



# Cybersecurity 701

Linux Personal File  
Encryption Lab



# Linux Personal File Encryption Materials

- Materials needed
  - Kali Linux Virtual Machine
- Software Tool used
  - GnuPG - GNU Privacy Guard
    - GPG - GnuPG encryption tool



# Objectives Covered

- Security+ Objectives (SY0-701)
  - Objective 1.4 – Explain the importance of using appropriate cryptographic solutions.
    - Encryption
      - File



# What is encryption?

- Encryption is taking normal, plaintext, and applying some sort of mathematical algorithm on it to make it look random. This random looking string is known as ciphertext.
- Simple examples of encryption include a Caesar Cipher (shifting each letter a set amount in the alphabet), old newspaper Cryptoquips, and even the Enigma which the Germans used in WWII.
- The purpose is to protect data so even if someone gains access to the data, they won't be able to understand it.



# Linux Personal File Encryption Lab

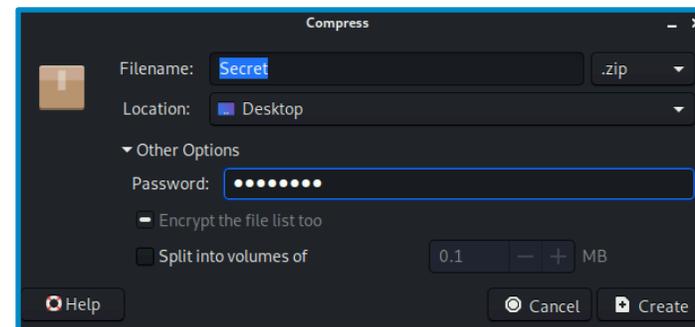
## Overview

1. Set up VM environment
2. Create a Secret File
3. Add a Password to the File
4. Remove the Original Directory
5. Open the File
6. Your Turn - Password Protecting a File
7. Create a New File
8. Create a GPG Key
9. Encrypt the File (Using GPG)
10. Decrypt the File

```
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
gpg: WARNING: server 'gpg-agent' is older than us (2.2.27 < 2.2.39)
gpg: Note: Outdated servers may lack important security fixes.
gpg: Note: Use the command "gpgconf --kill all" to restart them.
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
gpg: revocation certificate stored as '/home/kali/.gnupg/openpgp-revocs.d/10F6
public and secret key created and signed.

pub  rsa3072 2022-10-06 [SC] [expires: 2024-10-05]
     10F61BAE03786ECD42FD62D73D853112824BB92F
uid          Cyber
sub  rsa3072 2022-10-06 [E] [expires: 2024-10-05]

(kali@10.15.94.2) - [~/Desktop]
└─$
```



# Set up Environments

- Log into your range
- Open the Kali Linux Environment
  - You should be on your Kali Linux Desktop



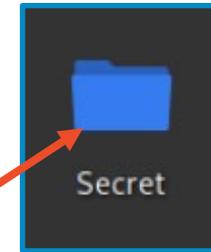
# Create a Secret File

Explore a simple way to password protect a file

- Open a terminal
- Navigate to the desktop:  
`cd Desktop`
- Create a directory named Secret:  
`mkdir Secret`

```
(kali@10.15.79.54) - [~]  
└─$ cd Desktop  
  
(kali@10.15.79.54) - [~/Desktop]  
└─$ mkdir Secret
```

You should see a folder named "Secret" appear on your Desktop



# Create a Secret File

Create a file inside of the directory

- Navigate into the Secret directory  
`cd Secret`
- Create a file called "Meeting"  
`touch Meeting`
- Open the file in the nano editor  
`nano Meeting`

```
(kali@10.15.79.54) - [~]  
$ cd Desktop  
  
(kali@10.15.79.54) - [~/Desktop]  
$ mkdir Secret  
  
(kali@10.15.79.54) - [~/Desktop]  
$ cd Secret  
  
(kali@10.15.79.54) - [~/Desktop/Secret]  
$ touch Meeting  
  
(kali@10.15.79.54) - [~/Desktop/Secret]  
$ nano Meeting
```

```
kali@kali: ~/Desktop/Secret  
ons Edit View Help  
U nano 5.8 Meeting
```

You should see  
the nano editor  
open



# Create a Secret File

Enter some text into the Meeting document

- Enter the following text in the nano editor:  
`18:00 in the park`
- Save and exit the nano editor:
  - Press **CTRL+X**
  - Type `'y'`
  - Hit **ENTER**
- Verify the text was saved
  - `cat Meeting`

```
File Edit View Terminal Tabs Help
GNU nano 5.2 Meeting
18:00 in the park
```

```
(kali@10.15.79.54) - [~/Desktop/Secret]
$ cat Meeting
18:00 in the park
```

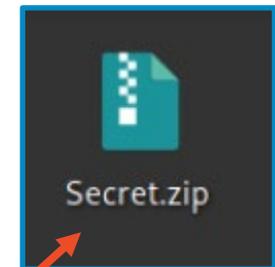
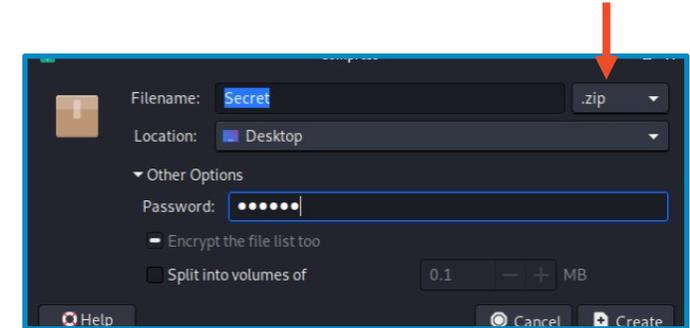
Verify the contents of Meeting were displayed in the Terminal



# Add a Password to the File

- On the desktop, right click the **Secret** directory
- Click **Create Archive...**
- Change the extension to **.zip**
- Click **Other Options**
  - This will allow you to enter a Password
- Enter **password** for the password
- Click on **Create**
  - A file named Secret.zip should be on the Desktop

Change the extension to .zip



Verify a file called Secret.zip is on the Desktop

# Remove the Original Directory

- Navigate to the Desktop

```
cd ..
```

- Remove the original directory

```
rm -r Secret
```

- Verify that the original directory has been removed from the Desktop

- Your Turn - Attempt to view the file inside of the zip without the password

```
(kali@10.15.79.54) - [~/Desktop/Secret]
$ cd ..

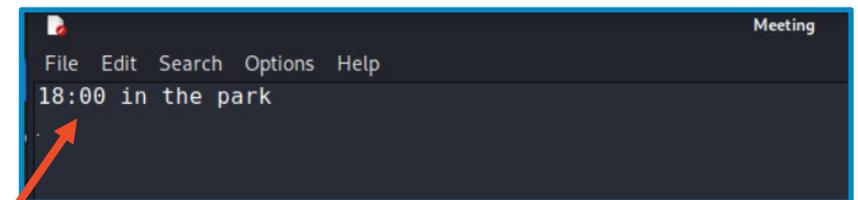
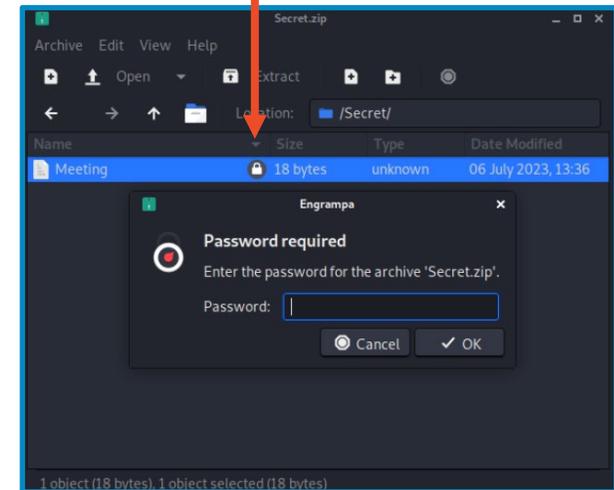
(kali@10.15.79.54) - [~/Desktop]
$ rm -r Secret
```



# Open the File

- Double click the Zip
  - It should open
- Double click the Secret Directory
  - That should open
- Double click on the Meeting file
  - You should be asked to enter a password
- Enter “password” and hit ENTER
  - This should open the file

Notice a lock icon next to the Meeting file



Verify the message inside of the Meeting file

# Your Turn - With a Partner

- Using the same encryption method
  - Create a new file inside a folder
  - Password protect the file
  - Delete the original file
  - Have another student attempt to open the file
  - Open the file with your password to verify the file



# Using GPG to Encrypt Files

- GPG can be used to encrypt files
  - More than just applying a password
- GPG - GNU Privacy Guard
  - Free-software replacement for Symantec's PGP (Pretty Good Privacy) cryptographic software suite
- GPG software will generate a key that could be shared among trusted users
  - Used to encrypt and decrypt messages
- Make sure GPG is installed and/or up to date with the following command:

```
sudo apt-get install gnupg -y
```

**\*If you're prompted to allow changes, hit enter**



# Create the File to Encrypt

- Make sure you are on the Desktop directory
- Create a file named Not\_Secret.txt  
`touch Not_Secret.txt`
- Open the file in the nano editor  
`nano Not_Secret.txt`
- Edit the file  
Enter **“Do or do not, there is no try”**
- Save and Exit the nano editor  
Press **CTRL+X**  
Type **‘y’**  
Hit **ENTER**

```
(kali@10.15.79.54) - [~/Desktop]
$ touch Not_Secret.txt

(kali@10.15.79.54) - [~/Desktop]
$ nano Not_Secret.txt
```

```
File Actions Edit View Help
GNU nano 5.8                               Not_Secret.txt
Do or do not, there is no try.
```



# Creating the Key

- Start to create a key
  - `gpg --gen-key`
- When asked for real name
  - Enter **Cyber**
- **Leave email address blank**
  - Hit **ENTER**
- Select “Okay”
  - Type **O** and hit **ENTER**

```
(kali@10.15.79.54) - [~/Desktop]
$ gpg --gen-key
gpg (GnuPG) 2.2.40; Copyright (C) 2022 g10 Code GmbH
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

gpg: keybox '/home/kali/.gnupg/pubring.kbx' created
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: Cyber
Email address:
You selected this USER-ID:
  "Cyber"

Change (N)ame, (E)mail, or (O)kay/(Q)uit? 0
```

```
Please enter the passphrase to
protect your new key

Passphrase: 
<OK> <Cancel>
```

Verify this brings  
up the  
Passphrase  
prompt



# Creating the Key

- Use Cyber1sAw3some! for the key
  - Type **Cyber1sAw3some!** for the password
    - If you try to use a "weak" passphrase, it won't be accepted and will prompt for another passphrase
  - Type **Cyber1sAw3some!** again to confirm
  - Click on **OK**

Verify the key  
has been created



```
gpg: WARNING: server 'gpg-agent' is older than us (2.2.27 < 2.2.40)
gpg: Note: Outdated servers may lack important security fixes.
gpg: Note: Use the command "gpgconf --kill all" to restart them.
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
gpg: /home/kali/.gnupg/trustdb.gpg: trustdb created
gpg: directory '/home/kali/.gnupg/openpgp-revocs.d' created
gpg: revocation certificate stored as '/home/kali/.gnupg/openpgp-revocs.d/A7031DC99FAFE94AD7ACE89E4E345B2D5D2A506F.rev'
public and secret key created and signed.

pub  rsa3072 2023-07-06 [SC] [expires: 2025-07-05]
     A7031DC99FAFE94AD7ACE89E4E345B2D5D2A506F
uid  Cyber
sub  rsa3072 2023-07-06 [E] [expires: 2025-07-05]
```



# Encrypt the File

Now that the key has been generated, encrypt the file

- Encrypt the file:

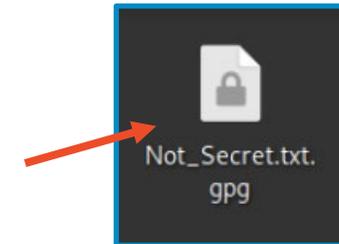
```
gpg -e -r Cyber Not_Secret.txt
```

```
(kali@10.15.79.54) - [~/Desktop]
└─$ gpg -e -r Cyber Not_Secret.txt
gpg: checking the trustdb
gpg: marginals needed: 3 completes needed: 1 trust model: pgp
gpg: depth: 0 valid: 1 signed: 0 trust: 0-, 0q, 0n, 0m, 0f, 1
u
gpg: next trustdb check due at 2025-07-05
```

- Looking at the command:

- **-e** is for encrypt
- **-r** is to encrypt for a user's name
- **"Cyber"** is the name you entered
- **Not\_Secret.txt** is the file to encrypt

Verify the  
encrypted file is  
on the Desktop



# Decrypt the Ciphertext File

- Use this command to decrypt the file:
  - `gpg -d -o Decrypted.txt Not_Secret.txt.gpg`
- Enter the password you created earlier
  - Enter **password**

```
(kali@10.15.79.54) - [~/Desktop]
└─$ gpg -d -o Decrypted.txt Not_Secret.txt.gpg
gpg: WARNING: server 'gpg-agent' is older than us (2.2.27 < 2.2.40)
gpg: Note: Outdated servers may lack important security fixes.
gpg: Note: Use the command "gpgconf --kill all" to restart them.
gpg: encrypted with 3072-bit RSA key, ID 03075AAF432CDC08, created 2
023-07-06
"Cyber"
```

```
Please enter the passphrase to unlock the OpenPGP secret key:
"Cyber"
3072-bit RSA key, ID 7FDFEA0CDE4C1977,
created 2024-11-22 (main key ID 22BD964CA69849DD).
```

```
Passphrase: █
```

```
<OK>
```

```
<Cancel>
```

```
File Edit Search Options Help
Do or do not, there is no try
```

Verify you can  
see the contents  
of the file



# Your Turn - With a Partner

- Using the same encryption method (GPG)
  - Create a new file inside a folder
  - Encrypt the file
  - Delete the original file
  - Have another student attempt to open the file
  - Open the file with your password to verify the file

