

Actividad 3 - Seguridad en Redes

Profesor: René Guerrero Torres

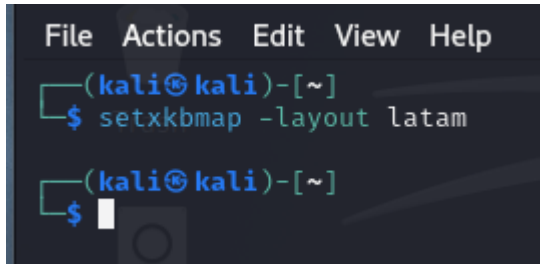
Ayudante: Iván Zuñiga

Alumno: Reinaldo Pacheco Parra

A. Cifrado Asimétrico

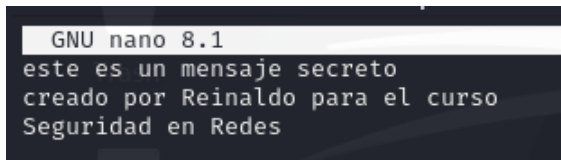
1.- Se inicia la máquina de Kali con la interfaz de red en modo NAT en modo Live

2.- Se configura el teclado en formato latinoamericano con el comando
`setxkbmap -layout latam`

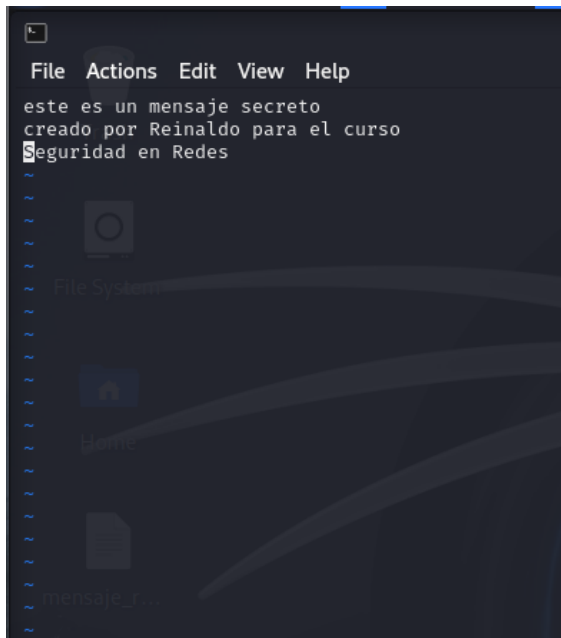


```
File Actions Edit View Help
(kali㉿kali)-[~]
$ setxkbmap -layout latam
(kali㉿kali)-[~]
$
```

3. Se crea un archivo de texto con el editor nano



```
GNU nano 8.1
este es un mensaje secreto
creado por Reinaldo para el curso
Seguridad en Redes
```



```
File Actions Edit View Help
este es un mensaje secreto
creado por Reinaldo para el curso
Seguridad en Redes
~
~
~
~ File System
~
~
~ Home
~
~
~ mensaje_r...
~
```

```
File Actions Edit View Help

(kali@kali)-[~/Desktop]
$ nano mensaje_reinaldo.txt

(kali@kali)-[~/Desktop]
$ vi mensaje_reinaldo.txt

(kali@kali)-[~/Desktop]
$ cat "mensaje_reinaldo.txt"
este es un mensaje secreto
creado por Reinaldo para el curso
Seguridad en Redes

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

4. Se genera la llave privada del usuario con el comando
openssl genrsa -out private.key 2048
Se visualiza con ls -l

```
(kali@kali)-[~/Desktop]
$ openssl genrsa -out private.key 2048

(kali@kali)-[~/Desktop]
$ ls -l
total 8
-rw-rw-r-- 1 kali kali 80 Nov 12 11:44 mensaje_reinaldo.txt
-rw----- 1 kali kali 1704 Nov 12 11:48 private.key

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

5. Se genera la llave pública para el usuario con el comando
openssl rsa -in private.key -pubout -out public.key
Se visualiza con ls -l

```
(kali@kali)-[~/Desktop]
$ openssl rsa -in private.key -pubout -out public.key
writing RSA key

(kali@kali)-[~/Desktop]
$ ls -l
total 12
-rw-rw-r-- 1 kali kali 80 Nov 12 11:44 mensaje_reinaldo.txt
-rw----- 1 kali kali 1704 Nov 12 11:48 private.key
-rw-rw-r-- 1 kali kali 451 Nov 12 11:53 public.key

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

6. Se cifra el mensaje con la llave pública del usuario con el comando
openssl pkeyutl -encrypt -inkey public.key -pubin -in mensaje_reinaldo.txt -out mensaje_reinaldo.enc
Se visualiza con ls -l

```
(kali@kali)-[~/Desktop]
$ openssl pkeyutl -encrypt -inkey public.key -pubin -in mensaje_reinaldo.txt -out mensaje_reinaldo.enc

(kali@kali)-[~/Desktop]
$ ls -l
total 16
-rw-rw-r-- 1 kali kali 256 Nov 12 11:59 mensaje_reinaldo.enc
-rw-rw-r-- 1 kali kali 80 Nov 12 11:44 mensaje_reinaldo.txt
-rw-rw-r-- 1 kali kali 1704 Nov 12 11:48 private.key
-rw-rw-r-- 1 kali kali 451 Nov 12 11:53 public.key

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

7. Se descifra el archivo usando la llave pública del usuario con el comando
openssl pkeyutl -decrypt -inkey private.key -in mensaje_reinaldo.enc -out mensaje2_reinaldo.txt
Se visualiza con ls -l

8. Se realiza la comparación entre ambos archivos con el comando
cmp mensaje_reinaldo.txt mensaje2_reinaldo.txt
xxd mensaje_reinaldo.txt
xxd mensaje2_reinaldo.txt

```
(kali@kali)-[~/Desktop]
$ cmp mensaje_reinaldo.txt mensaje2_reinaldo.txt

(kali@kali)-[~/Desktop]
$ xxd mensaje_reinaldo.txt
00000000: 6573 7465 2065 7320 756e 206d 656e 7361  este es un mensa
00000010: 6a65 2073 6563 7265 746f 0a63 7265 6164  je secreto.cread
00000020: 6f20 706f 7220 5265 696e 616c 646f 2070  o por Reinaldo p
00000030: 6172 6120 656c 2063 7572 736f 0a53 6567  ara el curso.Seg
00000040: 7572 6964 6164 2065 6e20 5265 6465 730a  uridad en Redes.

(kali@kali)-[~/Desktop]
$ xxd mensaje2_reinaldo.txt
00000000: 6573 7465 2065 7320 756e 206d 656e 7361  este es un mensa
00000010: 6a65 2073 6563 7265 746f 0a63 7265 6164  je secreto.cread
00000020: 6f20 706f 7220 5265 696e 616c 646f 2070  o por Reinaldo p
00000030: 6172 6120 656c 2063 7572 736f 0a53 6567  ara el curso.Seg
00000040: 7572 6964 6164 2065 6e20 5265 6465 730a  uridad en Redes.

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

Se puede ver que ambos archivos son iguales

9. Se genera un par de llaves públicas y privadas de 2048 bits usando la herramienta:
<https://www.devglan.com/online-tools/rsa-encryption-decryption>

Generate RSA Key Pair Online

Select RSA Key Size ?

2048 bit

Generate RSA Key Pair

Public Key(X.509 Format)

```
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAu0sHmgsgK
RaPKZsy8MOJ5W7FUDYRkq1hovYGp/83yv/lv0Zk3qmWXUwGV1Nb9
mrZ2yAsa5H9bOJYluxkN3nrLYwD+DXklAbmAGeOUDBH2BEfNDQuy
mhtNT7vgLlc0pDeAFK7hfQzTQq1j5487X/WS/6MVZsBFeZDNnp/hl6
Knj/DEdxC7smhl5+bLapGEcedoBNbgf08I7gNz6pG0Ebh9M9dkM54V
kA+8fUZltJ2p9SLioe78CP0Skzw10oL3GS5XGCUWJaMQXkzb742R1e
Yws4ZXThhKEBfw+1/CmCXDsmfB2lGwF0APi0keP2oGemvTmmiCx
-----END PUBLIC KEY-----
```

Download Public Key

Private Key(PKCS8 Format)

```
-----BEGIN RSA PRIVATE KEY-----
MIIEvQIBADANBgkqhkiG9w0BAQEFAASCBKcwggSjAgEAAoIBAQC7S
weaCyApFo8pmzLww4nlbsVR1hGSrWGi9gan/zfK/+W/RmTeqZZdTA
ZXU1v2atnblCxrkf1s4lgi7GQ3eestjAP4NeSUBuYAZ45QMEfYER80NC
7KaG01Pu+AuVzSkN4AUruF9DNNCrWPNjztF9ZL/oxVmwEV5kM2en+
GXoqeP8MR3ELuyaEjn5stqkYRx52gE1uB/Twjua3PqkbQRuH0z12Qz
-----END RSA PRIVATE KEY-----
```

Download Private Key

10. Se copia la llave pública y se cifra el mensaje

RSA Encryption

Enter Plain Text to Encrypt ?

Este es un mensaje que será cifrado

Enter Public/Private key ?

```
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAu0sHmgsgK
RaPKZsy8MOJ5W7FUDYRkq1hovYGp/83yv/lv0Zk3qmWXUwGV1Nb9
mrZ2yAsa5H9bOJYluxkN3nrLYwD+DXklAbmAGeOUDBH2BEfNDQuy
mhtNT7vgLlc0pDeAFK7hfQzTQq1j5487X/WS/6MVZsBFeZDNnp/hl6
Knj/DEdxC7smhl5+bLapGEcedoBNbgf08I7gNz6pG0Ebh9M9dkM54V
-----END PUBLIC KEY-----
```

RSA Key Type: ? ☒ Public key ☐ Private Key

Select Encryption Algorithm ?

RSA/ECB/PKCS1Padding

Encrypt

Encrypted Output (Base64):

RSA Decryption

Enter Encrypted Text to Decrypt (Base64) ?

Enter Encrypted Text to Decrypt

Enter Public/Private key ?

```
-----BEGIN RSA PRIVATE KEY-----
MIIEvQIBADANBgkqhkiG9w0BAQEFAASCBKcwggSjAgEAAoIBAQC7S
weaCyApFo8pmzLww4nlbsVR1hGSrWGi9gan/zfK/+W/RmTeqZZdTA
ZXU1v2atnblCxrkf1s4lgi7GQ3eestjAP4NeSUBuYAZ45QMEfYER80NC
7KaG01Pu+AuVzSkN4AUruF9DNNCrWPNjztF9ZL/oxVmwEV5kM2en+
GXoqeP8MR3ELuyaEjn5stqkYRx52gE1uB/Twjua3PqkbQRuH0z12Qz
-----END RSA PRIVATE KEY-----
```

RSA Key Type: ? ☐ Public key ☒ Private Key

Select Decryption Algorithm ?

RSA/ECB/PKCS1Padding

Decrypt

Decrypted Output:

Se obtiene el mensaje cifrado como output

Encrypted Output (Base64):

```
A0J/U5E+dDDen1e3sAPhpG0h2Mvr4rfGPhAIDZcuuXQq6F3tGCeFe6
wulfiVyhOKvBvpcUKyF30DxX8rbVejopeyo2MYIjw82bzurDPKzGj79zE
xJlZnoldPOBij2zxrYtgCdwj753488O/b60JogAEWJ2e48Db0zy2t36j/
MJBwtuN2HH4jn43+tkcPO60JKlu91ndVqh/thGAsNpEOsAXvY/FTR3
hKxj497JevkEpLgm5T3SjE2KelDA1bFpXFm5y3MoM/wfDz2A7wMZe
```

11. Se copia la llave privada y se descifra el mensaje que se escribió anteriormente obteniendo como output el mensaje inicial descifrado

RSA Decryption

Enter Encrypted Text to Decrypt (Base64) ?

```
OKvBvpcUKyff30DxX8rbVejopeyo2MYljw82bzurDPKzGj79zExJlzZnoldPOBij2
zxrYtgCdwj7534880/b60JogAEWJ2e48Db0zy2t36j/MJBwtuN2HH4jn43+tkcP
O60JKlu91ndVqh/thGAsNeOsAXvY/FTR3hxxj497JevkEpLGm5T3SjE2KelDA
1bFpXFm5y3MoM/wfDz2A7wMzEywQMU7Mwhf+Zyel/1CKbsMG83+CKvCYw
```

Enter Public/Private key ?

```
ECKDfWk3DcInLjpsjQb0Tjdu0vYJrgwmdIusezyQcYRGbncynRZCK6wedECQBN
4at8iWo/c0KAic+S0id6wz4atH3YTdu2c2wWWc+TFecpuVtBdZ1rGY3KKMmQ
KBgQDzrweQXnW3NU113YnfqZpm9fuyM60m+ybCm9q/D6RAv3+7t9f7j7dd7
S61Oe50GRBjn7xj3Oq/tGnPyfzyj8kTGgiYtnDpfRWXVot9eY/1Jok7WmpfxlWo
blmXT5GCodYlQ1RpLKOfE2D767cluOhPAWpNYfQN2UD/AA4DGDqKwwKBgC
GejLslXsQUAM4IFrUwZyrlbbtQHxSE2djXOTiLOWwyOI1qAsGFcHVwSEh65HQc
ZAVt11wYTwCcaTlvt3ld32fdTt/n7ZChMv+BiRdRWvGBELteMDckeHQ05LlW67
```

RSA Key Type: ? ☐ Public key ☒ Private Key

Select Decryption Algorithm ?

RSA/ECB/PKCS1Padding

Decrypt

Decrypted Output:

Este es un mensaje que será cifrado

4. Se generan las llaves privadas para los usuarios "private1.key" "private2.key" una para el usuario 1 y otra para el usuario 2 con el comando
openssl genpkey -paramfile dhp.pem -out private1.key
openssl genpkey -paramfile dhp.pem -out private2.key

```
(kali㉿kali)-[~/Desktop/algoritmo_dh]
$ openssl genpkey -paramfile dhp.pem -out private1.key

(kali㉿kali)-[~/Desktop/algoritmo_dh]
$ openssl genpkey -paramfile dhp.pem -out private2.key

(kali㉿kali)-[~/Desktop/algoritmo_dh]
$ ls -l
total 12
-rw-rw-r-- 1 kali kali 428 Nov 12 12:47 dhp.pem
-rw-rw-r-- 1 kali kali 806 Nov 12 12:55 private1.key
-rw-rw-r-- 1 kali kali 806 Nov 12 12:56 private2.key
```

Se visualizan las llaves generadas con `ls -l`

5. Se generan las llaves públicas para los usuarios "public1.key" y "public2.key" una para el usuario 1 y otra para el usuario 2 con el comando
openssl pkey -in private1.key -pubout -out public1.key
openssl pkey -in private2.key -pubout -out public2.key

Se visualiza la creación de las llaves con `ls -l`

```
(kali㉿kali)-[~/Desktop/algoritmo_dh]
$ openssl pkey -in private1.key -pubout -out public1.key

(kali㉿kali)-[~/Desktop/algoritmo_dh]
$ openssl pkey -in private2.key -pubout -out public2.key

(kali㉿kali)-[~/Desktop/algoritmo_dh]
$ ls -l
total 20
-rw-rw-r-- 1 kali kali 428 Nov 12 12:47 dhp.pem
-rw-rw-r-- 1 kali kali 806 Nov 12 12:55 private1.key
-rw-rw-r-- 1 kali kali 806 Nov 12 12:56 private2.key
-rw-rw-r-- 1 kali kali 800 Nov 12 13:02 public1.key
-rw-rw-r-- 1 kali kali 804 Nov 12 13:02 public2.key

(kali㉿kali)-[~/Desktop/algoritmo_dh]
$
```

6. Se genera el secreto compartido para ambos usuarios con el comando
openssl pkeyutl -derive -inkey private1.key -peerkey public2.key -out secret1reinaldo.bin


```
openssl pkeyutl -derive -inkey private2.key -peerkey  
public1.key -out secret2reinaldo.bin
```

Se visualiza el resultado obtenido con `ls -l`

```
(kali㉿kali)-[~/Desktop/algoritmo_dh]
$ openssl pkeyutl -derive -inkey private2.key -peerkey public1.key -out secret2reinaldo.bin

(kali㉿kali)-[~/Desktop/algoritmo_dh]
$ ls -l
total 28
-rw-rw-r-- 1 kali kali 428 Nov 12 12:47 dhp.pem
-rw-rw-r-- 1 kali kali 806 Nov 12 12:55 private1.key
-rw-rw-r-- 1 kali kali 806 Nov 12 12:56 private2.key
-rw-rw-r-- 1 kali kali 800 Nov 12 13:02 public1.key
-rw-rw-r-- 1 kali kali 804 Nov 12 13:02 public2.key
-rw-rw-r-- 1 kali kali 256 Nov 12 13:07 secret1reinaldo.bin
-rw-rw-r-- 1 kali kali 256 Nov 12 13:08 secret2reinaldo.bin
```

7. Se compara si ambos archivos son iguales con el comando
`cmp secret1reinaldo.bin secret2reinaldo.bin`
8. Se verifica que el contenido de ambos archivos sea el mismo, lo cual es correcto

```
(kali㉿kali)-[~/Desktop/algoritmo_dh]
$ xxd secret1reinaldo.bin
00000000: 58eb db98 5038 1ed6 40ad 6c18 799c 5344  X ... P8 ..@.l.y.SD
00000010: 8b70 17dd f065 8fdb f626 fd92 4808 2c66  .p ... e ... 6 ... H.,f
00000020: ce66 d20a 6879 b019 0a41 276f ad92 9966  .f..hy...A'o...f
00000030: 87ad 50a6 e523 e764 377a d5ee 5bd6 ad56  ..P...#.d7z...[...V
00000040: 7f61 90c6 e60e 82e2 5c10 d52a e2b7 1069  .a.....\..*...i
00000050: ce47 28b7 60e0 1800 1aee b5f6 c297 1dcf  .G(.`.....
00000060: fc3d 7b68 2906 59ba 6753 94f9 d834 b723  .={h).Y.gS...4.#
00000070: 044e 0c63 82de b7d7 897a 4b0c 250d dde2  .N.c.....zK.%...
00000080: dd8e 29eb 89d1 700b fcc2 4920 c8d0 160f  ..) ...p...I ....
00000090: b7dd bbab dcce a260 12c1 acbd f145 88a7  ....E...
000000a0: 5223 0af0 537a c0ba fa6b 7629 54a3 3389  R#..Sz...kv)T.3.
000000b0: 3f24 5c13 c53c e6a4 caba 1b2f bc39 e864  ?$\...<...../.9.d
000000c0: c88c 881c 0891 f4b8 6d9c f410 aad2 5641  ....m....VA
000000d0: 135f 8dcf 9bbe a47f 3ad6 be7b 3382 86f4  ._. ....:..{3...
000000e0: 6403 fd5f d180 847d c16a ffe6 0461 8cdf  d.._...}.j..a..
000000f0: 05cd 020e a84e 36e0 8844 f179 8466 3725  ....N6..D.y.f7%


(kali㉿kali)-[~/Desktop/algoritmo_dh]
$ xxd secret2reinaldo.bin
00000000: 58eb db98 5038 1ed6 40ad 6c18 799c 5344  X ... P8 ..@.l.y.SD
00000010: 8b70 17dd f065 8fdb f626 fd92 4808 2c66  .p ... e ... 6 ... H.,f
00000020: ce66 d20a 6879 b019 0a41 276f ad92 9966  .f..hy...A'o...f
00000030: 87ad 50a6 e523 e764 377a d5ee 5bd6 ad56  ..P...#.d7z...[...V
00000040: 7f61 90c6 e60e 82e2 5c10 d52a e2b7 1069  .a.....\..*...i
00000050: ce47 28b7 60e0 1800 1aee b5f6 c297 1dcf  .G(.`.....
00000060: fc3d 7b68 2906 59ba 6753 94f9 d834 b723  .={h).Y.gS...4.#
00000070: 044e 0c63 82de b7d7 897a 4b0c 250d dde2  .N.c.....zK.%...
00000080: dd8e 29eb 89d1 700b fcc2 4920 c8d0 160f  ..) ...p...I ....
00000090: b7dd bbab dcce a260 12c1 acbd f145 88a7  ....E...
000000a0: 5223 0af0 537a c0ba fa6b 7629 54a3 3389  R#..Sz...kv)T.3.
000000b0: 3f24 5c13 c53c e6a4 caba 1b2f bc39 e864  ?$\...<...../.9.d
000000c0: c88c 881c 0891 f4b8 6d9c f410 aad2 5641  ....m....VA
000000d0: 135f 8dcf 9bbe a47f 3ad6 be7b 3382 86f4  ._. ....:..{3...
000000e0: 6403 fd5f d180 847d c16a ffe6 0461 8cdf  d.._...}.j..a..
000000f0: 05cd 020e a84e 36e0 8844 f179 8466 3725  ....N6..D.y.f7%
```


9. Se generan las llaves públicas y privadas de un usuario usando el recurso <https://cryptotools.net/dhe>

You

Private key

ZioIpnbwaTwQ6cnqzr5qOrk5R6I0/QhYbIZ2ZyZztgA=

 Generate

Public key

DizCY7tsf1Youl96bXcISBG1rNwjDfYCWYKSZw04+nA=

Public key: ZioIpnbwaTwQ6cnqzr5qOrk5R6I0/QhYbIZ2ZyZztgA=


Private key: DizCY7tsf1Youl96bXcISBG1rNwjDfYCWYKSZw04+nA=

10. Se generan otra llaves públicas y privadas para otro usuario

You

Private key

SDmp7K/g86vE+ae1MEgzVdKroqT/AMf8z69fmo98kgA=

 Generate

Public key

CD5lJl0ST+zuujimXuxwY/e1Kye7RR/Q0wuVfYB4rkg=

Public key: SDmp7K/g86vE+ae1MEgzVdKroqT/AMf8z69fmo98kgA=

Private key: CD5lJl0ST+zuujimXuxwY/e1Kye7RR/Q0wuVfYB4rkg=


11. Se copian las llaves públicas de cada usuario y se entregan al otro

Usuario 1 entrega su llave pública a Usuario 2:

You

Private key

SDmp7K/g86vE+ae1MEgzVdKroqT/AMf8z69fmo98kgA=

 Generate

Public key

CD51J10ST+zuujimXuxwY/e1Kye7RR/Q0wuVFYB4rkg=

Partner

Private key

Public key

DizCY7tsf1Youl96bXcISBG1rNwjDFYCWYKSZw04+nA=

Shared secret


E+AZDbylC4Zo6FOApKQ0GtK6JKRYj9HG8mZ9VwXQfwg=

Usuario 2 entrega su llave pública al Usuario 1:

You

Private key

ZioIpnbwaTwQ6cnqzr5qOrk5R6I0/QhYbIZ2ZyZztgA=

 Generate

Public key

DizCY7tsf1Youl96bXcISBG1rNwjDFYCWYKSZw04+nA=

Partner

Private key

Public key

CD51J10ST+zuujimXuxwY/e1Kye7RR/Q0wuVFYB4rkg=

Shared secret

E+AZDbylC4Zo6FOApKQ0GtK6JKRYj9HG8mZ9VwXQfwg=

Se comparan las llaves secretas de cada usuario:

Llave secreta usuario 1:

You

Private key

ZioIpnbwaTwQ6cnqzr5qOrk5R6I0/QhYbIZ2ZyZztgA=

Generate

Public key

DizCY7tsf1You196bXcISBG1rNwjDfYCWYKSZw04+nA=

Partner

Private key

Public key

CD51J10ST+zuujimXuxwY/e1Kye7RR/QOwuVfYB4rkg=

Shared secret

E+AZDby1C4Zo6FOApKQ0GtK6JKRYj9HG8mZ9VWXQfwg=

Llave secreta usuario 2:

You

Private key

SDmp7K/g86vE+ae1MEgzVdKroqT/AMf8z69fmo98kgA=

Generate

Public key

CD51J10ST+zuujimXuxwY/e1Kye7RR/QOwuVfYB4rkg=

Partner

Private key

Public key

DizCY7tsf1You196bXcISBG1rNwjDfYCWYKSZw04+nA=

Shared secret

E+AZDby1C4Zo6FOApKQ0GtK6JKRYj9HG8mZ9VWXQfwg=

Ambos usuarios tienen la misma llave secreta.

C. Firma Digital con RSA

1. Se crea una carpeta y se accede con el comando
mkdir algoritmo_rsa
cd algoritmo_rsa

Luego, se crea con vi un archivo de texto con el comando
vi mensaje_rsa_reinaldo.txt


```
(kali㉿kali)-[~/algoritmo_rsa]
$ openssl rsa -in private.key -pubout -out public.key
writing RSA key

(kali㉿kali)-[~/algoritmo_rsa]
$ ls -l
total 12
-rw-rw-r-- 1 kali kali 84 Nov 12 13:53 mensaje_rsa_reinaldo.txt
-rw-rw-r-- 1 kali kali 1704 Nov 12 13:57 private.key
-rw-rw-r-- 1 kali kali 451 Nov 12 13:59 public.key
```

4. Se genera una firma digital para el usuario con el comando
openssl dgst -sha1 -sign private.key -out signed.sha1
mensaje_rsa_reinaldo.txt

Se visualiza con ls -l

```
(kali㉿kali)-[~/algoritmo_rsa]
$ openssl dgst -sha1 -sign private.key -out signed.sha1 mensaje_rsa_reinaldo.txt
mensaje_rsa_reinaldo.txt
(kali㉿kali)-[~/algoritmo_rsa]
$ ls -l
total 16
-rw-rw-r-- 1 kali kali 84 Nov 12 13:53 mensaje_rsa_reinaldo.txt
-rw-rw-r-- 1 kali kali 1704 Nov 12 13:57 private.key
-rw-rw-r-- 1 kali kali 451 Nov 12 13:59 public.key
-rw-rw-r-- 1 kali kali 256 Nov 12 14:02 signed.sha1
```

5. Se valida la firma digital generada con el comando
openssl dgst -sha1 -verify public.key -signature signed.sha1
mensaje_rsa_reinaldo.txt

```
(kali㉿kali)-[~/algoritmo_rsa]
$ openssl dgst -sha1 -verify public.key -signature signed.sha1 mensaje_rsa_reinaldo.txt
Verified OK
```

6. Se realiza una modificación al documento firmado con el comando
vi mensaje_rsa_reinaldo.txt

```
(kali㉿kali)-[~/algoritmo_rsa]
$ vi mensaje_rsa_reinaldo.txt
```

Se visualiza la modificación con
cat mensaje_rsa_reinaldo.txt

```
system: private key generated
File Actions Edit View Help
Este es un mensaje modificado por Reinaldo
para una firma digital con el
algoritmo RSA
~
~ Home public key algoritmo_dh
~
```

7. Se realiza la validación de la firma digital con el comando:
- ```
openssl dgst -sha1 -verify public.key -signature signed.sha1 mensaje_rsa_reinaldo.txt
```

```
(kali@kali)-[~/algoritmo_rsa]
$ openssl dgst -sha1 -verify public.key -signature signed.sha1 mensaje_rsa_reinaldo.txt
Verification failure
40873AA54C7F0000:error:02000068:rsa routines:ossl_rsa_verify:bad signature:../crypto/rsa/rsa_sign.c:426:
40873AA54C7F0000:error:1C880004:Provider routines:rsa_verify:RSA lib:../providers/implementations/signature/rsa_sig
801:
```

La validación falla debido a que el mensaje inicial fue modificado.

## D. SSL/TTL

1. Utilizamos el enlace: <https://www.ssllabs.com/ssltest/>

### SSL Server Test

This free online service performs a deep analysis of the configuration of any SSL web server on the public Internet. **Please note that the information you submit here is used only to provide you the service. We don't use the domain names or the test results, and we never will.**

Hostname:

☐ Do not show the results on the boards

2. Se ingresa una dirección web para realizar un análisis, la dirección es: demo.testfire.net

3. Se obtienen los resultados de demo.testfire.net

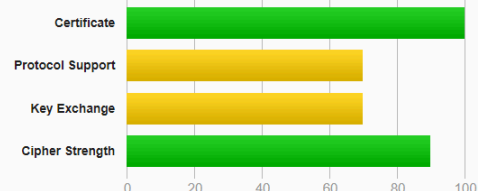
### SSL Report: demo.testfire.net (65.61.137.117)

Assessed on: Tue, 12 Nov 2024 18:32:45 UTC | [Clear cache](#)

[Scan Another »](#)

#### Summary

Overall Rating




Visit our [documentation page](#) for more information, configuration guides, and books. Known issues are documented [here](#).

This server supports weak Diffie-Hellman (DH) key exchange parameters. Grade capped to B. [MORE INFO »](#)

This server supports TLS 1.0 and TLS 1.1. Grade capped to B. [MORE INFO »](#)

4. Los protocolos aceptados por la página, son:

| Configuration                                                                     |           |     |
|-----------------------------------------------------------------------------------|-----------|-----|
|  | Protocols |     |
|                                                                                   | TLS 1.3   | No  |
|                                                                                   | TLS 1.2   | Yes |
|                                                                                   | TLS 1.1   | Yes |
|                                                                                   | TLS 1.0   | Yes |
|                                                                                   | SSL 3     | No  |
|                                                                                   | SSL 2     | No  |

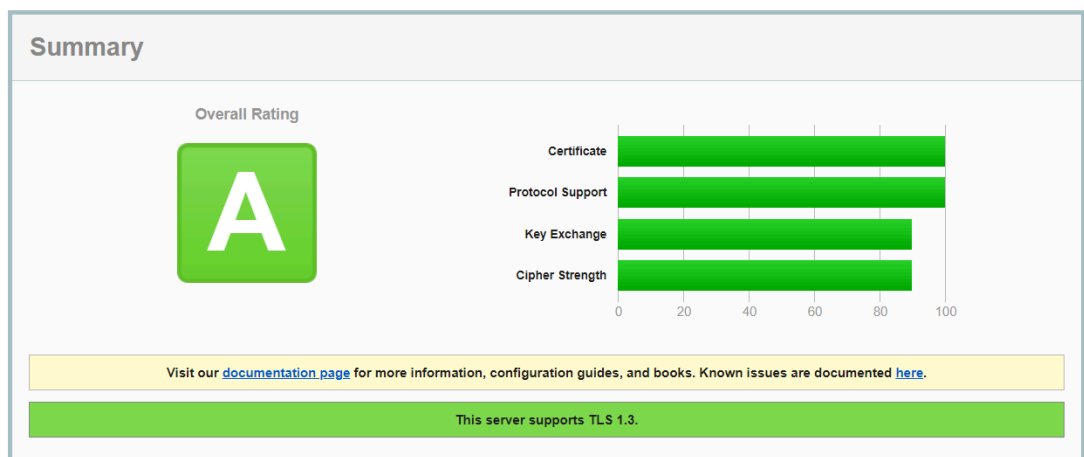
5. Se repite la operación con el link de otras páginas, por ejemplo:

<https://www.usachatiende.cl/>


### SSL Report: [www.usachatiende.cl](https://www.usachatiende.cl/) (143.244.158.179)

Assessed on: Tue, 12 Nov 2024 19:54:29 UTC | [Hide](#) | [Clear cache](#)

[Scan Another »](#)

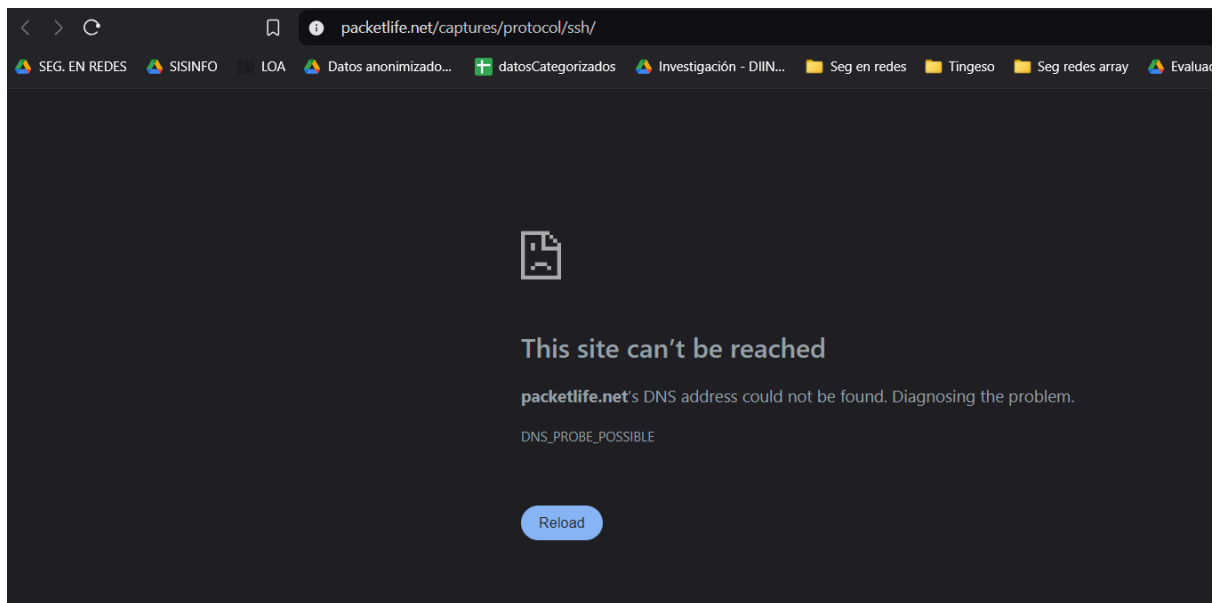


Los protocolos aceptados por la página son TLS 1.2 y TLS 1.3

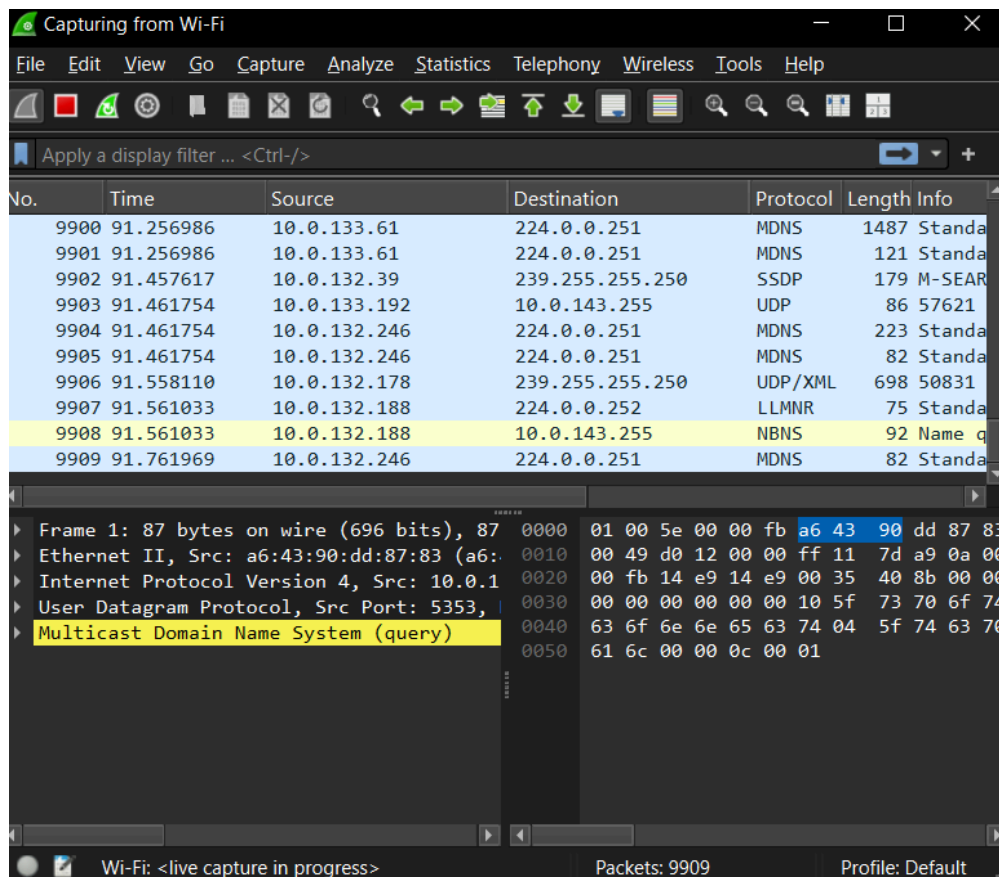
| Configuration                                                                       |           |     |
|-------------------------------------------------------------------------------------|-----------|-----|
|  | Protocols |     |
|                                                                                     | TLS 1.3   | Yes |
|                                                                                     | TLS 1.2   | Yes |
|                                                                                     | TLS 1.1   | No  |
|                                                                                     | TLS 1.0   | No  |
|                                                                                     | SSL 3     | No  |
|                                                                                     | SSL 2     | No  |

6. No se pudo realizar la actividad <https://packetlife.net/captures/protocol/ssh> debido a que la página estaba caída

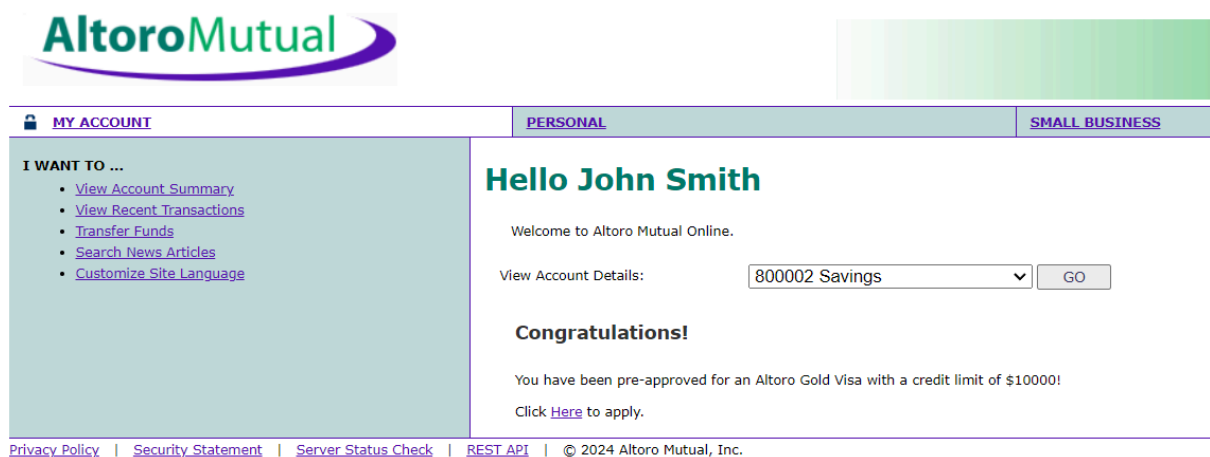




7. Se ingresa al enlace <http://altoromutual.com:8080/login.jsp> y, con la ayuda de la herramienta Wireshark, se verificará si las credenciales de ingreso se muestran en texto claro. Antes de ingresar las credenciales, se inicia Wireshark y se selecciona la interfaz Wi-Fi para comenzar la captura de tráfico.



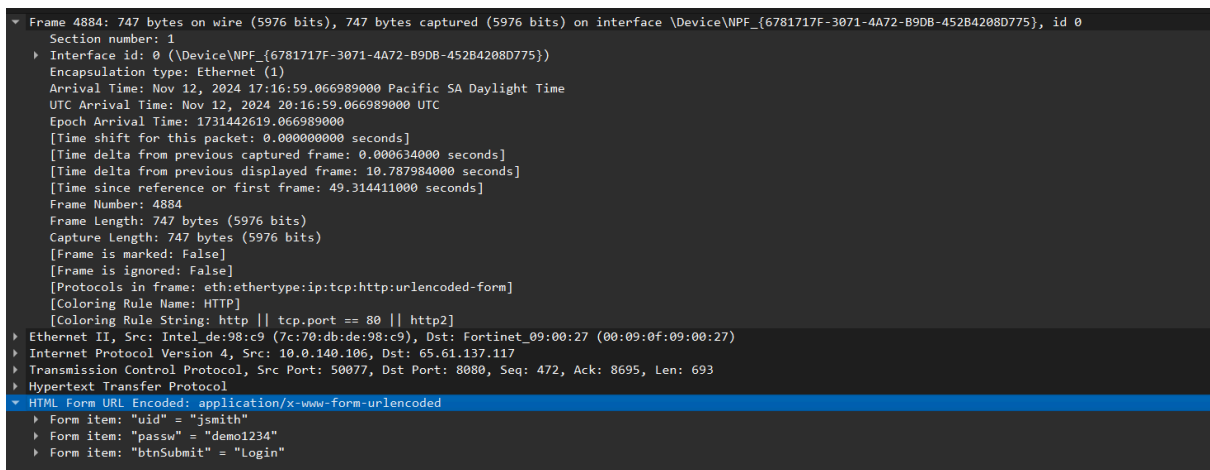
Luego, se ingresan las siguientes credenciales en el formulario  
username: jsmith  
password: demo1234



Una vez ingresadas las credenciales en la página, se procede a buscar en Wireshark el paquete de tipo POST correspondiente al inicio de sesión. En este caso, se identifica el paquete como  
65.61.137.117 HTTP 747 POST /doLogin HTTP/1.1  
(application/x-www-form-urlencoded)

|               |           |                                                                |
|---------------|-----------|----------------------------------------------------------------|
| 10.0.140.106  | HTTP      | 1434 Continuation                                              |
| 10.0.140.106  | HTTP      | 1434 Continuation                                              |
| 10.0.140.106  | HTTP      | 468 Continuation                                               |
| 65.61.137.117 | HTTP      | 747 POST /doLogin HTTP/1.1 (application/x-www-form-urlencoded) |
| 10.0.140.106  | HTTP      | 343 HTTP/1.1 302 Found                                         |
| 65.61.137.117 | HTTP      | 739 GET /bank/main.jsp HTTP/1.1                                |
| 10.0.140.106  | HTTP      | 792 HTTP/1.1 200 OK (text/html)                                |
| 10.0.129.149  | HTTP      | 237 GET /ssdp/device-desc.xml HTTP/1.1                         |
| 10.0.140.106  | HTTP/X... | 1244 HTTP/1.1 200 OK                                           |
| 10.0.129.217  | HTTP      | 237 GET /ssdp/device-desc.xml HTTP/1.1                         |
| 10.0.140.106  | HTTP/X... | 1244 HTTP/1.1 200 OK                                           |

Luego, se examina el contenido del paquete para localizar la sección “HTML Form URL Encoded” en donde se pueden visualizar los valores enviados en el formulario..



Los valores capturados corresponden a los ingresados en la página

