

MANUAL - Google_Dorking_Advanced + domain_checker.py

=====

MANUAL DE USO - Google_Dorking_Advanced Tool

=====

DESCRIPCIÓN GENERAL

Google_Dorking_Advanced es una herramienta pensada para pentesters y bug bounty hunters que quieran:

- Automatizar búsquedas avanzadas en Google usando dorks personalizados.
- Integrar análisis de dominios y subdominios.
- Usar la API de VirusTotal para verificar URLs y dominios.
- Exportar resultados en .txt / .json / .html.
- Comprobar subdominios usando patrones tipo *.dominio.com y reemplazarlos por wordlists.

Incluye un módulo adicional llamado "domain_checker.py" que permite leer dominios desde un archivo, expandir comodines (*) con palabras de una wordlist y verificar si están activos.

REQUISITOS PREVIOS

1. Tener Python 3.8 o superior.
2. Instalar dependencias con:


```
pip install -r requirements.txt
```
3. Crear un archivo .env en el directorio raíz con tu API key de VirusTotal:


```
VIRUSTOTAL_API_KEY=tu_api_key_aqui
```
4. Tener conexión a Internet.

ESTRUCTURA DE LA CARPETA

Google_Dorking_Advanced/

├── Google_Dorking_Advanced.py → Script principal de dorking

├── domain_checker.py → Módulo para comprobar dominios/subdominios

- └─ file_downloader.py → Descarga de resultados o archivos
 - └─ googlesearch.py → Módulo auxiliar de búsqueda
 - └─ results_parser.py → Limpieza y exportación de resultados
 - └─ html_template.html → Plantilla para exportar a HTML
 - └─ .env → Variables de entorno (API keys)
 - └─ requirements.txt → Dependencias necesarias
 - └─ wordlist.txt → Lista de posibles subdominios (opcional)
 - └─ dorks.txt → Lista de dorks para cargar
 - └─ README.txt → Este manual
-

USO DEL SCRIPT PRINCIPAL (Google_Dorking_Advanced.py)

Ejemplo básico:

```
python Google_Dorking_Advanced.py -d "site:ejemplo.com" -o resultados.txt
```

Usando lista de dorks:

```
python Google_Dorking_Advanced.py --dorks dorks.txt -o resultados.json
```

Con análisis en VirusTotal:

```
python Google_Dorking_Advanced.py -d "site:ejemplo.com" --virustotal
```

Exportar a HTML:

```
python Google_Dorking_Advanced.py -d "site:ejemplo.com" -o resultados.html
```

Parámetros:

-d, --dork → Dork único a buscar.

--dorks → Archivo con múltiples dorks.

-o, --output → Nombre del archivo de salida (.txt, .json, .html).

--virustotal → Habilita la integración con la API de VirusTotal.

--limit → Límite de resultados por dork (por defecto: 10).

-v, --verbose → Muestra más información por pantalla.

USO DEL SCRIPT domain_checker.py

Comprobar lista de dominios sin wordlist:

```
python domain_checker.py dominios.txt
```

Comprobar dominios con patrón * usando wordlist:

```
python domain_checker.py dominios.txt -w wordlist.txt
```

Parámetros:

input → Archivo con dominios o patrones (*.ejemplo.com*).

-w, --wordlist → *Archivo con posibles reemplazos para* .

-t, --threads → Número de hilos para acelerar el escaneo.

FORMATO DE ARCHIVOS

Ejemplo de "dominios.txt":

google.com

*.example.com

testsite.org

Ejemplo de "wordlist.txt":

www

mail

ftp

admin

Ejemplo de "dorks.txt":

site:ejemplo.com

intitle:"index of" site:gov

filetype:pdf "confidential"

CONSEJOS DE USO

- Usar proxies o VPN para evitar bloqueos por muchas peticiones a Google.
- Combinar el checker de dominios con los resultados del dorking.
- Si el dork devuelve muchas URLs, guarda en JSON y luego filtralo.
- La API de VirusTotal tiene un límite de peticiones diarias (depende del plan).

=====

Autor: Fabián I. Bertotti Guglielmone

Versión: 2.0 (2025)

=====