HACKING ÉTICO

Unidad 3. Actividad 1

13 DE MARZO DE 2024

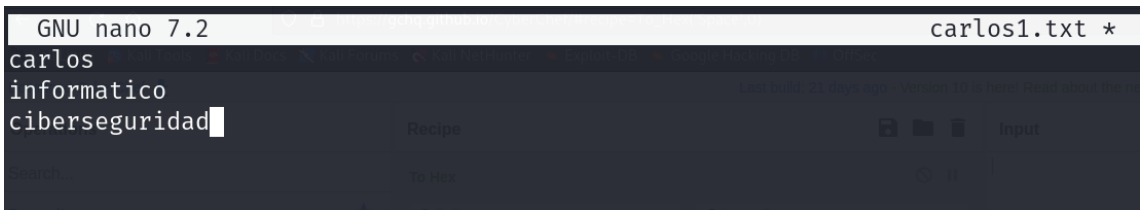
CARLOS DÍAZ MONTES
ESPECIALIZACIÓN DE CIBERSEGURIDAD

Índice

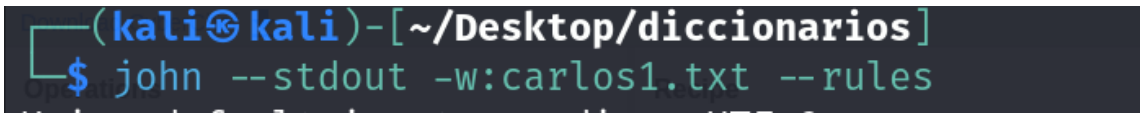
Crea tus propios diccionarios

Ejercicio 1: Crea tu primer diccionario

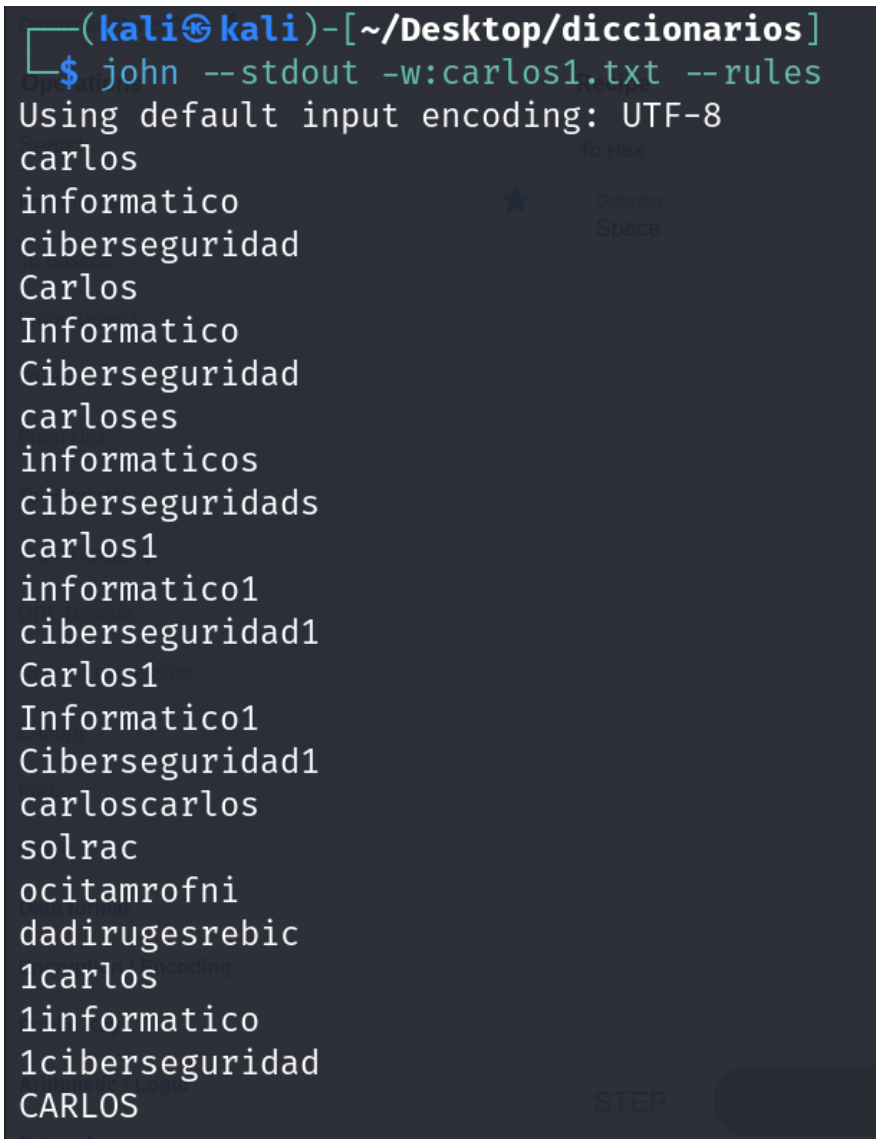
Primero me he creado el archivo carlos1.txt, y he introducido en el 3 reglas:



Ahora ejecutamos el comando:



El resultado:



Ejercicio 2: Busca una regla concreta …

La regla es :

```
  GNU nano 7.2                                  /etc/john/john.conf

[List.Rules:KoreLogicRulesPrepend4NumAppendSpecial]
cA0"[0-9][0-9][0-9][0-9]"Az"[!$@#%.]"
cA0"[0-9][0-9][0-9][0-9]"Az"[^&()_+\-={}|\[\]\\;':,/\<\>?`~*]"
A0"[0-9][0-9][0-9][0-9]"Az"[!$@#%.]"
A0"[0-9][0-9][0-9][0-9]"Az"[^&()_+\-={}|\[\]\\;':,/\<\>?`~*]"

[List.Rules:KoreLogicRulesAppend4NumSpecial]
cAz"[0-9][0-9][0-9][0-9][!$@#%.]"
cAz"[0-9][0-9][0-9][0-9][^&()_+\-={}|\[\]\\;':,/\<\>?`~*]"
Az"[0-9][0-9][0-9][0-9][!$@#%.]"
Az"[0-9][0-9][0-9][0-9][^&()_+\-={}|\[\]\\;':,/\<\>?`~*]"

[List.Rules:KoreLogicRulesAppend3NumSpecial]
cAz"[0-9][0-9][0-9][!$@#%.]"
cAz"[0-9][0-9][0-9][^&()_+\-={}|\[\]\\;':,/\<\>?`~*]"
Az"[0-9][0-9][0-9][!$@#%.]"
Az"[0-9][0-9][0-9][^&()_+\-={}|\[\]\\;':,/\<\>?`~*]"
```

El comando usado es(solo he puesto mi nombre pq sino iba a salir demasiado texto):

```
┌──(kali㉿kali)-[~/Desktop/diccionarios]
└─$ john --stdout -w:carlos1.txt --rules:KoreLogicRulesAppend3NumSpecial
Using default input encoding: UTF-8
Carlos000!
Carlos000$
Carlos000@
Carlos000#
Carlos000%
Carlos000.
Carlos001!
Carlos001$
Carlos001@
Carlos001#
Carlos001%
Carlos001.
Carlos002!
Carlos002$
Carlos002@
Carlos002#
Carlos002%
Carlos002.
Carlos003!
Carlos003$
Carlos003@
Carlos003#
Carlos003%
```

Ejercicio 3: Regla personalizada …

Nos creamos las reglas:

```
##############################################
#########   Mis reglas    ##################
##############################################

# Para concatenar tst al final de la palabra
[List.Rules:CarlosAppendtst]
cAZ"[tT][sS][tT]"
# otra para el principio de la palabra
[List.Rules:CarlosPrependtst]
A0"[tT][sS][tT]"
```

La ejecutamos y comprobamos que funciona:

```
┌──(kali㉿kali)-[~/Desktop/diccionarios]
└─$ sudo john -stdout -w:diccionario.txt --rules:CarlosPrependtst,CarlosAppendtst

Using default input encoding: UTF-8
tstseiya
tsthyoga
tsTseiya
tsThyoga
tStseiya
tSthyoga
tSTseiya
tSThyoga
Tstseiya
Tsthyoga
TsTseiya
TsThyoga
TStseiya
TSthyoga
TSTseiya
TSThyoga
Seiyatst
Hyogatst
SeiyatsT
HyogatsT
SeiyatSt
HyogatSt
SeiyatST
HyogatST
SeiyaTst
```

Ejercicio 4: Crunch – palabras de 5 dígitos

El comando usado:

```
┌──(kali㉿kali)-[~/Desktop/diccionarios]
└─$ crunch 5 5 -o carloscruch.txt
Crunch will now generate the following amount of data: 71288256 bytes
67 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 11881376

crunch: 100% completed generating output
```

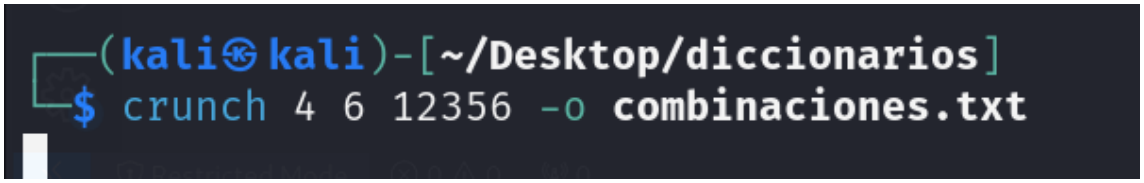
Me ha generado 11881376 contraseñas.

Lo que me genera:

```
amttt
amttu
amttv
amttw
amttx
amtty
amttz
amtua
amtub
amtuc
amtud
amtue
amtuf
amtug
amtuh
amtui
amtuj
amtuk
amtul
amtum
amtun
amtuo
amtup
amtuq
amtur
amtus
amtut
amtuu
amtuv
amtuw
```

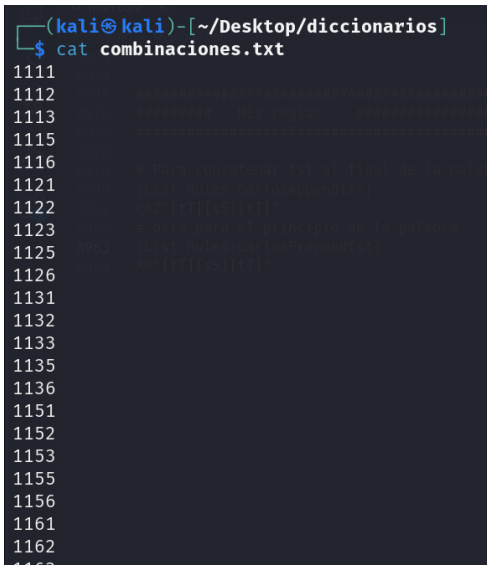
Ejercicio 5: Crunch – conjunto de caracteres

En este tengo una duda y no sabia si el pin que tiene es de 4 digitos o de 6 digitos. Como no estaba seguro he puesto este comando:



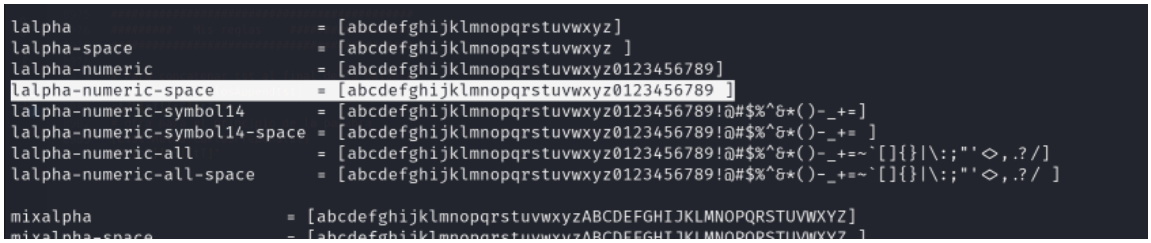
Este te crea combinaciones con esos numero de hasta 4 caracteres como mínimo y máximo como 6, en caso de ser solo de 4 caracteres cambias el 6 por un 4.

El resultado de las combinaciones:



Ejercicio 6: Crunch – Usando charsets

Investigando en /usr/share/crunch/charset.lst he visto que hay una forma de poner minúsculas, números y espacios en blanco:



El comando usado es:

```
┌──(kali㉿kali)-[~/Desktop/diccionarios]
└─$ crunch 5 6 lalpha-numeric-space -o ejercicio6.txt
Crunch will now generate the following amount of data: 36015421 bytes
34 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 5198102

crunch: 100% completed generating output

┌──(kali㉿kali)-[~/Desktop/diccionarios]
└─$
```

Como resultado tenemos:

```
mcnal
mcnaa
mcnap
mcnah
mcna-
mcnan
mcnau
mcnam
mcnae
mcnar
mcnai
mcnac
mcnas
mcnpl
mcnpa
mcnpp
mcnph
mcnp-
mcnpn
mcnpu
mcnpm
mcnpe
mcnpr
mcnpi
mcnpc
mcnps
mcnhl
mcnha
mcnhp
```

Ejercicio 7: Crunch – Uso de un patrón

```
┌──(kali㉿kali)-[~/Desktop/diccionarios]
└─$ crunch 4 4 -t %%C^ ejercicio7.txt
Crunch will now generate the following amount of data: 16500 bytes
0 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 3300
00C!
00C@
00C#
00C$
00C%
00C^
00C&
00C*
00C(
00C)
00C-
00C_
00C+
00C=
00C~
00C`
00C[
00C]
00C{
00C}
00C|
```

Ejercicio 8: Crunch – Repeticiones de caracteres

El comando usado:

```
┌──(kali㉿kali)-[~/Desktop/diccionarios]
└─$ crunch 5 6 lalpha-numeric-space -o ejercicio8.txt -d 3^
Crunch will now generate the following amount of data: 36015421 bytes
34 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 5198102

crunch:  99% completed generating output
```

El resultado:

```
┌──(kali㉿kali)-[~/Desktop/diccionarios]
└─$ cat ejercicio8.txt
lllll
llll a
llll p
llll h
llll-
llll n
llll u
llll m
llll e
llll r
llll i
llll c
llll s
llla l
llla a
llla p
llla h
llla-
```

Ejercicio 9: Crunch – Combinaciones de palabras

```
┌──(kali㉿kali)-[~/Desktop/diccionarios]
└─$ crunch 5 10 -p Carlos Diaz Montes 2024 !
Crunch will now generate approximately the following amount of data: 2640 bytes
0 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 120
!2024CarlosDiazMontes
!2024CarlosMontesDiaz
!2024DiazCarlosMontes
!2024DiazMontesCarlos
!2024MontesCarlosDiaz
!2024MontesDiazCarlos
!Carlos2024DiazMontes
!Carlos2024MontesDiaz
!CarlosDiaz2024Montes
!CarlosDiazMontes2024
!CarlosMontes2024Diaz
!CarlosMontesDiaz2024
!Diaz2024CarlosMontes
!Diaz2024MontesCarlos
!DiazCarlos2024Montes
!DiazCarlosMontes2024
!DiazMontes2024Carlos
!DiazMontesCarlos2024
!Montes2024CarlosDiaz
!Montes2024DiazCarlos
!MontesCarlos2024Diaz
```