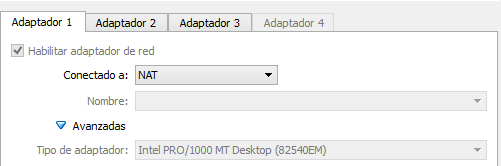


**Titulo: Uso de herramientas de criptografía**

1.- Inicie la máquina virtual Kali Linux con la interfaz de red en modo NAT



2.- Inicie una ventana de comandos como root

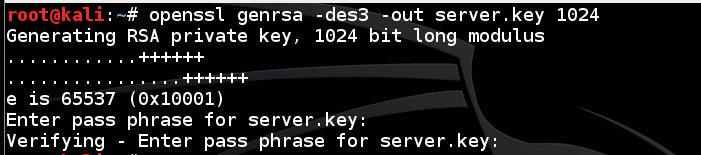
3.- Ejecute el siguiente comando para ver las opciones disponibles

#openssl help

**GENERACION DE LLAVES DE CIFRADO**

4.- Ejecute el siguiente comando para generar una llave privada

#openssl genrsa -des3 -out server.key 1024



5.- Ejecute el siguiente comando para ver el archivo creado.

#more server.key



6.- Genere un segundo archivo de clave privada con llave de 2048 bits y compárelo con el anterior.

7.- ¿Que puede decir de los tiempos demorados entre uno y caso?

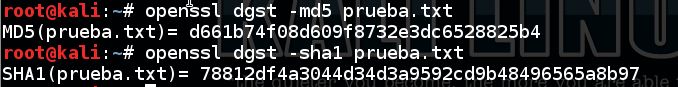
R: La key de 2048 bits tarda mucho más en leer

**GENARACION DE HASH**

8.- Cree un archivo de texto con el comando nano y ejecute los siguientes comandos:

#openssl dgst -md5 “nombre archivo”

#openssl dgst -sha1 “nombre archivo”



9.- Compare ambos resultados, ¿qué puede decir del largo de cada digest?

R: El cifrado sha-1 es más seguro que el md5

10.- Realice una modificación simple al archivo, solo cambie una letra y repita el procedimiento del punto 8

¿Qué puede concluir respecto de los digest?

R: Obtengo un digest diferente

**FIRMA DIGITAL**

11.- Cree un archivo de texto, llamado “ejemplo.txt” con el comando nano y escriba un texto cualquiera dentro del archivo.

12.- Genere una firma digital del archivo usando el siguiente comando:

#openssl dgst -sha1 -sign server.key -out signed.sha1 ejemplo.txt –aca nos pide la llave del primer server.key



13.- Visualice el archiveed.sha1” con el comando more

14.- Genera la llave pública asociada utilizando el siguiente comando:

#openssl rsa -in server.key -pubout > public.key



15.- Valide la firma digital usando el siguiente comando:

#openssl dgst -sha1 -verify public.key -signature signed.sha1 ejemplo.txt



16.- Modifique el archivo creado y valide la firma nuevamente

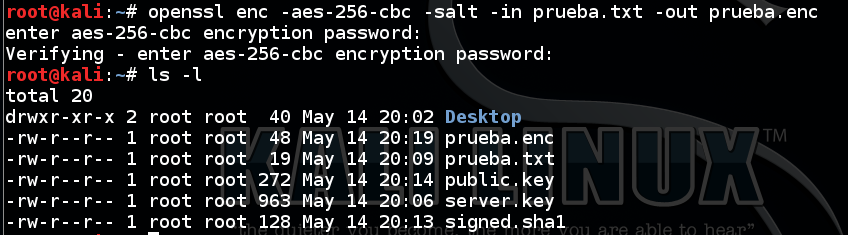
¿Qué puede comentar del resultado?

R: Al crear la firma digital si volvemos a modificar el archivo podemos ver nos da error al verificar la firma digital

**CIFRADO SIMETRICO DE UN ARCHIVO**

17.- Realice el cifrado del archivo “prueba.txt” usando el siguiente comando:

#openssl enc -aes-256-cbc -salt -in prueba.txt -out prueba.enc



18.- Visualice el archivo con el comando:

# more ejemplo.enc

¿Pudo rescatar la información del archivo?

19.- Envíe el archivo cifrado a su compañero vía correo electrónico

20.- Pida a su compañero que ejecute el siguiente comando para descifrar el archivo.

# openssl enc -d -aes-256-cbc -in ejemplo.enc

¿Qué puede concluir de esta experiencia?

**MEDICION DE RENDIMIENTO**

21.- Realice una medición del tiempo de cifrado para los algoritmos

- DES, 3DES, AES-128

con el comando:

#openssl speed "algoritmo"

realice una tabla según la siguiente estructura:

Algoritmo DES

|  |  |
| --- | --- |
| Tamaño de bloque | Cantidad de bloques cifrados |
| 16 |  |
| 64 |  |
| 256 |  |
| 1024 |  |
| 8192 |  |

Algoritmo 3DES

|  |  |
| --- | --- |
| Tamaño de bloque | Cantidad de bloques cifrados |
| 16 |  |
| 64 |  |
| 256 |  |
| 1024 |  |
| 8192 |  |

Algoritmo AES-192

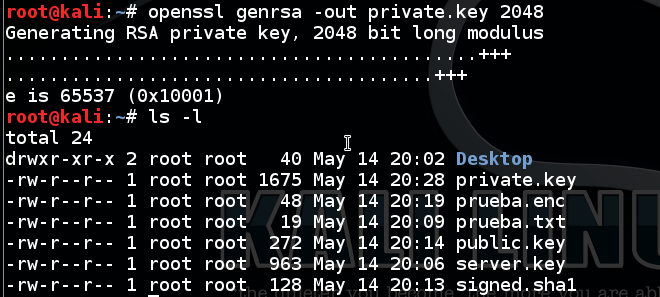
|  |  |
| --- | --- |
| Tamaño de bloque | Cantidad de bloques cifrados |
| 16 |  |
| 64 |  |
| 256 |  |
| 1024 |  |
| 8192 |  |

**CIFRADO ASIMETRICO DE UN ARCHIVO**

22.- Genere un archivo de texto con el comando nano con un mensaje guardado

23.- Genere una llave privada usando el siguiente comando:

# openssl genrsa -out private.key 2048

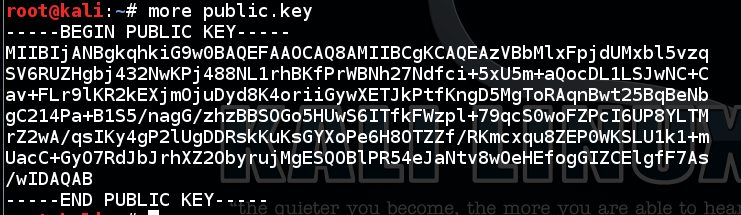


24.- Genere una llave pública usando el siguiente comando:

# openssl rsa -in private.key -pubout -out public.key



25.- Revise con el comando more los archivos generados



26.- Envíe a un compañero su archivo de llave pública vía correo electrónico

27.- Pídale que cifre su archivo de texto con el siguiente comando:

# openssl rsautl -encrypt -inkey publica.key -pubin -in "nombre archivo" -out "nombre archivo cifrado"

28.- Pídale que le envíe el archivo cifrado, cuando lo reciba cópielo a su directorio trate de visualizarlo con un editor de texto. ¿Puede ver el contenido?

29.- Descífrelo con el siguiente comando:

# openssl rsautl -decrypt -inkey privada.key -in "nombre archivo cifrado" -out "archivo de salida"

30.- Compare el archivo de salida con el mensaje original de su compañero.