

Uso de GPG para el cifrado de archivos

Introducción a GPG y cifrado

El cifrado es una técnica que transforma datos en un formato ilegible para protegerlos de accesos no autorizados.

Cifrado simétrico: Utiliza la misma clave para cifrar y descifrar los datos.

Cifrado asimétrico: Utiliza un par de claves (clave pública y clave privada). La clave pública cifra los datos y solo la clave privada correspondiente puede descifrarlos.

Importancia del cifrado: Garantiza que solo las personas autorizadas puedan acceder a los datos, protege la confidencialidad de la información y ayuda a cumplir normativas como GDPR, ISO 27001 y otras.

GPG: GNU Privacy Guard es una herramienta de código abierto que implementa el estándar de cifrado OpenPGP, el mismo es ampliamente utilizado para: Cifrar y descifrar archivos, Crear y verificar firmas digitales, Gestionar claves públicas y privadas.

Ventajas de GPG: Código abierto y gratuito, Compatible con sistemas Windows, macOS y Linux, Garantiza la confidencialidad, integridad y autenticidad de los datos.

Informe de Creación y Gestión de Claves GPG

La seguridad de los datos es una prioridad en cualquier entorno de TI, y GPG (GNU Privacy Guard) se presenta como una herramienta esencial para garantizar la confidencialidad e integridad de la información. En este informe, se documenta el proceso de instalación, configuración, creación y gestión de claves GPG, acompañado de evidencias gráficas del procedimiento realizado.

Instalación de GPG:

Se ejecutaron los siguientes comandos para asegurarse de que GPG estuviera correctamente instalado y actualizado en el sistema:

```
sudo apt-get update && sudo apt-get upgrade -y  
sudo apt-get install gnupg
```

Se verificó que la versión de GPG instalada en el sistema es la 2.2.43, como se muestra en la primera captura de pantalla

Generación de Claves GPG:

El comando ejecutado para iniciar el proceso de generación de claves fue: `gpg --gen-key`

Detalles del proceso:

Nombre del usuario: `asusvivo`

Correo electrónico: `xxxxxx@yahoo.com`

Tipo de clave: RSA de 3072 bits.

Validez: 3 años.

Se siguieron las instrucciones en pantalla para generar entropía necesaria moviendo el ratón y utilizando el teclado. Como resultado, se generó un par de claves pública y privada, registrado en el archivo `pubring.kbx`.

Identificador: `4FAC4735C3619B0035A5E1AFDAA2509CA10C6B02`

Fecha de expiración: 4 de enero de 2028.

Listado de Claves Generadas:

Se utilizaron los comandos:

```
gpg --list-keys
gpg --list-secret-keys
```

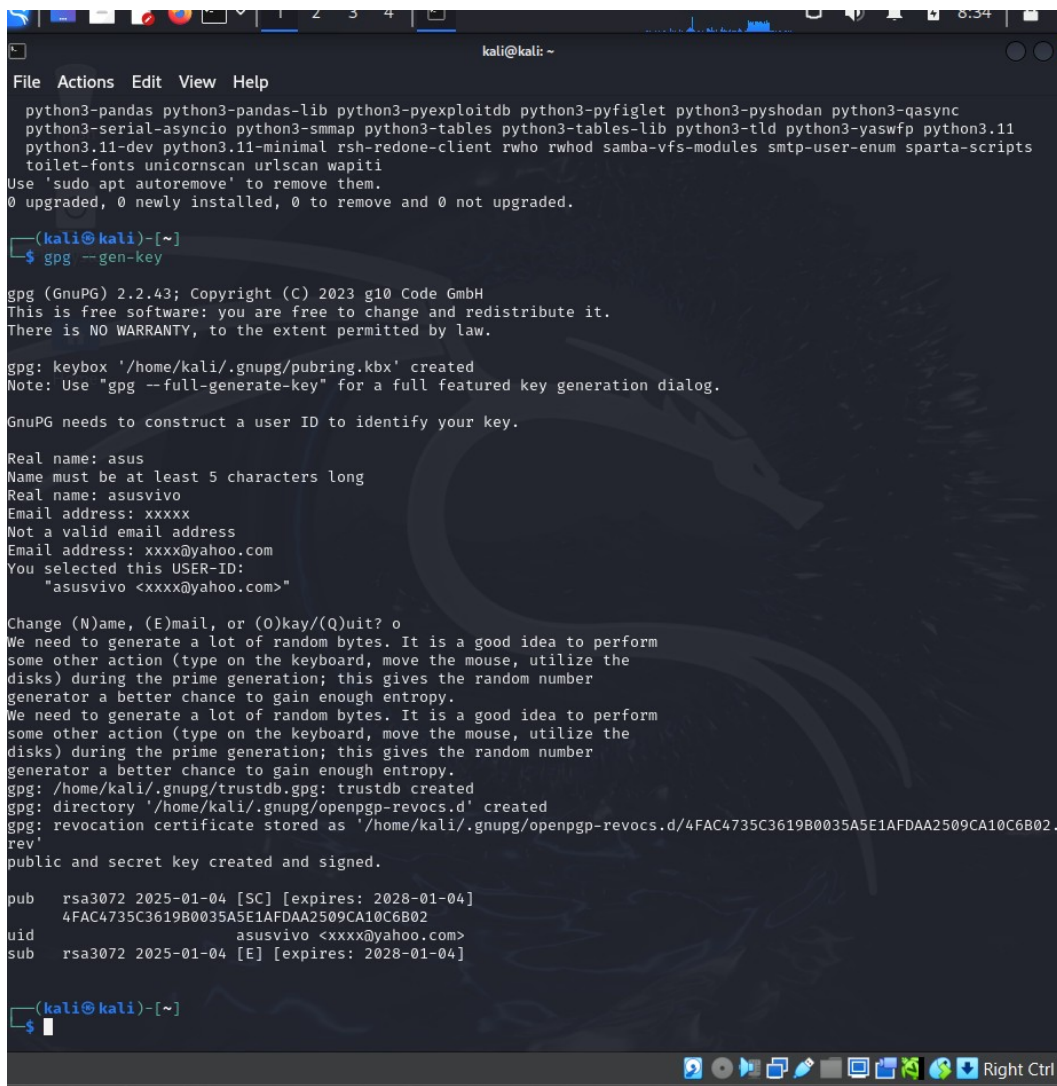
Ambos comandos enumeraron las claves públicas y secretas presentes en el sistema.

Exportación e Importación de Claves:

Para garantizar la portabilidad de las claves públicas y privadas, se ejecutaron los comandos:

```
gpg --export -a "asusvivo" > public.key
gpg --import public.key
```

La clave fue exportada exitosamente a un archivo public.key e importada nuevamente al anillo de claves del sistema.



```
kali@kali: ~
File Actions Edit View Help
python3-pandas python3-pandas-lib python3-pyexploitdb python3-pyfiglet python3-pyshodan python3-qasync
python3-serial-asyncio python3-smmap python3-tables python3-tables-lib python3-tld python3-yaswfp python3.11
python3.11-dev python3.11-minimal rsh-redone-client rwho rhod samba-vfs-modules smtp-user-enum sparta-scripts
toilet-fonts unicornscan urlscan wapiti
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.

(kali@kali)-[~]
$ gpg --gen-key

gpg (GnuPG) 2.2.43; Copyright (C) 2023 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: asus
Name must be at least 5 characters long
Real name: asusvivo
Email address: xxxxx
Not a valid email address
Email address: xxxxx@yahoo.com
You selected this USER-ID:
"asusvivo <xxxxx@yahoo.com>"

Change (N)ame, (E)mail, or (O)kay/(Q)uit? o
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.
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/4FAC4735C3619B0035A5E1AFDAA2509CA10C6B02.
rev'
public and secret key created and signed.

pub  rsa3072 2025-01-04 [SC] [expires: 2028-01-04]
     4FAC4735C3619B0035A5E1AFDAA2509CA10C6B02
uid          asusvivo <xxxxx@yahoo.com>
sub  rsa3072 2025-01-04 [E] [expires: 2028-01-04]

(kali@kali)-[~]
$
```

```
kali@kali: ~  
File Actions Edit View Help  
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: asus  
Name must be at least 5 characters long  
Real name: asusvivo  
Email address: xxxxx  
Not a valid email address  
Email address: xxxx@yahoo.com  
You selected this USER-ID:  
"asusvivo <xxxx@yahoo.com>"  
  
Change (N)ame, (E)mail, or (O)kay/(Q)uit? o  
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.  
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/4FAC4735C3619B0035A5E1AFDAA2509CA10C6B02.  
rev'  
public and secret key created and signed.  
  
pub  rsa3072 2025-01-04 [SC] [expires: 2028-01-04]  
      4FAC4735C3619B0035A5E1AFDAA2509CA10C6B02  
uid      asusvivo <xxxx@yahoo.com>  
sub  rsa3072 2025-01-04 [E] [expires: 2028-01-04]  
  
(kali@kali)-[~]  
$ gpg --list-keys  
  
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, 1u  
gpg: next trustdb check due at 2028-01-04  
/home/kali/.gnupg/pubring.kbx  
  
pub  rsa3072 2025-01-04 [SC] [expires: 2028-01-04]  
      4FAC4735C3619B0035A5E1AFDAA2509CA10C6B02  
uid      [ultimate] asusvivo <xxxx@yahoo.com>  
sub  rsa3072 2025-01-04 [E] [expires: 2028-01-04]  
  
(kali@kali)-[~]  
$
```

```
File Actions Edit View Help

pub  rsa3072 2025-01-04 [SC] [expires: 2028-01-04]
    4FAC4735C3619B0035A5E1AFDAA2509CA10C6B02
uid      asusvivo <xxxx@yahoo.com>
sub  rsa3072 2025-01-04 [E] [expires: 2028-01-04]

(kali㉿kali)-[~]
$ gpg --list-keys

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, 1u
gpg: next trustdb check due at 2028-01-04
/home/kali/.gnupg/pubring.kbx

pub  rsa3072 2025-01-04 [SC] [expires: 2028-01-04]
    4FAC4735C3619B0035A5E1AFDAA2509CA10C6B02
uid      [ultimate] asusvivo <xxxx@yahoo.com>
sub  rsa3072 2025-01-04 [E] [expires: 2028-01-04]

(kali㉿kali)-[~]
$ gpg --list-secret-keys

/home/kali/.gnupg/pubring.kbx

sec  rsa3072 2025-01-04 [SC] [expires: 2028-01-04]
    4FAC4735C3619B0035A5E1AFDAA2509CA10C6B02
uid      [ultimate] asusvivo <xxxx@yahoo.com>
ssb  rsa3072 2025-01-04 [E] [expires: 2028-01-04]

(kali㉿kali)-[~]
$ gpg --export -a "TuNombre" > public.key

gpg: WARNING: nothing exported

(kali㉿kali)-[~]
$ gpg --export -a "asusvivo" > public.key

(kali㉿kali)-[~]
$ gpg --import public.key

gpg: key DAA2509CA10C6B02: "asusvivo <xxxx@yahoo.com>" not changed
gpg: Total number processed: 1
gpg:      unchanged: 1

(kali㉿kali)-[~]
$
```

Informe de Uso de GPG para el Cifrado de Archivos

Generación de Claves GPG

Creación de un Par de Claves:

Se ejecutó el comando `gpg --gen-key` para generar una clave pública y una clave privada asociada.

Nombre real: asus

Correo electrónico: asusvivo xxxx@yahoo.com

Tipo de clave: RSA 3072 bits (expira en 3 años).

Listar Claves GPG:

Uso del comando `gpg --list-keys` para verificar la clave generada.

Detalle:

Clave: 4FAC4753C3619B0035A5E1AFDAA2509CA10C6B02

Exportación de la Clave Pública:

Se exportó la clave pública al archivo `public.key`

```
gpg --export -a "asusvivo" > public.key
```

Importación de la Clave Pública:

```
gpg --import public.key
```

Cifrado y Descifrado de Archivos

Creación del Archivo de Prueba:

Se creó un archivo de texto simple `archivo_prueba.txt` con el siguiente contenido:

Este es un archivo de prueba confidencial.

Cifrado del Archivo:

```
gpg --output archivo_cifrado.gpg --encrypt --recipient "asusvivo"
archivo_prueba.txt
```

Verificación del Archivo Cifrado:

Uso del comando `cat` para visualizar el contenido cifrado (en formato no legible).

Descifrado del Archivo:

```
gpg --output archivo_descifrado.txt --decrypt archivo_cifrado.gpg
```

Archivo descifrado exitosamente con el contenido original intacto.

Firmas Digitales

```
gpg --output firma.sig --detach-sig archivo_prueba.txt
```

Generación del archivo `firma.sig` que contiene la firma digital.

Verificación de la Firma Digital:

```
gpg --verify firma.sig archivo_prueba.txt
```

Confirmación de la autenticidad del archivo y de su creador.

```
File Actions Edit View Help

sec  rsa3072 2025-01-04 [SC] [expires: 2028-01-04]
     4FAC4735C3619B0035A5E1AFDAA2509CA10C6B02
uid      [ultimate] asusvivo <xxxx@yahoo.com>
ssb  rsa3072 2025-01-04 [E] [expires: 2028-01-04]

(kali㉿kali)-[~]
$ gpg --export -a "TuNombre" > public.key

gpg: WARNING: nothing exported

(kali㉿kali)-[~]
$ gpg --export -a "asusvivo" > public.key

Home
(kali㉿kali)-[~]
$ gpg --import public.key

gpg: key DAA2509CA10C6B02: "asusvivo <xxxx@yahoo.com>" not changed
gpg: Total number processed: 1
gpg:      unchanged: 1

(kali㉿kali)-[~]
$ echo "Este es un archivo de prueba confidencial." > archivo_prueba.txt

(kali㉿kali)-[~]
$ gpg --output archivo_cifrado.gpg --encrypt --recipient "asusvivo" archivo_prueba.txt

(kali㉿kali)-[~]
$ cat archivo_cifrado.gpg

??0%??l
  5???R???X3]??S?B6i+???;.Kk?4"??3???J?#?dy{;#~?j???^???^???a37?j??A?1Yf

?? |?m???Y???  ?z??`)LRq??U^M?x???t???
??P???

i
yg???<?Rm]???H?dB  ?????OWb?5??M???="???2??@???f?X??^?e
.N??
????3?<G?6X?H
]pJ???Y=
+FH???Gc?æ}K?I)?k?e
  ??|?||q????o?I??.*?D???`??|o???I+?i???t?Yg???N?---?Cf?(?.;????]S?K3?m???CB?z???o"??V???3???7?e???y???Z
????Tx?8?`6????YZ????8;?'?|DB?Q?P?N?;g#  ??f?Q4?p?a???T??

(kali㉿kali)-[~]
$
```



```
kali@kali: ~  
File Actions Edit View Help  
(kali@kali)-[~]  
$ gpg --export -a "asusvivo" > public.key  
  
(kali@kali)-[~]  
$ gpg --import public.key  
gpg: key DAA2509CA10C6B02: "asusvivo <xxxx@yahoo.com>" not changed  
gpg: Total number processed: 1  
gpg: unchanged: 1  
  
(kali@kali)-[~]  
$ echo "Este es un archivo de prueba confidencial." > archivo_prueba.txt  
  
(kali@kali)-[~]  
$ gpg --output archivo_cifrado.gpg --encrypt --recipient "asusvivo" archivo_prueba.txt  
  
(kali@kali)-[~]  
$ cat archivo_cifrado.gpg  
??0%?l  
5R[X3]SB6i+;;.Kk4"3}J#dy{;#~j^a37jA1Yf  
  
+|+m+Y+ z+)LRq+U^M+x+t+  
+P+  
  
i  
yg[+Rm]+H+B +OWb+5+M+"=2+@+f+X+^e  
N+  
+3+<G6X+H  
]p]++Y=  
+FH++Gc+æ}KI)+k+e  
++|+||q++++o+2+.*D++`+|o+++I++i++t+Yg+++N+-++Cf+(+.;++++]S+K3+++m++CB+z++o"++V+++3++7++y+++Z  
++++Tx+8+`6++++YZ++++8;+'+|DB+Q+p++N;g# ++f+Q4+p+a++T++  
  
(kali@kali)-[~]  
$ gpg --output archivo_descifrado.txt --decrypt archivo_cifrado.gpg  
gpg: encrypted with 3072-bit RSA key, ID 3FB74F2587FB806C, created 2025-01-04  
"asusvivo <xxxx@yahoo.com>"  
  
(kali@kali)-[~]  
$ cat archivo_descifrado.txt  
Este es un archivo de prueba confidencial.  
  
(kali@kali)-[~]  
$
```


File Actions Edit View Help

—(kali㉿kali)-[~]

—\$ echo "Este es un archivo de prueba confidencial." > archivo_prueba.txt

—(kali㉿kali)-[~]

—\$ gpg --output archivo_cifrado.gpg --encrypt --recipient "asusvivo" archivo_prueba.txt

—(kali㉿kali)-[~]

—\$ cat archivo_cifrado.gpg

```
??0%??l
      5???R???X3]??S?B6i+???;.Kk?4"??3???}J?#?dy{;#~?j???^???^???a37?j?+A?1Yf

?? |?m???Y??? ?z??`)LRq??U^M?x???t???
??P???

      i
      yg[???<??Rm]???H?dB ????OWb?5??M???="???2??@???f??X??^?e
aN??
????3?<G[6X?H
]p]???Y=
FH???Gc?æ}K?I)?k?e
      ??|?||q????o?Q??.*?D???`??|o???I+?i??t?Yg???N+---Cf?(?.;????]S?K3??m??CB?z?+o"??V???3??7?_??y???Z
????Tx?8?`6????Y?Z???????8;?`?|DB?Q?P?+N?;g# ????f?Q4?p+a??[????T??
```

—(kali㉿kali)-[~]

—\$ gpg --output archivo_descifrado.txt --decrypt archivo_cifrado.gpg

gpg: encrypted with 3072-bit RSA key, ID 3FB74F2587FB806C, created 2025-01-04
"asusvivo <xxxx@yahoo.com>"

—(kali㉿kali)-[~]

—\$ cat archivo_descifrado.txt

Este es un archivo de prueba confidencial.

—(kali㉿kali)-[~]

—\$ gpg --output firma.sig --detach-sig archivo_prueba.txt

—(kali㉿kali)-[~]

—\$ gpg --verify firma.sig archivo_prueba.txt

gpg: Signature made Sat 04 Jan 2025 08:44:03 AM EST
gpg: using RSA key 4FAC4735C3619B0035A5E1AFDAA2509CA10C6B02
gpg: Good signature from "asusvivo <xxxx@yahoo.com>" [ultimate]

—(kali㉿kali)-[~]

—\$

Right Ctrl