PRÁCTICA 4 UD 1 Footprinting y Enumeración con DNS

Hecho por: Izan Navarro

izan navarro lujan
IES SERRA PERENXISA

INDICE

1.PREPARACIÓN PREVIA:

1) Guardo una copia de seguridad del archivo de resolución DNS:

```
┌──(izan㉿kali)-[~/Desktop]
└─$ sudo cp /etc/resolv.conf /etc/resolv.conf.back
```

2) Apunta temporalmente el resolver de Kali al DNS de tu Ubuntu (ejemplo de la act. 10.10.10.20):

```
izan@kali: ~
Session  Actions  Edit  View  Help
 GNU nano 8.6                    /etc/resolv.conf
# Generated by NetworkManager
nameserver 10.10.10.20
```

3) Para restaurar los cambios efectuados en el resolv.conf ejecutaré el comando **“sudo mv etc/resolv.conf.back /etc/resolv.conf”**

2.FIERCE:

1) Ejecutamos el “fierce –help”:

```
options:
  -h, --help            show this help message and exit
  --domain DOMAIN       domain name to test
  --connect             attempt HTTP connection to non-RFC 1918 hosts
  --wide                scan entire class c of discovered records
  --traverse TRAVERSE   scan IPs near discovered records, this won't enter ad
jacent class c's
  --search SEARCH [SEARCH ...]
                        filter on these domains when expanding lookup
  --range RANGE         scan an internal IP range, use cidr notation
  --delay DELAY         time to wait between lookups
  --subdomains SUBDOMAINS [SUBDOMAINS ...]
                        use these subdomains
  --subdomain-file SUBDOMAIN_FILE
                        use subdomains specified in this file (one per line)
  --dns-servers DNS_SERVERS [DNS_SERVERS ...]
                        use these dns servers for reverse lookups
  --dns-file DNS_FILE   use dns servers specified in this file for reverse lo
okups (one per line)
  --tcp                 use TCP instead of UDP
```

2) Ejecutamos “fierce –domain [Nombre de dominio]” en nuestro caso, Google.com para hacer un ataque básico por diccionario:

```
┌──(izan㉿kali)-[~]
└─$ sudo fierce --domain ceti.local
NS: ubuntusrv.ceti.local.
SOA: ubuntusrv.ceti.local. (10.10.10.20)
Zone: success
{<DNS name @>: '@ 604800 IN SOA ubuntusrv root 3 604800 86400 2419200 604800\
n'
               '@ 604800 IN NS ubuntusrv',
 <DNS name ceo>: 'ceo 604800 IN A 10.10.10.4',
 <DNS name clase>: 'clase 604800 IN A 193.54.21.1',
 <DNS name consultor>: 'consultor 3600 IN CNAME ceo',
 <DNS name correo>: 'correo 604800 IN A 10.10.10.44',
 <DNS name dns>: 'dns 604800 IN A 10.10.10.33',
 <DNS name fsystem>: 'fsystem 604800 IN A 10.10.10.40',
 <DNS name ftp>: 'ftp 604800 IN A 10.10.10.69',
 <DNS name impresora>: 'impresora 604800 IN A 10.10.10.60',
 <DNS name mercedes>: 'mercedes 604800 IN A 35.157.228.228',
 <DNS name mx1>: 'mx1 604800 IN MX 10 10.10.10.99',
 <DNS name test>: 'test 604800 IN SRV 10 100 5060 ceo',
 <DNS name ubuntusrv>: 'ubuntusrv 604800 IN A 10.10.10.20',
 <DNS name xirivella>: 'xirivella 604800 IN A 10.10.10.89',
 <DNS name zaragoza>: 'zaragoza 604800 IN A 10.10.10.88'}
```

3) Ejecutamos “fierce –domain [ejemplo.com] –subdomain-file [ip_server]” para especifar un archivo con dns.

```
┌──(izan㉿kali)-[~]
└─$ sudo fierce --domain ceti.local --subdomain-file diccionarioIzan.txt
[sudo] password for izan:
NS: ubuntusrv.ceti.local.
SOA: ubuntusrv.ceti.local. (10.10.10.20)
Zone: success
{<DNS name @>: '@ 604800 IN SOA ubuntusrv root 3 604800 86400 2419200 604800\
n'
               '@ 604800 IN NS ubuntusrv',
 <DNS name ceo>: 'ceo 604800 IN A 10.10.10.4',
 <DNS name clase>: 'clase 604800 IN A 193.54.21.1',
 <DNS name consultor>: 'consultor 3600 IN CNAME ceo',
 <DNS name correo>: 'correo 604800 IN A 10.10.10.44',
 <DNS name dns>: 'dns 604800 IN A 10.10.10.33',
 <DNS name fsystem>: 'fsystem 604800 IN A 10.10.10.40',
 <DNS name ftp>: 'ftp 604800 IN A 10.10.10.69',
 <DNS name impresora>: 'impresora 604800 IN A 10.10.10.60',
 <DNS name mercedes>: 'mercedes 604800 IN A 35.157.228.228',
 <DNS name mx1>: 'mx1 604800 IN MX 10 10.10.10.99',
 <DNS name test>: 'test 604800 IN SRV 10 100 5060 ceo',
 <DNS name ubuntusrv>: 'ubuntusrv 604800 IN A 10.10.10.20',
 <DNS name xirivella>: 'xirivella 604800 IN A 10.10.10.89',
 <DNS name zaragoza>: 'zaragoza 604800 IN A 10.10.10.88'}
```

3.DNSRECON:

1) Usamos el comando “dnsrecon – help”:

```
options:
  -h, --help            show this help message and exit
  -d, --domain DOMAIN   Target domain.
  -n, --name_server NS_SERVER
                        Domain server to use. If none is given, the SOA of the target will be used. Multiple s
ervers can be specified using a comma separated list.
  -r, --range RANGE     IP range for reverse lookup brute force in formats   (first-last) or in (range/bitmask
).
  -D, --dictionary DICTIONARY
                        Dictionary file of subdomain and hostnames to use for brute force.
  -f                    Filter out of brute force domain lookup, records that resolve to the wildcard defined
IP address when saving records.
  -a                    Perform AXFR with standard enumeration.
  -s                    Perform a reverse lookup of IPv4 ranges in the SPF record with standard enumeration.
  -b                    Perform Bing enumeration with standard enumeration.
  -y                    Perform Yandex enumeration with standard enumeration.
  -k                    Perform crt.sh enumeration with standard enumeration.
  -w                    Perform deep whois record analysis and reverse lookup of IP ranges found through Whois
 when doing a standard enumeration.
  -z                    Performs a DNSSEC zone walk with standard enumeration.
  --threads THREADS     Number of threads to use in reverse lookups, forward lookups, brute force and SRV reco
rd enumeration.
  --lifetime LIFETIME   Time to wait for a server to respond to a query. default is 3.0
  --tcp                 Use TCP protocol to make queries.
  --db DB               SQLite 3 file to save found records.
  -x, --xml XML         XML file to save found records.
  -c, --csv CSV         Save output to a comma separated value file.
  -j, --json JSON       save output to a JSON file.
  --iw                  Continue brute forcing a domain even if a wildcard record is discovered.
  --disable_check_recursion
                        Disables check for recursion on name servers
  --disable_check_bindversion
                        Disables check for BIND version on name servers
  -V, --version         Show DNSrecon version
  -v, --verbose         Enable verbose
  -t, --type TYPE       Type of enumeration to perform.
                        Possible types:
                            std:      SOA, NS, A, AAAA, MX and SRV.
                            rvl:      Reverse lookup of a given CIDR or IP range.
                            brt:      Brute force domains and hosts using a given dictionary.
```

2) Ejecutamos el comando “dnsrecon -d ceti.local -n 10.10.10.20 -t brt” para realizar un ataque de fuerza bruta sin usar directorio personalizado.

```
┌──(izan㉿kali)-[~]
└─$ dnsrecon -d ceti.local -n 10.10.10.20 -t brt
[*] No dictionary file has been specified.
[*] Using the dictionary file: /usr/share/dnsrecon/dnsrecon/data/namelist.txt (provided by tool)
[*] brt: Performing host and subdomain brute force against ceti.local ...
[+]      A correo.ceti.local 10.10.10.44
[+]      A dns.ceti.local 10.10.10.33
[+]      A ftp.ceti.local 10.10.10.69
[+] 3 Records Found
```

3) Ejecutamos el comando “dnsrecon -d ceti.local -n 10.10.10.20 -t brt -D (diccionario.ej)” para realizar un ataque de fuerza bruta a un directorio personalizado.

```
┌──(izan㉿kali)-[~]
└─$ dnsrecon -d ceti.local -n 10.10.10.20 -t brt -D diccionarioIzan.txt
[*] Using the dictionary file: diccionarioIzan.txt (provided by user)
[*] brt: Performing host and subdomain brute force against ceti.local ...
[+]      A fsystem.ceti.local 10.10.10.40
[+] 1 Records Found
```

4) Ejecutamos el comando “dnsrecon -d ceti.local -n 10.10.10.20 -t brt --json /home/kali/resultados.json” para guardar los resultados del ataque en un fichero .json “resultados”:

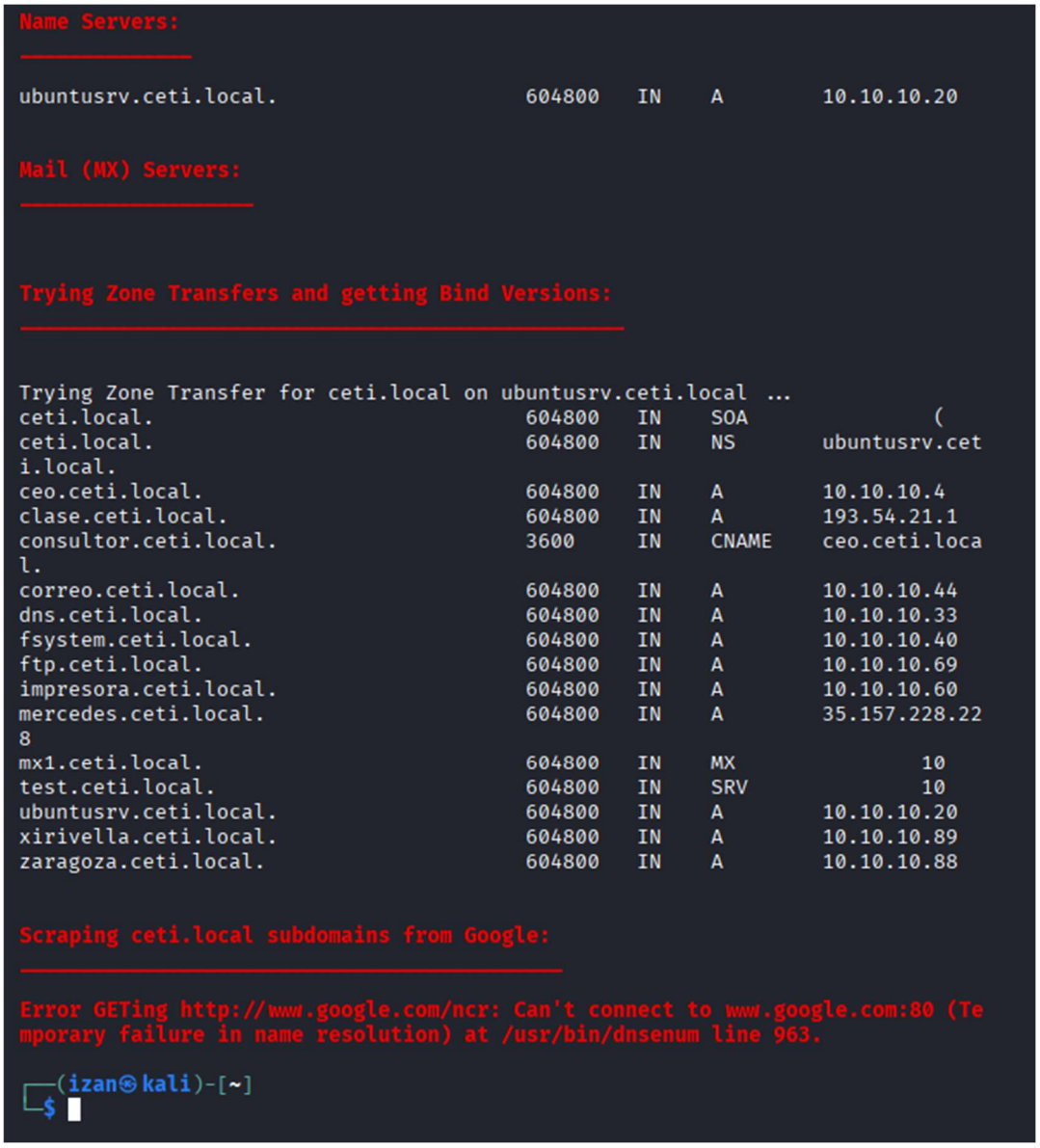
```
┌──(izan㉿kali)-[~]
└─$ dnsrecon -d ceti.local -n 10.10.10.20 -t brt --json /home/kali/Desktop/resultados.json
[*] No dictionary file has been specified.
[*] Using the dictionary file: /usr/share/dnsrecon/dnsrecon/data/namelist.txt (provided by tool)
[*] brt: Performing host and subdomain brute force against ceti.local ...
[+]      A correo.ceti.local 10.10.10.44
[+]      A dns.ceti.local 10.10.10.33
[+]      A ftp.ceti.local 10.10.10.69
[+] 3 Records Found
[*] Saving records to JSON file: /home/kali/Desktop/resultados.json

┌──(izan㉿kali)-[~]
└─$
```

4.DNSENUM:

1) Usamos el comando “dnsenum –help”:

```
GENERAL OPTIONS:
  --dnsserver   <server>
                        Use this DNS server for A, NS and MX queries.
  --enum                Shortcut option equivalent to --threads 5 -s 15 -w.
  -h, --help            Print this help message.
  --noreverse           Skip the reverse lookup operations.
  --nocolor             Disable ANSIColor output.
  --private             Show and save private ips at the end of the file domain_ips.txt
.
  --subfile <file>      Write all valid subdomains to this file.
  -t, --timeout <value> The tcp and udp timeout values in seconds (default: 10s).
  --threads <value>     The number of threads that will perform different queries.
  -v, --verbose         Be verbose: show all the progress and all the error messages.
GOOGLE SCRAPING OPTIONS:
  -p, --pages <value>   The number of google search pages to process when scraping name
s,
                        the default is 5 pages, the -s switch must be specified.
  -s, --scrap <value>   The maximum number of subdomains that will be scraped from Goog
le (default 15).
BRUTE FORCE OPTIONS:
  -f, --file <file>     Read subdomains from this file to perform brute force. (Takes p
riority over default dns.txt)
  -u, --update  <a|g|r|z>
```

2) Ejecutamos “dnsenum --dnsserver 10.10.10.20 ceti.local” para poder hacer un ataque sin diccionario personalizado.



3) realizamos el comando “dnsenum --dnsserver 10.10.10.20 -f /ruta/tu/diccionario.txt ceti.local” para realizar un ataque a un diccionario personalizado.

```
Name Servers:
_____________

ubuntusrv.ceti.local.                   604800   IN    A        10.10.10.20


Mail (MX) Servers:
__________________


Trying Zone Transfers and getting Bind Versions:
________________________________________________


Trying Zone Transfer for ceti.local on ubuntusrv.ceti.local ...
ceti.local.                             604800   IN    SOA                (
ceti.local.                             604800   IN    NS       ubuntusrv.ceti.local.
ceo.ceti.local.                         604800   IN    A        10.10.10.4
clase.ceti.local.                       604800   IN    A        193.54.21.1
consultor.ceti.local.                   3600     IN    CNAME    ceo.ceti.local.
correo.ceti.local.                      604800   IN    A        10.10.10.44
dns.ceti.local.                         604800   IN    A        10.10.10.33
fsystem.ceti.local.                     604800   IN    A        10.10.10.40
ftp.ceti.local.                         604800   IN    A        10.10.10.69
impresora.ceti.local.                   604800   IN    A        10.10.10.60
mercedes.ceti.local.                    604800   IN    A        35.157.228.228
mx1.ceti.local.                         604800   IN    MX              10
test.ceti.local.                        604800   IN    SRV             10
ubuntusrv.ceti.local.                   604800   IN    A        10.10.10.20
xirivella.ceti.local.                   604800   IN    A        10.10.10.89
zaragoza.ceti.local.                    604800   IN    A        10.10.10.88


Brute forcing with diccionarioIzan.txt:
_______________________________________


ceti.local class C netranges:
_____________________________

 35.157.228.0/24
 193.54.21.0/24


Performing reverse lookup on 512 ip addresses:
______________________________________________


0 results out of 512 IP addresses.
```

5.THEHARVESTER:

1) Ejecutamos el comando theHarvester –help y nos salen todas estas opciones:

```
  -h, --help            show this help message and exit
  -d, --domain DOMAIN   Company name or domain to search.
  -l, --limit LIMIT     Limit the number of search results, default=500.
  -S, --start START     Start with result number X, default=0.
  -p, --proxies         Use proxies for requests, enter proxies in proxies.yaml.
  -s, --shodan          Use Shodan to query discovered hosts.
  --screenshot SCREENSHOT
                        Take screenshots of resolved domains specify output
                        directory: --screenshot output_directory
  -v, --virtual-host    Verify host name via DNS resolution and search for virtual
                        hosts.
  -e, --dns-server DNS_SERVER
                        DNS server to use for lookup.
  -t, --take-over       Check for takeovers.
  -r, --dns-resolve [DNS_RESOLVE]
                        Perform DNS resolution on subdomains with a resolver list or
                        passed in resolvers, default False.
  -n, --dns-lookup      Enable DNS server lookup, default False.
  -c, --dns-brute       Perform a DNS brute force on the domain.
  -f, --filename FILENAME
                        Save the results to an XML and JSON file.
  -w, --wordlist WORDLIST
                        Specify a wordlist for API endpoint scanning.
  -a, --api-scan        Scan for API endpoints.
  -q, --quiet           Suppress missing API key warnings.
  -b, --source SOURCE   baidu, bevigil, bing, bingapi, brave, bufferoverun,
                        builtwith, censys, certspotter, criminalip, crtsh, dehashed,
                        dnsdumpster, duckduckgo, fullhunt, github-code, hackertarget,
                        haveibeenpwned, hunter, hunterhow, intelx, leaklookup,
```

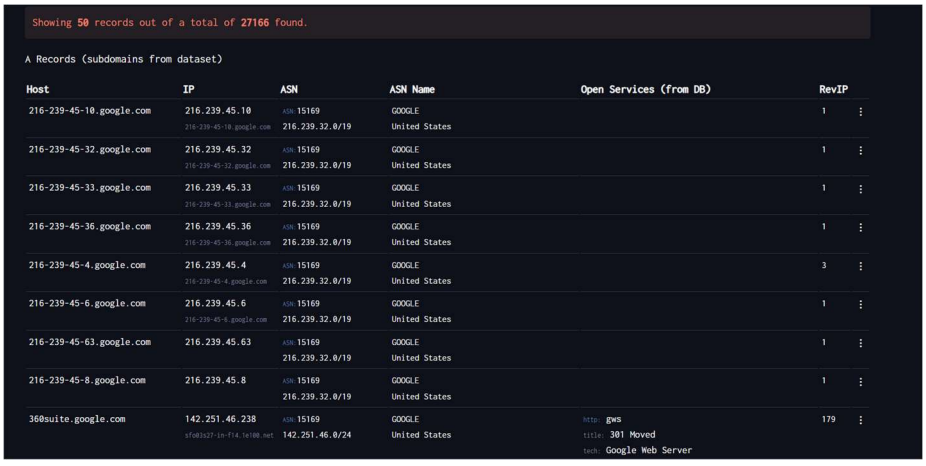
2) Usaremos el comando “theHarvester -d ceti.local -c -e 10.10.10.20 -v” para usar la función de búsqueda por fuerza bruta con los dns del archivo dns-names que bien dentro de la carpeta “wordlists”.

```
┌──(izan㉿kali)-[/usr/…/dist-packages/theHarvester/data/wordlists]
└─$ theHarvester -d ceti.local -c -e 10.10.10.20 -v
Read proxies.yaml from /etc/theHarvester/proxies.yaml
*******************************************************************
*  _   _                                            _             *
* | |_| |__   ___    /\  /\__ _ _ ____   _____  ___| |_ ___ _ __  *
* | __|  _ \ / _ \  / /_/ / _` | '__\ \ / / _ \/ __| __/ _ \ '__| *
* | |_| | | |  __/ / __  / (_| | |   \ V /  __/\__ \ ||  __/ |    *
*  \__|_| |_|\___| \/ /_/ \__,_|_|    \_/ \___||___/\__\___|_|    *
*                                                                 *
* theHarvester 4.8.2                                              *
* Coded by Christian Martorella                                   *
* Edge-Security Research                                          *
* cmartorella@edge-security.com                                   *
*                                                                 *
*******************************************************************

[*] No IPs found.

[*] No emails found.

[*] No people found.

[*] No hosts found.



[*] Starting DNS brute force.
Starting DNS brute forcing with 4989 words
```

```
www.jp.ceti.local:
mall.ceti.local:
tristan.ceti.local:
company.ceti.local:
template.ceti.local:
engineering.ceti.local:
rh.ceti.local:
stargate.ceti.local:
mobility.ceti.local:
saas.ceti.local:
tickets.ceti.local:
wholesale.ceti.local:
www.labs.ceti.local:
mk.ceti.local:
prime.ceti.local:
www.reg.ceti.local:
patch.ceti.local:
s19.ceti.local:
widgets.ceti.local:
websrv.ceti.local:
serv2.ceti.local:
adm.ceti.local:
host2123.ceti.local:
consulting.ceti.local:
messenger.ceti.local:
zero.ceti.local:
fs2.ceti.local:
depot.ceti.local:
www.preview.ceti.local:
box.ceti.local:
chelyabinsk.ceti.local:
cms.ceti.local:
teszt.ceti.local:
fms.ceti.local:
pms.ceti.local:
kraken.ceti.local:
sigma.ceti.local:
ts1.ceti.local:
rabbit.ceti.local:
atlantis.ceti.local:
isis.ceti.local:
www.abc.ceti.local:
sbc.ceti.local:
toyota.ceti.local:
greetings.ceti.local:
test1.ceti.local:
astra.ceti.local:
sss.ceti.local:
www.br.ceti.local:
att.ceti.local:
europe.ceti.local:
livestream.ceti.local:
```

6. DNSDUMPSTER:

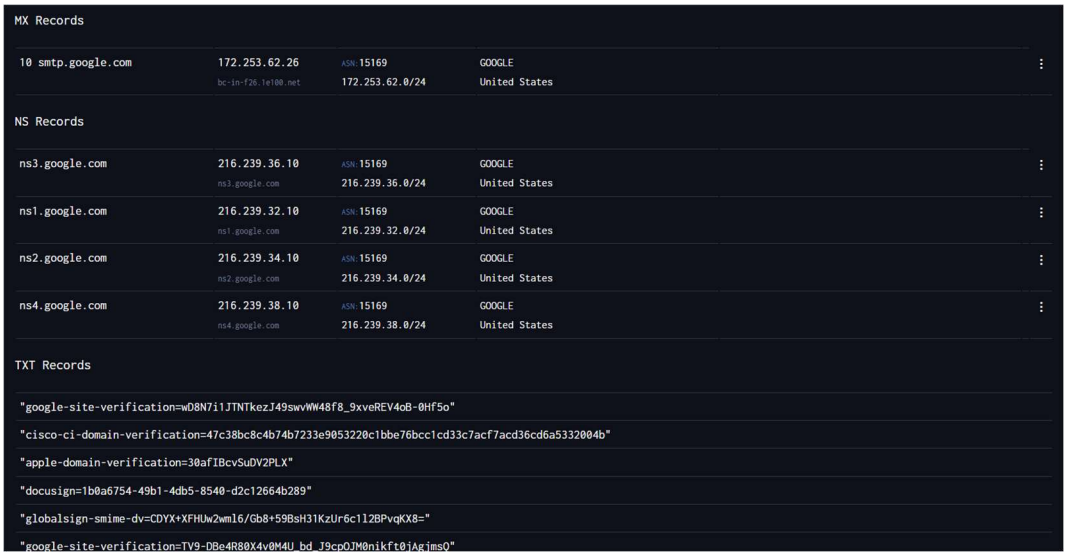
1) Buscamos el dominio público de “Google.com”:



Showing 50 records out of a total of 27166 found.

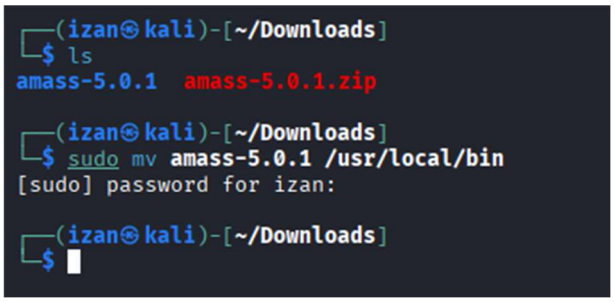
A Records (subdomains from dataset)

Host	IP	ASN	ASN Name	Open Services (from DB)	RevIP
216-239-45-10.google.com	216.239.45.10 216-239-45-10.google.com	ASN 15169 216.239.32.0/19	GOOGLE United States		1
216-239-45-32.google.com	216.239.45.32 216-239-45-32.google.com	ASN 15169 216.239.32.0/19	GOOGLE United States		1
216-239-45-33.google.com	216.239.45.33 216-239-45-33.google.com	ASN 15169 216.239.32.0/19	GOOGLE United States		1
216-239-45-36.google.com	216.239.45.36 216-239-45-36.google.com	ASN 15169 216.239.32.0/19	GOOGLE United States		1
216-239-45-4.google.com	216.239.45.4 216-239-45-4.google.com	ASN 15169 216.239.32.0/19	GOOGLE United States		3
216-239-45-6.google.com	216.239.45.6 216-239-45-6.google.com	ASN 15169 216.239.32.0/19	GOOGLE United States		1
216-239-45-63.google.com	216.239.45.63	ASN 15169 216.239.32.0/19	GOOGLE United States		1
216-239-45-8.google.com	216.239.45.8	ASN 15169 216.239.32.0/19	GOOGLE United States		1
360suite.google.com	142.251.46.238 sfo03s27-in-f14.1e100.net	ASN 15169 142.251.46.0/24	GOOGLE United States	http: gws title: 301 Moved tech: Google Web Server	179



7. AMASS:

1) Descargamos e instalamos la última versión reconocida en el repositorio de Github.

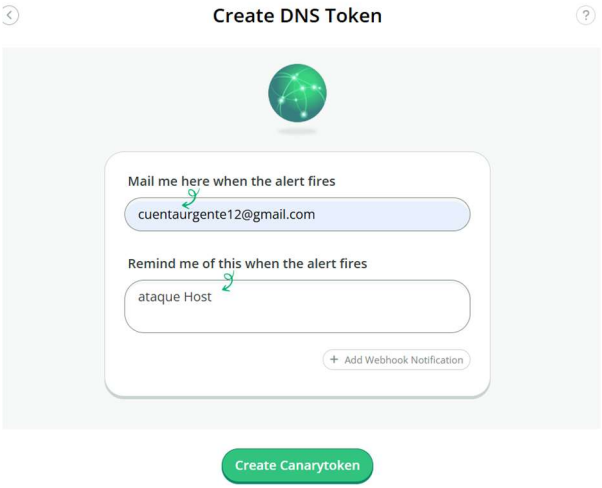


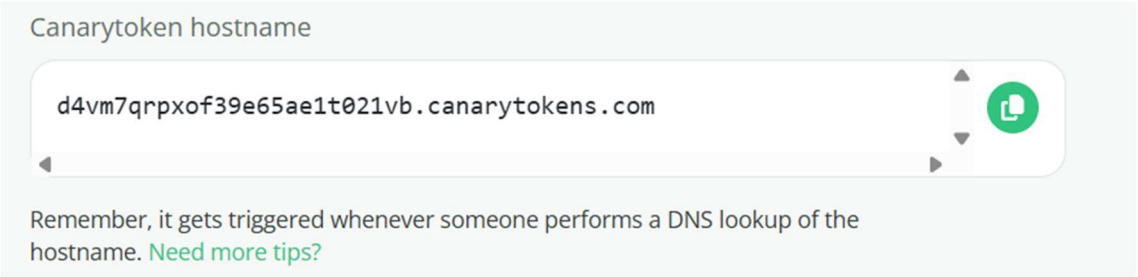
2) Realiza: Enumeración de subdominios con fuerza bruta. Pero sin resultados ya que no encuentra nada con nuestro servidor 10.10.10.20

```
Querying Searx for google.com subdomains
Querying Searchcode for google.com subdomains
Querying Active DNS for google.com subdomains
Querying DuckDuckGo for google.com subdomains
Querying Gists for google.com subdomains
Querying PKey for google.com subdomains
Querying Maltiverse for google.com subdomains
Querying Crtsh for google.com subdomains
Querying HAW for google.com subdomains
Querying RapidDNS for google.com subdomains
Querying DNS SRV for google.com subdomains
Querying Greynoise for google.com subdomains
Querying UKWebArchive for google.com subdomains
Querying AnubisDB for google.com subdomains
Querying CertSpotter for google.com subdomains
Querying Synapsint for google.com subdomains
Querying SubdomainCenter for google.com subdomains
Querying AbuseIPDB for google.com subdomains
Querying Pulsedive for google.com subdomains
Querying Sublist3rAPI for google.com subdomains
Querying Google for google.com subdomains
Querying ThreatMiner for google.com subdomains
Querying Active Crawl for google.com subdomains
Querying Mnemonic for google.com subdomains
mars.google.com (FQDN) ⟶ cname_record ⟶ www3.l.google.com (FQDN)
security.google.com (FQDN) ⟶ cname_record ⟶ www3.l.google.com (FQDN)
help.google.com (FQDN) ⟶ cname_record ⟶ www3.l.google.com (FQDN)
sprint.google.com (FQDN) ⟶ cname_record ⟶ www3.l.google.com (FQDN)
billing.google.com (FQDN) ⟶ a_record ⟶ 142.250.185.14 (IPAddress)
billing.google.com (FQDN) ⟶ aaaa_record ⟶ 2a00:1450:4003:807::200e (IPAddress)
shop.google.com (FQDN) ⟶ a_record ⟶ 74.125.206.92 (IPAddress)
shop.google.com (FQDN) ⟶ aaaa_record ⟶ 2a00:1450:400c:c02::5c (IPAddress)
products.google.com (FQDN) ⟶ cname_record ⟶ www3.l.google.com (FQDN)
labs.google.com (FQDN) ⟶ cname_record ⟶ www3.l.google.com (FQDN)
edu.google.com (FQDN) ⟶ a_record ⟶ 142.250.200.78 (IPAddress)
edu.google.com (FQDN) ⟶ aaaa_record ⟶ 2a00:1450:4003:80e::200e (IPAddress)
dir.google.com (FQDN) ⟶ cname_record ⟶ directory.google.com (FQDN)
investors.google.com (FQDN) ⟶ cname_record ⟶ www3.l.google.com (FQDN)
google.com (FQDN) ⟶ cname_record ⟶ forcesafesearch.google.com (FQDN)
1.google.com (FQDN) ⟶ cname_record ⟶ www3.l.google.com (FQDN)
admin.google.com (FQDN) ⟶ a_record ⟶ 142.250.200.110 (IPAddress)
admin.google.com (FQDN) ⟶ aaaa_record ⟶ 2a00:1450:4003:804::200e (IPAddress)
tv.google.com (FQDN) ⟶ cname_record ⟶ www3.l.google.com (FQDN)
support.google.com (FQDN) ⟶ a_record ⟶ 216.58.215.174 (IPAddress)
support.google.com (FQDN) ⟶ aaaa_record ⟶ 2a00:1450:4003:80f::200e (IPAddress)
ns3.google.com (FQDN) ⟶ a_record ⟶ 216.239.36.10 (IPAddress)
ns3.google.com (FQDN) ⟶ aaaa_record ⟶ 2001:4860:4802:36::a (IPAddress)
local.google.com (FQDN) ⟶ cname_record ⟶ maps.l.google.com (FQDN)
142.250.160.0/19 (Netblock) ⟶ contains ⟶ 142.250.185.14 (IPAddress)
15169 (ASN) ⟶ managed_by ⟶ GOOGLE - Google LLC (RIROrganization)
15169 (ASN) ⟶ announces ⟶ 142.250.160.0/19 (Netblock)
```

8.CANARY:

1) Entramos en el enlace de https://canarytokens.org/nest/ y sacamos el token canary





2) Añadimos la siguiente información dentro del UbuntuServer (Ultima línea) poniendo el token proporcionado por canary.

```
  GNU nano 7.2                                                  /etc/bind/zones/db.ceti.local
$TTL 604800
; SOA record with MNAME and RNAME updated
@       IN      SOA     ubuntusrv root.ceti.local. (
                        3               ; Serial Note: increment after each change
                        604800          ; Refresh
                        86400           ; Retry
                        2419200         ; Expire
                        604800 )        ; Negative Cache TTL

; Name server record
@       IN      NS      ubuntusrv.ceti.local.

; A record for name server
ubuntusrv       IN      A       10.10.10.20

; A record for clients
impresora        IN      A       10.10.10.60
ceo             IN      A       10.10.10.4
fsystem         IN      A       10.10.10.40
correo          IN      A       10.10.10.44
ftp             IN      A       10.10.10.69
dns             IN      A       10.10.10.33

test            SRV     10 100 5060 ceo.ceti.local.
mx1             IN      MX      10 10.10.10.99
consultor       3600    IN      CNAME   ceo.ceti.local.
zaragoza        IN      A       10.10.10.88
xirivella       IN      A       10.10.10.89
mercedes        IN      A       35.157.228.228
clase           IN      A       193.54.21.1
windows         IN      CNAME   mwrv1wqinbnus90dqgjnq2uln.canarytokens.com
```

3) Hacemos un Reload al bind del Ubuntu Server para guardar los cambios y así poder comprobarlo desde nuestra VM Kali.

```
root@ubuntusrv:~# sudo systemctl reload bind9
root@ubuntusrv:~# dig @10.10.10.20 windows.ceti.local CNAME +short
mwrv1wqinbnus90dqgjnq2uln.canarytokens.com.ceti.local.
```

4) Dentro de Kali hacemos “dig Windows.ceti.local” y nos mostrará la última línea introducido al UbuntuServer con los cambios.

```
┌──(izan㉿kali)-[~]
└─$ dig windows.ceti.local

; <<>> DiG 9.20.11-4+b1-Debian <<>> windows.ceti.local
;; global options: +cmd
;; Got answer:
;; WARNING: .local is reserved for Multicast DNS
;; You are currently testing what happens when an mDNS query is leaked to DNS
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 2639
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: 38fdd78ef68835f70100000068ffbd6a8fbcea0845c7f40c (good)
;; QUESTION SECTION:
;windows.ceti.local.            IN      A

;; ANSWER SECTION:
windows.ceti.local.     604800  IN      CNAME   mwrv1wqinbnus90dqgjnq2uln.canarytokens.com.ceti.local.

;; AUTHORITY SECTION:
ceti.local.             604800  IN      SOA     ubuntusrv.ceti.local. root.ceti.local. 3 604800 86400 2419200 604800

;; Query time: 4 msec
;; SERVER: 10.10.10.20#53(10.10.10.20) (UDP)
;; WHEN: Mon Oct 27 14:43:53 EDT 2025
;; MSG SIZE  rcvd: 193
```