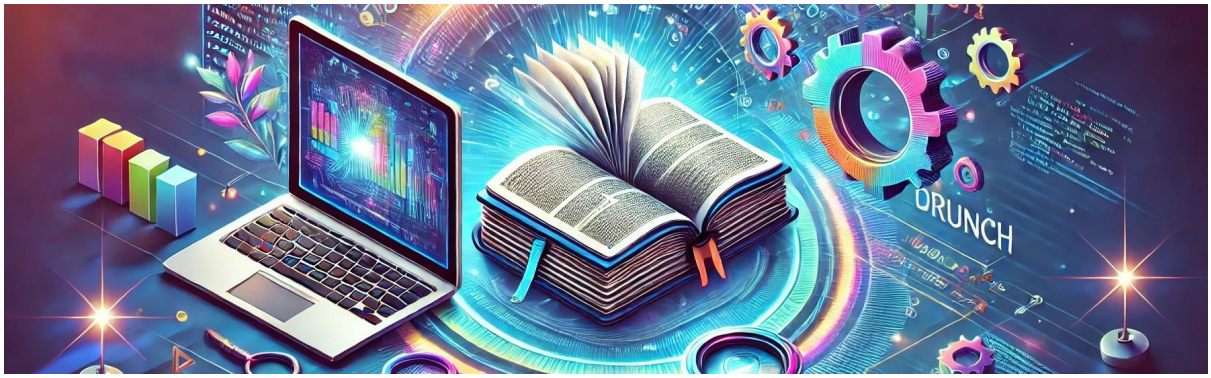

Creación de wordlist con Crunch

1. Descripción de Crunch

Crunch *es una herramienta de código abierto que permite generar diccionarios personalizados para ataques de fuerza bruta, como los que se usan en ataques a contraseñas. Crunch es útil para crear listas de palabras con diferentes combinaciones de caracteres, patrones y longitudes. Ofrece gran flexibilidad para crear listas específicas adaptadas a los requerimientos de seguridad o testing.*

---

2. Instalación de Crunch en Kali Linux

Crunch ya viene preinstalado en **Kali Linux**, pero en caso de que no lo tengas, puedes instalarlo con los siguientes comandos:

```bash
sudo apt update sudo
apt install crunch
```

Una vez instalado, verifica su correcta instalación usando:

```bash
crunch --help
```

Deberías ver una lista de opciones y ejemplos de uso.

3. Uso Básico de Crunch

El uso más básico de Crunch es generar diccionarios de un tamaño determinado con un conjunto de caracteres predefinido.

El comando básico para usar Crunch sigue la estructura:

Sintaxis Básica

```bash
crunch <min> <max> <charset>
```

min: Longitud mínima de las cadenas. **max**: Longitud máxima de las cadenas. **charset**: Conjunto de caracteres a utilizar.

Ejemplo:

```bash
crunch 4 4 abcdef
```

Esto generará todas las combinaciones de longitud 4 posibles utilizando los caracteres `**abcdef**`. **Salida:**

```bash
aaaa
aaab
aaac
aaad
...
fffd
fffe
ffff
```

Ejemplo 1: Generar combinaciones simples

Generar un diccionario con todas las combinaciones posibles de longitud 3, utilizando los caracteres `abc`.

```bash
crunch 3 3 abc
```

Esto genera todas las combinaciones de tres letras usando solo `**a**`, `**b**`, y `**c**`. **Salida:**

```bash
aaa
aab
aac
aba
abb
...
ccc
```

Este comando genera todas las combinaciones posibles de 3 caracteres a partir de **`abc`**, generando un total de 27 combinaciones (3^3 = 27).

Ejemplo 2: Generar combinaciones de longitud variable

Crunch también permite generar palabras de longitudes variables. Por ejemplo, para generar todas las combinaciones de entre 2 y 4 caracteres con los números **`0-9`**:

```bash
crunch 2 4 0123456789
```

Este comando genera todas las combinaciones de entre 2 y 4 dígitos utilizando solo números del 0 al 9.

Salida (abreviada):

```bash
00
01
...
9999
```

Este comando generará una gran cantidad de combinaciones, desde `**00**` hasta `**9999**`, pasando por todas las posibles combinaciones intermedias.

Ejemplo 3: Generar combinaciones con letras y números

Crunch permite usar cualquier combinación de letras, números y símbolos. Por ejemplo, generar combinaciones de 5 caracteres usando letras (`**abcde**`) y números (`**123**`):

```bash
crunch 5 5 abcde123
```

Este comando generará todas las combinaciones de 5 caracteres posibles usando las letras `**abcde**` y los números `**123**`. **Salida (abreviada):**

```bash
aaaaa
aaaab
aaaac
...
eeeee
```

Ejemplo 4: Generar con alfabetos predefinidos

Crunch facilita el uso de alfabetos completos mediante el uso de conjuntos de caracteres predefinidos. Para hacerlo, puedes usar los conjuntos proporcionados en

`**/usr/share/crunch/charset.lst**`. En este directorio, encontrarás conjuntos de caracteres predefinidos como: **mixalpha**: Letras minúsculas y mayúsculas (A-Z, a-z).

numeric: Números (0-9).

symbols: Símbolos especiales como `**!@#$%^&*()**`.

Ejemplo:

Si quieres generar una lista de contraseñas de longitud 6 usando letras minúsculas, letras mayúsculas y números:

```bash
crunch 6 6 -f /usr/share/crunch/charset.lst mixalpha-numeric
```

En este caso, **mixalpha-numeric** es un conjunto de caracteres predefinido que incluye letras minúsculas, mayúsculas y números. Esto generará combinaciones de exactamente 6 caracteres. **Salida (abreviada):**

```bash
aaaaaa
aaaaab
...
ZZZZZZ
```

Opciones adicionales útiles en uso básico

1. **Mostrar solo el conteo de combinaciones sin generar el archivo**:

Si quieres saber cuántas combinaciones generará Crunch sin imprimir todas las combinaciones en pantalla (o sin escribirlas en un archivo), puedes usar la opción **`-q`**:

```bash
crunch 4 4 abc123 -q
```

Esto devolverá el número de combinaciones generadas sin mostrarlas. **Salida:**

```bash
Crunch will now generate the following amount of data: 1296 bytes
1296 lines
```

En este caso, generaría 1296 combinaciones de longitud 4 usando los caracteres **`abc123`**.

2. **Limitar el número de combinaciones mostradas**:

Si estás generando muchas combinaciones y quieres limitar el número de resultados que ves en pantalla (para no saturar la consola), puedes usar la opción **`-c`** para especificar cuántas combinaciones se mostrarán a la vez.

```bash
crunch 4 4 abc123 -c 50
```

Esto mostrará 50 combinaciones a la vez y te pedirá que pulses enter para continuar.

En resumen, la estructura básica de Crunch permite especificar la longitud mínima y máxima de las palabras generadas, así como el conjunto de caracteres a usar. Aquí algunos puntos clave del uso básico:

Puedes crear combinaciones de diferentes longitudes entre un rango específico.

Puedes definir cualquier conjunto de caracteres manualmente o usar los predefinidos que Crunch proporciona.

Crunch genera todas las combinaciones posibles entre los parámetros dados.

Esta es la base para generar diccionarios simples. A medida que avancemos en el tutorial, exploraremos opciones más avanzadas como la generación de patrones específicos y la exclusión de caracteres.

4. Especificación de un Conjunto de Caracteres

Crunch permite usar conjuntos de caracteres predefinidos para facilitar la creación de diccionarios. Algunos conjuntos de caracteres comunes incluyen:

`**@**`: Minúsculas (a-z)

`**,**`: Mayúsculas (A-Z)

`**%**`: Números (0-9)

`**^**`: Símbolos especiales

Ejemplo:

```bash
crunch 6 6 -f /usr/share/crunch/charset.lst mixalpha-numeric -o dict.txt
```

Esto generará una lista de palabras de 6 caracteres con letras y números, guardada en un archivo llamado `dict.txt`.

5. Generación de Patrones Específicos

Crunch también permite crear diccionarios basados en patrones específicos. En el patrón, puedes definir qué tipo de carácter debe aparecer en cada posición.

`**@**`: Minúsculas

`**,**`: Mayúsculas

`**%**`: Números

`**^**`: Símbolos especiales **Ejemplo**:

```bash
crunch 8 8 -t pa%%%%%% -o passwordlist.txt
```

Esto generará palabras de 8 caracteres que comienzan con `**pa**` y tienen números (`**%**`) en las seis posiciones restantes. **Salida:**

```bash
pa000000
pa000001
pa000002
...
pa999999
```

6. Guardar el Diccionario en un Archivo

Puedes utilizar la opción `**-o**` para guardar el resultado en un archivo en lugar de mostrarlo en la consola.

Ejemplo:

```bash
crunch 5 5 abc123 -o mi_diccionario.txt
```

Esto genera todas las combinaciones de 5 caracteres posibles usando `**abc123**` y las guarda en `mi_diccionario.txt`.

---

7. Personalizar los Conjuntos de Caracteres

Además de los conjuntos predefinidos, puedes especificar tus propios conjuntos de caracteres. Por ejemplo, si quieres usar solo ciertos símbolos o una combinación específica de letras y números. **Ejemplo**:

```bash
crunch 4 4 -o symbols.txt -t [email protected]%^ -o symbols.txt
```

Esto genera una lista de todas las combinaciones posibles con los símbolos **`!@#$%^`**.

8. Excluir Caracteres Específicos

Puedes excluir ciertos caracteres del conjunto de caracteres predeterminado usando la opción **`-s`** para comenzar en una cadena específica y la opción **`-e`** para terminar en otra. Si quieres, por ejemplo, excluir combinaciones que comienzan con una letra en particular:

Ejemplo:

```bash
crunch 4 4 abc123 -s b000 -e bzzz
```

Esto generará combinaciones de 4 caracteres que comienzan en **`b000`** y terminan en **`bzzz`**.

9. Combinar la Exclusión de Caracteres con Patrones

Puedes combinar patrones con exclusiones para tener un control más preciso sobre los diccionarios generados.

Ejemplo:

```bash
crunch 8 8 -t ab%%%%% -s ab000001 -e ab999999 -o range_dict.txt
```

Este comando genera una lista de contraseñas que comienzan con **`ab`** y tienen números del **`000001`** al `999999`, guardando el resultado en `range_dict.txt`.

---

10. Otras Opciones Útiles Dividir archivos grandes: Si el diccionario generado es muy grande, puedes dividirlo en archivos más pequeños con **`-b`**.

Ejemplo:

```bash
crunch 5 5 abc123 -o diccionario.txt -b 10mb
```

Esto divide el archivo generado en fragmentos de 10 MB.

Generar combinaciones con espacios: Puedes agregar espacios en blanco entre los caracteres generados con la opción **`-s`**.

Ejemplo:

```bash
crunch 4 4 abc -s " "
```

Generará combinaciones de 4 caracteres separados por espacios.
