GPG Encryption

Transcribed on July 7, 2025 at 1:36 PM by Minutes AI

Speaker 1 (00:06)

Bienvenidos a esta nueva sesión donde vamos a trabajar sobre la herramienta GPG Tool.

Bueno, como sabéis es una herramienta de cifrado, es una herramienta que hemos visto ya un poco en sesiones anteriores sobre lo que podemos hacer y en esta sesión vamos a trabajar demostraciones, sobre todo prácticas.

Pero antes vamos a pasar a la parte de parámetros para que tengamos este listado en mente.

Como veis hay una gran cantidad de parámetros, la herramienta tiene muchos más, trabajaremos un poco sobre ellos a continuación.

Como digo, vamos a trabajar diferentes parámetros, algunos no están aquí, por ejemplo, el tema de cizado simétrico también lo trabajaremos, no se encuentra en esta lista.

Vamos a pasar ahora a la parte de la máquina virtual.

Estoy conectando la máquina virtual, estoy levantando un terminal, aquí tenemos el terminal, es una distribución de Kali y bueno, lo primero que vamos a hacer es ir viendo las diferentes posibilidades que tenemos con Grg tool.

Yo ya tengo una cuenta creada de la sesión anterior y vamos a listar las claves que tenemos disponibles.

Como se puede ver aquí, yo ya tengo esta clave, su clave pública, su clave privada, tengo la dirección de correo electrónico asociado a esa clave y cuando expira y toda esa información que nos doña importante, la parte fingerprint que aparece ahí también, que veremos después.

Vamos a generar ahora una generación de par de clave pública, clave privada, metemos el nombre nuestro de usuario, metemos un email, en este caso Pablo y y con este mecanismo es muy sencillo, fijaros, aquí nos pide un passphrase, esto para proteger la clave privada.

El uso de la clave privada es algo crítico, por lo cual nos pide que introduzcamos un passphrase.

La anterior clave, la sesión anterior yo no metí ningún valor, pero aquí sí, vamos a meter 1,23 ,4 y aquí lo vamos a tener.

Bien, le damos aquí al Ok, tomar esto y obtendremos que lanzar el proceso generando entropía.

Ahora va a salir aquí a continuación, tardamos un poquito pero enseguida tendremos el resultado de la clave pública, clave privada, ahí la tenemos.

Vamos a listar ahora las claves y veremos como en el anillo de claves, en el fichero pupring, ahí lo tenemos, tenemos ya las dos totalmente añadidas.

Si alguien nos da una clave pública, su clave pública y lo queremos importar, aquí tendríamos cómo hacerlo.

Con el parámetro import podríamos importar a nuestro fichero pum ring, anillo de claves para poder, por ejemplo, cifrarle archivos y poder mandárselos para que solamente él con su clave privada pudiera descifrarlos.

Recordad que la clave pública es la clave que nosotros podemos distribuir tranquilamente.

Bien, con esto, bueno, fijaos en la parte de fingerprint, de nuevo vuelvo a recalcarlo porque es la forma que tenemos de diferenciar qué clave estamos usando si tuviéramos dos claves con el mismo email.

Bien, vamos a hablar ahora de la parte del keyserver.

Estoy aquí introduciendo el parámetro keyserver, por ejemplo, si queremos hacer búsqueda de claves en servidores de claves en Internet, uno de los más famosos es el del meet.

Ahí tenéis el dominio y parámetro search keys y puedo introducir aquí el email que quiero buscar.

Lógicamente este no lo encontraría porque no está distribuida la clave, no está subida a ningún servidor de claves, pero si existiera, pues lo podemos encontrar y podríamos descargarla, podríamos obtenerla, introducirla en nuestro fichero de claves, nuestro anillo de claves para poder utilizarla otra vez para ese usuario.

Bien, dicho esto, bueno, ahí soy un poco pesado, pero voy a hacer otra vez lo del fingerprint porque creo que es importante.

Cada clave pública tiene un fingerprint diferente y eso es importante porque en el caso de la búsqueda con el servidor de claves nos van a dar ese fingerprint.

¿Es más, si el usuario, si un usuario me envía su clave pública, me tiene que decir oye, qué fingerprint tiene mi clave?

Porque me llega un fichero y yo tengo que validar, como veis aquí, con este comando lo podemos validar, esta clave tiene este fingerprint, esta huella y eso que tengo que validar que yo importe, la clave que yo importe e introduzca en mi fichero o ring, debe ser coherente con la información, con la huella que me dé el propietario de esa clave.

Es un mecanismo para validar que realmente estoy utilizando la clave pública real última o la que tenga que utilizar.

Como veis aquí tenemos diferentes fingerprints porque son diferentes claves públicas, las he generado yo las dos en este caso son diferentes, diferentes, lógicamente.

Bien, vamos a pasar ahora a la parte ya de descifrar y descifrar ficheros.

Para ello, bueno, el parámetro output que os permite generar, bueno, con el menos o es importante cuando ya utilicéis la herramienta varias veces, pues vais a no, antes de pasar a la parte de cifrar y descifrar, vamos a utilizar, vamos a exportar, vamos a hacer el proceso de exportar.

Aquí tenéis comando, me creo en barra tmp un fichero donde estaré exportando mi clave pública.

Ahí tenéis el mecanismo con el armor pongo para que sea legible, ahí veis codificado en base 64 ese bloque de pgp y ahí eso es lo que vosotros directamente codificado en base 64 ese bloque de pgp y ahí eso es lo que vosotros directamente podemos pegar en un correo y dárselo a alguien y esa persona ya tendría nuestra clave pública o subrogar es el fichero binario de la clave.

Se puede dar si no indicamos el armor, pues fijaros también con el comando file cómo puedo validar que es un archivo de clave pública y está exportado y luego lo podemos transferir o enviar a quien nosotros queremos o incluso lógicamente subirlo a un servidor de claves con el propio GPG.

Hay un parámetro Sendkeys que nos permitiría enviarlo a un key server que vosotros indicarais.

Bien, pues ahora sí creo que vamos a pasar a la parte de cifrado y descifrado.

Para ello, bueno, antes vamos a ver un poco todos los parámetros que tiene.

Fijaos todas las opciones que tenemos, son bastantes, aunque muchas son bastante sencillas cuando tenemos claro los conceptos de criptografía, tanto simétrico como asimétrico, tenemos ahí también un poco los comandos largos, tenemos también las abreviaturas con el guión y la letra y bueno, tanto la parte simétrica como asimétrica está contemplada.

Fijaros que hay un parámetro para hacer cifrado simétrico que lo veremos después al final de la sesión iremos trabajando sobre ello.

Cuando no indicamos qué es un cifrado simétrico siempre vamos a utilizar un cifrado asimétrico a través de clave pública y clave privada.

Visto esto vamos a irnos a la parte de cifrar, para ello vamos a crearnos un fichero a continuación vamos a crearnos un fichero que le vamos a llamar, lo vamos a llamar Secret y con eso vamos a ir trabajando los diferentes conceptos del cifrado, el cifrado, clave simétrica y con las firmas.

Bueno, vamos a abrir coordinado, creamos un fichero secret y le vamos a poner un texto, por ejemplo Hola yo soy un secreto, hola yo soy un secreto.

Guardamos el fichero y vamos a empezar a trabajar con ello.

Bien, pues me falta, no, no me falta nada, vamos a continuar.

Si es importante también que ver que en algunos casos cifrado y descifrado podemos utilizar un parámetro armor que luego veremos también que es.

Vale, pues ahí tenemos el fichero, pesa 18 bytes, fichero que solamente tiene esa frase y no tiene más fichero de texto, muy sencillo y y ahora lo que vamos a hacer es empezar a trabajar con el comando encrypt, vamos a poner armor, ahora veremos qué es recipient, que es email asociado a la clave pública que vamos a utilizar en este caso y pondremos al final el fichero que queremos cifrar, que en este caso es el fichero secret.

Cuando hagamos esta operación nos va a pedir, va a hacer la operación, se nos va a generar un fichero sig asc porque está en ascii, lo que estamos diciéndoles generame con armor, generame un fichero que sea legible pero que esté cifrado.

¿Cómo es esto?

Pues esto es como vais a ver a continuación, porque tenemos el fichero cifrado pero lo codificamos en base 64, con lo cual generamos ese bloque pgp que podríamos pegar en un correo y enviarlo y ya tendríamos un envío de un fichero cifrado pero está codificado en base 64, de manera que puede ser utilizado en protocolos de texto o en otros protocolos de texto que podemos pegar directamente y que podamos visualizar un cut de ese fichero y bueno, no entendemos lo que hay ahí pero está cifrado pero es legible porque está codificado base 64 pero es la cifra.

Entonces ahora lo que vamos a hacer es, bueno, tenemos cifrado esta información en el fichero secret asc, lo que vamos a hacer es utilizar el descifrado de la siguiente manera, ponemos el parámetro decrype, ponemos el fichero sig y le decimos o redirigimos a s, fijaos que vamos a utilizar clave privada por primera vez, por lo cual nos pide la contraseña, digo lo de primera vez porque puede ser que si lo utilizamos después no se haya pasado el tiempo este cacheado o la contraseña y no nos pide el passphrase de la clave privada, eso puede ocurrir, pero ahora mismo lo que está ocurriendo es que nos está pidiendo cuál es el passphrase de la clave privada porque vamos a descifrar un fichero con mi clave privada.

Le damos aquí a ok y obtenemos el fichero s.

Vemos que el contenido vuelve a ser el original del fichero secret, es decir, tenemos ya descifrado el fichero que hemos cifrado.

Bien, ahora vamos a utilizar otro método, otro mecanismo, otra forma de cifrar.

Realmente el cambio va a estar en que no ponemos el armo, no hacemos que sea un bloque pgp donde esté codificado en base ese cifrado, sino que lo que vamos a hacer es hacer que directamente el fichero que me devuelva ese cifrado esté un en.

¿Vamos a generar de nuevo esto, veis?

Metemos el ingred, metemos el recipient, metemos la cuenta del correo que está asociado a la clave pública e introducimos el fichero.

El resultado ahora va a ser un fichero secret gpg, es diferente, claro, vamos a ver que va a ser binario, vamos a hacer un ls, vemos el fichero gpg, está ahí y si hacemos un cut sobre ese fichero vamos a ver que pues el contenido está cifrado.

Fijaos que el secret GPG son 480 bytes, es mucho menor que 711 en el caso de las y de la base 64.

Bueno, ahí veis que eso pues no podemos pegar en un correo por ejemplo directamente porque eso no es legible.

Si hacemos con file vemos que efectivamente me está identificando que es un fichero edipo PGP.

Bien, comentarse, comentaros, comentaros.

Bueno, tenemos eso y ahora vamos a ir a la parte de firma.

Antes de pasarle a la firma, si quería enseñaros otra forma de hacer el descifrado, pues es poniendo para otro output y decirle yo quiero este este fichero con este nombre y aplico sobre el fichero secret GPG.

Eso ya automáticamente va a saber con qué clave pública se cifró y con clave privada.

Fijaros que no se pide el pin o el passphrase de la clave privada, en este caso no se ha pedido porque está cacheado en esta sesión.

Si apagáramos máquina pasaba un tiempo.

Bien, pues ahora sí vamos a vamos a pasar a la parte de firma.

Bien, pues vamos a continuar, estamos llegando al final, tenemos, bueno hemos hecho el cifrado, hemos hecho el descifrado y ahora nos falta ver la parte de firma.

En este caso vamos a generar parámetro output para obtener la firma de este fichero, del contenido de este fichero, bueno, contenido vamos a poner aquí sí y obtendremos un fichero que se llama secretchicket.

Si vamos a utilizar el parámetro verify para hacer la validación y ahí se puede ver que la firma es correcta.

Fijaros que además en la cabecera del fichero viene con quién está firmado, con qué clave y entonces utilizamos la clave pública, en este caso de ese usuario que nos han dado.

Imaginaos que el usuario firma un fichero, lo proporciona a otros usuarios y los usuarios tienen la clave pública del usuario que firmó, de modo que puede validar que el contenido está íntegro.

Y luego nos faltaría por ver la parte del cifrado simétrico con GPG.

Es bastante sencillo porque podemos decir que ficheros sí que lo queremos cifrar.

Un poco de conflicto con c o c nos permite cifrar con claves simétricas, es importante esto, nos pide el uso de la clave, nos permite no, mejor dicho introducir con qué valor por ejemplo queremos cifrar.

Después utilice algoritmos como AES, nos da esa anemilla, nos dice bueno todo igual, este fichero está filtrado pero ya se fija asimétricas y está descifrado con clave simétrica.

Si nos fijamos también aquí vamos a ver que pesa mucho menos que el gpg de la otra manera.

Ahora vamos a ver cómo descifrar.

Así que nada, con esto finalizamos y nos vemos sin la.