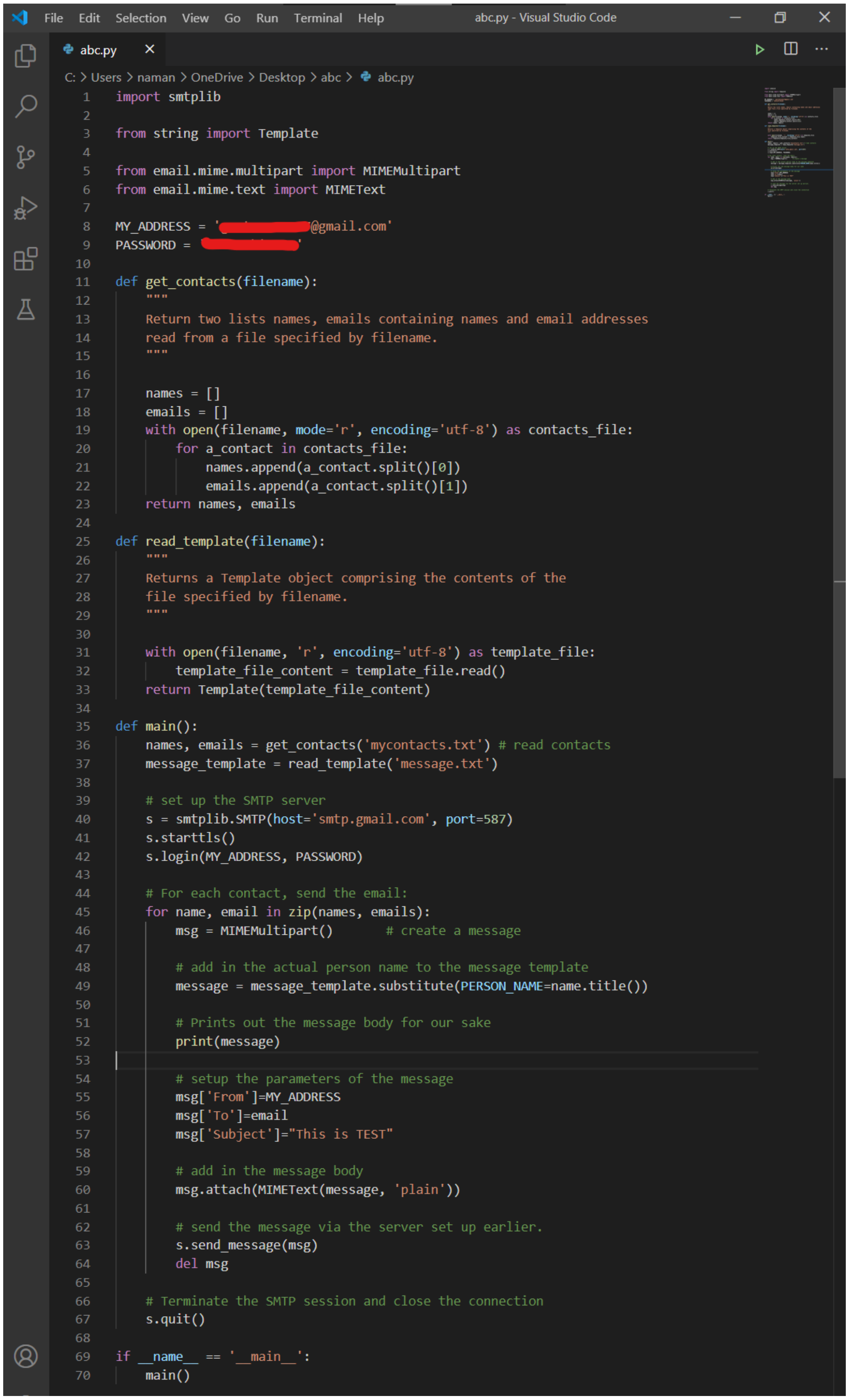
```
Dear ${PERSON_NAME},

This is a test message sent using Python Mailer
script.
Have a great weekend!

Yours Truly
```

mycontacts.txt - This file will contain all of the Recipient names and email addresses to whom the email has to be sent.



```
prerna [redacted]@gmail.com
naman [redacted]@outlook.com
drishti [redacted]@gmail.com
jaimani [redacted]@gmail.com
shashi [redacted]@gmail.com
```

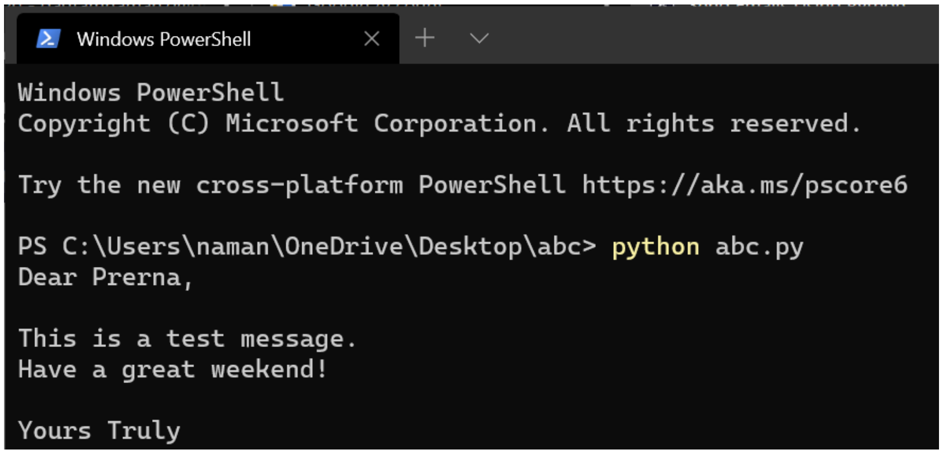
abc.py - Contains the main python script which will be executed later on for the actual activity.


File Edit Selection View Go Run Terminal Help
abc.py - Visual Studio Code
abc.py
C: > Users > naman > OneDrive > Desktop > abc > abc.py

```python
import smtplib

from string import Template

from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText

MY_ADDRESS = '[redacted]@gmail.com'
PASSWORD = [redacted]

def get_contacts(filename):
    """
    Return two lists names, emails containing names and email addresses
    read from a file specified by filename.
    """

    names = []
    emails = []
    with open(filename, mode='r', encoding='utf-8') as contacts_file:
        for a_contact in contacts_file:
            names.append(a_contact.split()[0])
            emails.append(a_contact.split()[1])
    return names, emails

def read_template(filename):
    """
    Returns a Template object comprising the contents of the
    file specified by filename.
    """

    with open(filename, 'r', encoding='utf-8') as template_file:
        template_file_content = template_file.read()
    return Template(template_file_content)

def main():
    names, emails = get_contacts('mycontacts.txt') # read contacts
    message_template = read_template('message.txt')

    # set up the SMTP server
    s = smtplib.SMTP(host='smtp.gmail.com', port=587)
    s.starttls()
    s.login(MY_ADDRESS, PASSWORD)

    # For each contact, send the email:
    for name, email in zip(names, emails):
        msg = MIMEMultipart()       # create a message

        # add in the actual person name to the message template
        message = message_template.substitute(PERSON_NAME=name.title())

        # Prints out the message body for our sake
        print(message)

        # setup the parameters of the message
        msg['From']=MY_ADDRESS
        msg['To']=email
        msg['Subject']="This is TEST"

        # add in the message body
        msg.attach(MIMEText(message, 'plain'))

        # send the message via the server set up earlier.
        s.send_message(msg)
        del msg

    # Terminate the SMTP session and close the connection
    s.quit()

if __name__ == '__main__':
    main()
```

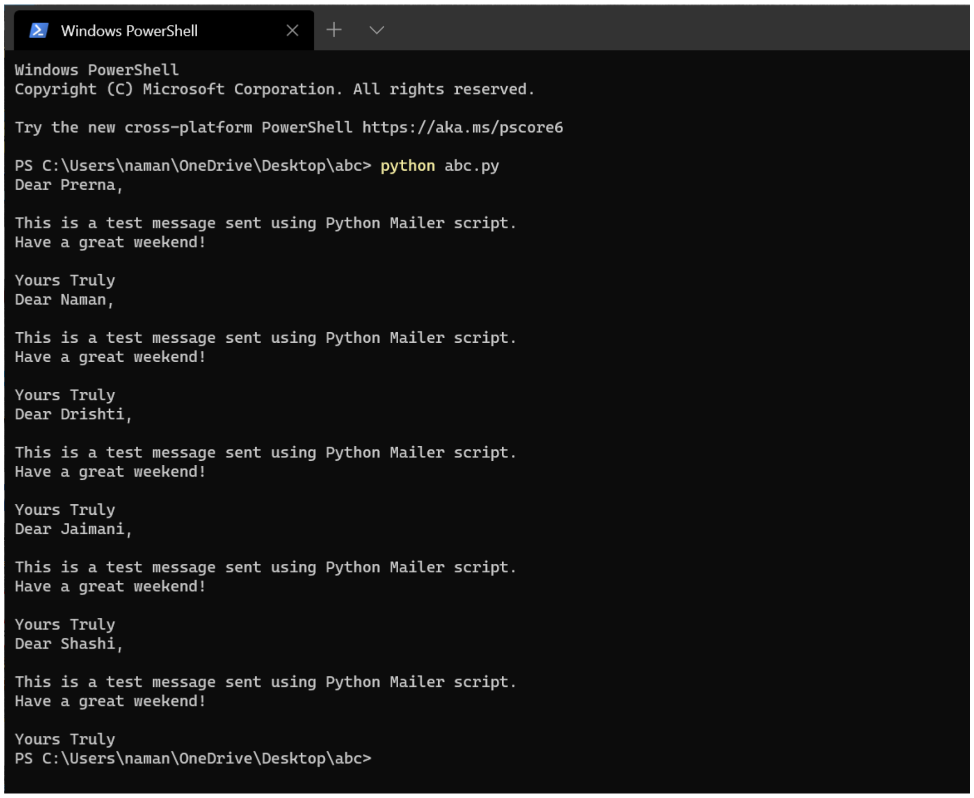
2. The script abc.py should contain the email address and the password filled out to facilitate the sending of emails from that account. Also, the server settings should be configured. In our case, we have configured the server address as ‘smpt.gmail.com’ & server tls port as ‘587’.
3. After doing the above settings all we need to do is run the script. The script picks up the names & email addresses of the recipients and then imposes it with the template of ‘message.txt’. After superimposing it, the emails are sent out.

- When we try to send it to a single recipient:



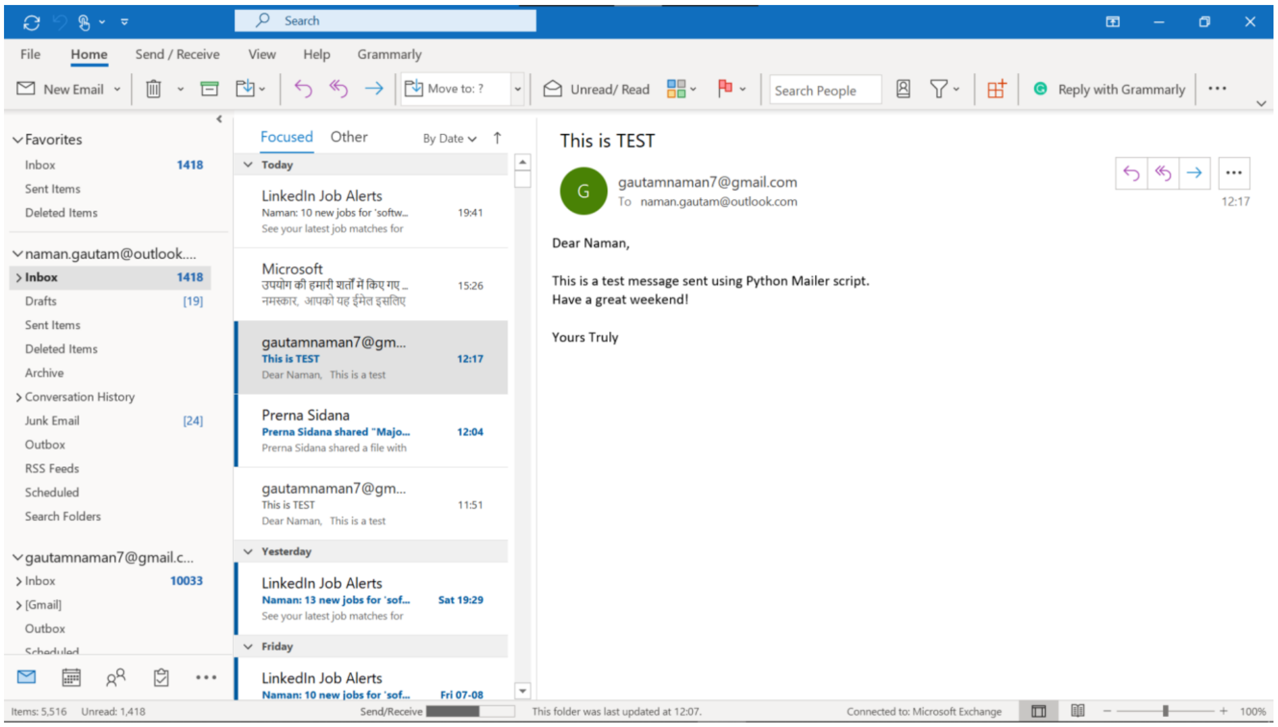
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\naman\OneDrive\Desktop\abc> python abc.py
Dear Prerna,

This is a test message.
Have a great weekend!

Yours Truly
```

- When we try to send it to multiple addresses :



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\naman\OneDrive\Desktop\abc> python abc.py
Dear Prerna,

This is a test message sent using Python Mailer script.
Have a great weekend!

Yours Truly
Dear Naman,

This is a test message sent using Python Mailer script.
Have a great weekend!

Yours Truly
Dear Drishti,

This is a test message sent using Python Mailer script.
Have a great weekend!

Yours Truly
Dear Jaimani,

This is a test message sent using Python Mailer script.
Have a great weekend!

Yours Truly
Dear Shashi,

This is a test message sent using Python Mailer script.
Have a great weekend!

Yours Truly
PS C:\Users\naman\OneDrive\Desktop\abc>
```

4. Check-in your recipient mailbox. You will be able to see the email in it.



LinkedIn Job Alerts
Naman: 10 new jobs for 'softw... 19:41
See your latest job matches for

Microsoft
उपयोग की हमारी शर्तों में किए गए ... 15:26
नमस्कार, आपको यह ईमेल इसलिए

gautamnaman7@gm...
This is TEST 12:17
Dear Naman, This is a test

Prerna Sidana
Prerna Sidana shared "Majo... 12:04
Prerna Sidana shared a file with

gautamnaman7@gm...
This is TEST 11:51
Dear Naman, This is a test

LinkedIn Job Alerts
Naman: 13 new jobs for 'sof... Sat 19:29
See your latest job matches for

LinkedIn Job Alerts
Naman: 10 new jobs for 'sof... Fri 07-08

This is TEST

G gautamnaman7@gmail.com
To naman.gautam@outlook.com
12:17

Dear Naman,

This is a test message sent using Python Mailer script.
Have a great weekend!

Yours Truly

This is how we can use python to send out bulk emails automatically to as many people as we want. This facilitates the people to send emails when organizing a function or any event. Also, this can be used by companies to advertise their products when needed.

Assessment Questions

Q1) (a) Why do email services "read" your email? What is their goal?

Ans) Employees of an email-service providing company are not personally ‘reading’ your emails, they are being scanned to deliver more relevant ads and search results. It is possible to opt-out of seeing these ads, but no matter what, Gmail will be scanning the content of your emails as a security measure.

Goals:

- ✓ The automated systems of email services analyze your content to provide you **personally relevant product features, such as customized search results, tailored advertising, and spam and malware detection**. This analysis occurs as the content is sent, received, and when it is stored.
- ✓ Scanning each email is necessary **to provide a free and safe service to the public.** Example: Privacy advocates disagree, pointing out that Google scans emails from non-Gmail users who have not agreed to the company's privacy policy or Gmail terms of service.
- ✓ Email messages and files stored on their cloud platforms are also scanned if provided to the user.
- ✓ To detect spam, which can often contain harmful links, by scanning each message and analyzing it for words, phrases, and links that are often used in spam messages. Example: **Google puts these messages in a special folder called ‘Spam’, so you do not have to see them or interact** with them, which decreases your risk.
- ✓ It also scans each message **to look for malware**, such as attachments containing viruses, trojans, or worms, **thereby reducing phishing** attempts.

Q1) (b) How does PGP secure email differently than Gmail?

PGP SECURITY

- Pretty Good Privacy, or PGP, was developed in the early 1990s to guarantee the security and privacy of email communications over insecure networks. Its basic concept involves the use of private and public key pairs and is implemented today in secure communications tools like ProtonMail and Signal.

- When you send an email encrypted with PGP, you use your public key like a padlock to secure the contents, in addition to using your password to authenticate with your email service. The recipient then uses their private key to unlock the padlock and read the message. Public and private keys are simply long strings of text, like passwords.

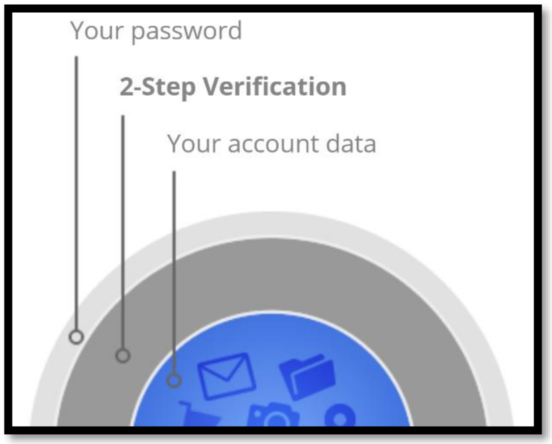
- Keys can act as an email address, in that people can send emails to your public key. Some tools expose the key functionality to the user (adding complexity), while others just control your keys in the background and handle it all for you.

- PGP can be done manually without any special email tool. If an adversary does not crack your hundreds-of-characters-long password, it is one of the most secure methods.

GMAIL SECURITY

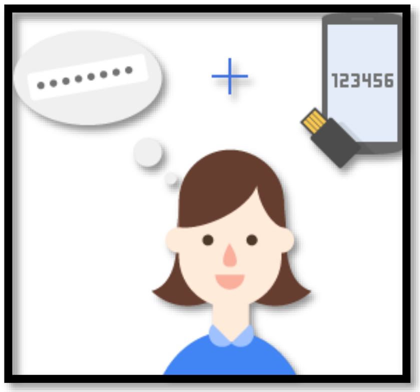
The extra layer of security

Most people only have one layer – their password – to protect their accounts. **With 2-Step Verification,** if a bad guy hacks through your password layer, he will still need your phone or Security Key to get into your account.



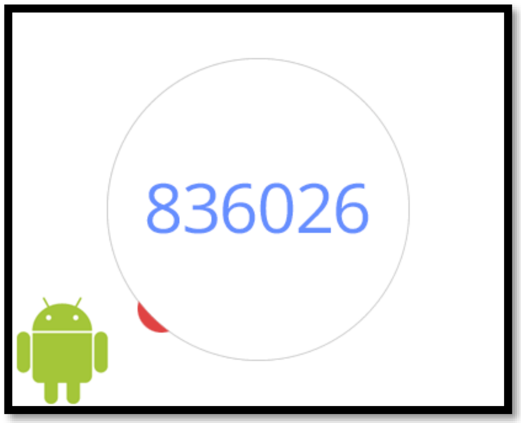
Sign-in will require something you know and something you have

With 2-Step Verification, you will protect your account with something you know (your password) and something you have (your phone or Security Key).



Verification codes made just for you

Codes are uniquely crafted for your account when you need them. If you choose to use verification codes, they will be sent to your phone via text, voice call, or our mobile app. Each code can only be used once.



✓ **Any of these common actions could put you at risk of having your password stolen:**

- Using the same password on more than one site
- Downloading software from the Internet
- Clicking on links in email messages

Q2) Why don't people use services like PGP more often?

- PGP uses encryption technologies that have evolved both technically and legally. As patents expire, for example, developers incorporated better methods into PGP. Open-source programmers also created a free version of PGP called GNU Privacy Guard, or GPG, which lacks features in the paid product, such as support for certain encryption algorithms like RSA, found in other versions of PGP.

- Compatibility Issues

Evolving versions of PGP use different methods of encryption. If you encrypt an email using PGP with one type of encryption, a recipient using PGP with a different version cannot read your message, although you may be able to decode messages sent to you.

- Complexity

The complexity and learning curve of using PGG can be intimidating. Other security schemes use symmetric encryption, which uses one key, or asymmetric encryption, which uses two. For example, when you use online banking to pay bills, your Web browser automatically sets up asymmetric encryption to protect your online session. PGP takes a hybrid approach, using symmetric encryption with two keys. It is more complex and less familiar than the traditional symmetric or asymmetric methods, so developers require more training to become effective at PGP encryption.

- No Recovery

Computer administrators frequently face emergencies involving lost or forgotten passwords. For some types of security software, an administrator can use a special program to retrieve passwords. For example, a technician who has physical access to a PC can recover forgotten log-in passwords to Microsoft Windows. PGP offers no such workaround; the encryption methods are strong, so forgotten passwords result in lost messages, lost files, or inaccessible hard drives.

Q3) What is phishing?

Phishing is an attack that cybercriminals run to get people to reveal their sensitive information unwittingly. They accomplish this by creating **fake emails and websites**, which is called spoofing.

Victims believe these spoofed emails and sites are legitimate, so they log in. As a result, the cybercriminals receive the login details, which they use to try and access other accounts across different websites. Other types of information they obtain include credit card and bank account numbers.

Q4) What is spear-phishing?

Spear phishing is a type of phishing, but more targeted. Attackers typically go after either an individual or business. As with regular phishing, cybercriminals try to trick people into handing over their credentials. However, *the goal reaches farther than just financial details.*

Instead, they aim to access sensitive company data and trade secrets. If cybercriminals can get hold of these things, they stand to make a significant amount of money by either blackmailing the organization or selling the data.