# Ethical Hacking and Countermeasures
# DNSRecon Cheat Sheet

EC-Council
Building A Culture Of Security

**DNSRecon**
*Source: https://github.com*

DNSRecon is for performing the reverse DNS lookup on the target host, check NS Records for zone transfer, exploit vulnerabilities and obtain network information of a target domain and further launch Internet-based attacks, enumerate DNS Records for domains (MX, SOA, NS, A, AAAA, SPF, and TXT), perform common SRV record enumeration, Top Level Domain (TLD) expansion, check for wildcard resolution, brute Force subdomain and host A and AAAA records given a domain and a wordlist, perform a PTR Record lookup for a given IP Range or CIDR, check a DNS server cached records for A, AAAA and CNAME Records provided a list of host records in a text file to check, enumerate common mDNS records in the local network enumerate hosts and subdomains using Google.

## Syntax

```
dnsrecon.py [-h] [-d DOMAIN] [-n
NS_SERVER] [-r RANGE] [-D
DICTIONARY][-f] [-t TYPE] [-a] [-s] [-g]
[-b] [-k] [-w] [-z] [--threads THREADS]
[--lifetime LIFETIME] [--tcp] [--db DB]
[-x XML] [-c CSV] [-j JSON] [--iw] [-v]
```

## DNSRecon Installation

| |
|---|
| aptitude install dnsrecon On Parrot, or: |
| git clone https://github.com/darkoperator/dnsrecon.git |
| cd dnsrecon |
| pip install -r requirements.txt |

| | |
|---|---|
| `--db` | SQLite 3 file |
| `--xml` | XML file |
| `--json` | JSON file |
| `--csv` | CSV file |

| Command | Description |
|---|---|
| `dnsrecon -d <Target Domain> -j <results json file>` | Save results in a json file |
| `dnscan.py -l $domains_file -o outfile -w $wordlist` | Subdomain brute-force of domains listed in a file (one by line) |
| `dnscan.py -d target.com -o outfile -w $wordlist` | Subdomain brute-force of a domain |
| `dnssearch -domain <Target Domain> -wordlist $wordlist` | Dnssearch Subdomain brute-force |
| `dnsrecon -d zonetransfer.me` | Use Robin Wood's zonetransfer.me site to enumerate and Run a scan |
| `dnsrecon -d zonetransfer.me -D <namelist.txt> -t brt` | Brute Force scan |
| `dnsrecon -d zonetransfer.me -a` | Zone Transfer |
| `dnsrecon -d zonetransfer.me -a --db ~/Desktop/dnsrecon/dnsrecon-db` | Look at SQLite database file |
| `dnsrecon -d zonetransfer.me -a --xml ~/Desktop/dnsrecon/dnsrecon-xml` | Save the results in XML format |
| `dnsrecon -d TARGET -D /usr/share/wordlists/dnsmap.txt -t std --xml ouput.xml` | DNS Zone Transfers |
| `dnsrecon -d <Target IP> -t std -D /usr/share/wordlists/dnsmap.txt` | DNS (reverse) lookups / Enumeration DNS / Brute force subdomains |
| `$ python dnsrecon.py -n ns1.<Target Domain> -d <Target Domain> -D subdomains-top1mil-5000.txt -t brt` | DNS enumeration tool |
| `dnsrecon -w` | DNS Reconnaissance |

| Command | Description |
|---|---|
| `dnsrecon -r <Target IP range>` | Reverse DNS lookup on the target host |
| `dnsrecon -t axfr -d <Target Domain>` | DNS zone transfer |
| `dnsrecon -d <Target Domain> -z` | Zone enumeration against a target domain |
| `dnsrecon -d <Target Domain> -a ./dnsrecon.py -d <Target Domain> -a or dnsrecon -d <Target Domain> -t axfr ./dnsrecon.py -d <Target Domain> -t axfr` | Zone transfer |
| `dnsrecon -r <start Target IP>-<end Target IP> ./dnsrecon.py -r <start Target IP>-<end Target IP> ./dnsrecon.py -r <Target IP range>` | Reverse Lookup against IP range |
| `dnsrecon -d <Target Domain> -s ./dnsrecon.py -d <Target Domain> -s` | Reverse Lookup against all ranges in SPF records |
| `dnsrecon -d <Target Domain> -D <namelist.txt> -t brt ./dnsrecon.py -d <Target Domain> -D <namelist> -t brt` | Domain Brute Force Enumeration |
| `dnsrecon -d <Target Domain> -D /usr/share/wordlists/dnsmap.txt -t std --xml ouput.xml` | DNS Brute force |
| `dnsrecon -t snoop -n <Server IP> -D <namelist.txt> ./dnsrecon.py -t snoop -n <Server IP> -D <dictionary file>` | Cache Snooping against name servers |
| `dnsrecon -d <Target Domain> ./dnsrecon.py -d <Target Domain>` | Standard Records Enumeration/ enumerate DNS record of targeted website |
| `dnsrecon -d <Target Domain> -t zonewalk` | Zone Walking |
| `dnsrecon -d <Target Domain> -t rvl` | Reverse lookup of a given CIDR or IP range |
| `dnsrecon -d <Target Domain> -t brt -D <Subdomains Dictionary>` | Brute force domains and hosts using a given dictionary |
| `dnsrecon -d <Target Domain> -t brt -D <Subdomains Dictionary> --iw` | Brute force domains and hosts using a given dictionary. Continue brute-forcing a domain even if wildcard records are discovered |
| `dnsrecon -d <Target Domain> -t srv` | SRV records |
| `dnsrecon -d <Target Domain> -t axfr` | Test all NS servers for a zone transfer |
| `dnsrecon -d <Target Domain> -t goo` | Google search for subdomains and hosts |
| `dnsrecon -d <Target Domain> -t tld` | Remove the TLD of a given domain and test against all TLDs registered in IANA |
| `dnsrecon -d <Target Domain> -t zonewalk` | DNSSEC zone walk using NSEC records |
| `dnsrecon -d <Target Domain> --db <results sqlite File>` | Save results in a sqlite file |
| `dnsrecon -d demo.com --xml <results xml file>` | Save results in an xml file |
| `dnsrecon -d <Target Domain> -c <results csv file>` | Save results in a csv file |

## Arguments

| | |
|---|---|
| `-h, --help` | Help message and exit |
| `-d DOMAIN, --domain DOMAIN` | Target domain |
| `-n NS_SERVER, --name_server NS_SERVER` | Domain server to use. If none is given, the SOA of the target will be used |
| `-n nsserver.com` | Use a custom name server |
| `-r RANGE, --range RANGE` | IP range for reverse lookup brute force in formats (first-last) or in (range/bitmask) |
| `-D DICTIONARY, --dictionary DICTIONARY` | Dictionary file of subdomain and hostnames to use for brute force. Filter out of brute force domain lookup, records that resolve to the wildcard defined IP address when saving records |
| `-f` | Filter out of brute force domain lookup, records that resolve to the wildcard defined IP address when saving records |
| `-t TYPE, --type TYPE` | Type of enumeration to perform |
| `-a` | AXFR with standard enumeration |
| `-r` | Recursively scan subdomains |
| `-s` | Reverse lookup of IPv4 ranges in the SPF record with standard enumeration |
| `-T` | TLD expansion |
| `-g` | Google enumeration with standard enumeration |
| `-b` | Bing enumeration with standard enumeration |
| `-k` | Crt.sh enumeration with standard enumeration |
| `-w` | Deep whois record analysis and reverse lookup of IP ranges found through Whois when doing a standard enumeration |
| `-z` | DNSSEC zone walk with standard enumeration |
| `--threads THREADS` | Number of threads to use in reverse lookups, forward lookups, brute force, and SRV record enumeration |
| `--lifetime LIFETIME` | Time to wait for a server to respond to a query |
| `--tcp` | Use TCP protocol to make queries |
| `--db DB` | SQLite 3 file to save found records/ save results to SQLite database file |
| `-x XML, --xml XML` | XML file to save found records/ save results to the XML file |
| `-c CSV, --csv CSV` | Comma-separated value file |
| `-j JSON, --json JSON` | JSON file |
| `-i $file` | Output discovered IP addresses to a text file |
| `--iw` | Continue brute-forcing a domain even if wildcard records are discovered |
| `-v` | Enable verbose |