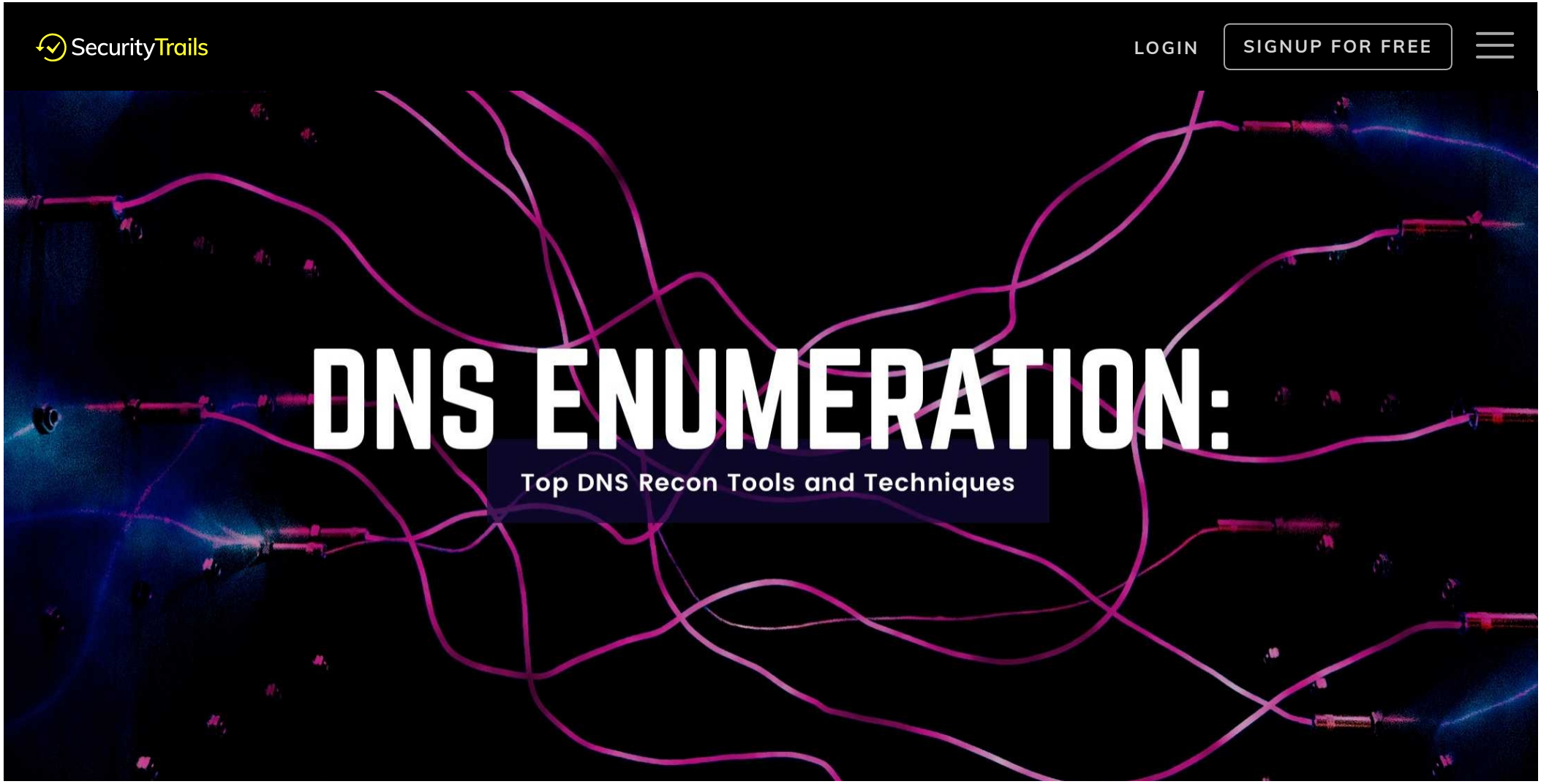



TIPS TOOLS RECONNAISSANCE

SECURITYTRAILS BLOG · OCT 03 · SECURITYTRAILS TEAM

# DNS Enumeration: Top DNS Recon Tools and Techniques

Reading time: 13 minutes



DNS servers are the heart and soul of the Internet. Without them we couldn't resolve hostnames and domain names into IP addresses.



TABLE OF CONTENTS

What's DNS enumeration?

- Impact

Top 7 DNS enumeration tools

- Dig
- Host
- DNSenum
- Nmap
- DNS Recon
- Fierce
- SecurityTrails advanced DNS enumeration

DNS enumeration API

However, DNS is also one of the most frequently attacked protocols, where different types of DNS attacks are spread from home users to small, mid and large companies.

That's why, in the information gathering process, the most common practice is to create a full inventory of all internet-connected devices and domain names from the company you're investigating.

We all know that DNS servers are basically computers connected to the Internet, and that helps us to resolve hostnames into IP addresses. They're in charge of managing and processing DNS requests from clients that need to fetch fresh domain name information, along with DNS records.

That's where the weak link shows up, thanks to the way the DNS was built. It's a bit vulnerable, which allows us to perform DNS enumeration (also known as DNS recon) easily.

But, for those who are just starting out in this OSINT world, let's first find out what DNS enumeration is, and the different techniques and DNS search tools that can help us on this journey.

## What's DNS enumeration?

DNS enumeration is one of the most popular reconnaissance tasks there is for building a profile of your target.

In plain english, it's the act of detecting and enumerating all possible DNS records from a domain name. This includes hostnames, DNS record names, DNS record types, TTLs, IP addresses, and a bit more, depending on how much information you're looking for.

With effective DNS enumeration, you can clone DNS zones manually, using scripts or by exploiting DNS zone transfer vulnerabilities, known as AXFR (Asynchronous Transfer Full Range) Transfer. This latter type of DNS transfer takes place when an attacker detects a misconfigured DNS server that is actually responding to AXFR requests.

## Impact

Once DNS enumeration is completed, unauthenticated users may use this information to observe internal network records, grabbing useful DNS information that provides the attacker access to a full DNS map. This allows him to explore the attack surface area of any company, so he can later scan it, collect data, and—while he's at it—exploit it if there's an open opportunity.

In the past we've seen a bit of DNS enumeration, such as in the How to Find Subdomains article. However, that was only focused on subdomains. Today we'll go one step forward and show you how to perform full DNS enumeration.

# Top 7 DNS enumeration tools

There are plenty of DNS recon and DNS enumeration tools around, from Bash to Python scripts; however, the easiest DNS enumeration can be performed with a single system command.

Let's explore the best ways to perform a DNS enumeration.

## Dig

Once again, our beloved dig command comes to the rescue, helping us perform DNS enumeration by querying popular types of DNS records. Here's how it's done.

To perform a simple domain lookup to fetch A records:

```
dig securitytrails.com +short
```

Expected output:

```
[research@securitytrails.com ~]$ dig securitytrails.com +short
151.139.243.5
[research@securitytrails.com ~]$
```

Now let's grab some mail server information adding -t mx parameters:

```
[research@securitytrails.com ~]$ dig securitytrails.com -t mx +short
10 aspmx2.googlemail.com.
```

```
1 aspmx.l.google.com.
5 alt2.aspmx.l.google.com.
5 alt1.aspmx.l.google.com.
10 aspmx3.googlemail.com.
[research@securitytrails.com ~]$
```

That's right, securitytrails.com uses Google Apps for the email services.

The same goes for NS, CNAME and other records:

```
dig securitytrails.com -t ns +short
```

Output:

```
[research@securitytrails.com ~]$ dig securitytrails.com -t ns +short
ns07.domaincontrol.com.
ns08.domaincontrol.com.
[research@securitytrails.com ~]$
```

Great! With this last command we now have the authoritative name servers.

Of course, dig allows DNS transfers and this leads us to use the AXFR argument against the ns08.domaincontrol.com NS, as you see below:

```
[researcg@securitytrails.com ~]$ dig axfr securitytrails.com ns08.domaincontrol.com
; <<>> DiG 9.11.10-RedHat-9.11.10-1.fc30 <<>> axfr securitytrails.com ns08.domaincontrol.com
;; global options: +cmd
```

```
; Transfer failed.
[research@securitytrails.com ~]$
```

And the transfer failed, because our DNS servers are well secured.

Recon tip: sometimes a particular name server may be configured to reject AXFR requests. However, some others may not—that's why we suggest you launch AXFR queries against all the authoritative NS.

### Host

Host is a command used to resolve the IP address of any given domain name.

As an example, we'll use it to get all the public DNS records from securitytrails.com. See below:

```
host securitytrails.com
```

Expected output:

```
[research@securitytrails.com ~]$ host securitytrails.com
securitytrails.com has address 151.139.243.5
securitytrails.com mail is handled by 5 alt2.aspmx.l.google.com.
securitytrails.com mail is handled by 5 alt1.aspmx.l.google.com.
securitytrails.com mail is handled by 10 aspmx3.googlemail.com.
securitytrails.com mail is handled by 1 aspmx.l.google.com.
securitytrails.com mail is handled by 10 aspmx2.googlemail.com.
```

The default host command retrieves A, AAAA and MX records.

If you want to specify any specific type of DNS record, you can use the -t option. For example:

```
host -t ns securitytrails.com
```

Expected output:

```
[research@securitytrails.com ~]$ host -t ns securitytrails.com
securitytrails.com name server ns08.domaincontrol.com.
securitytrails.com name server ns07.domaincontrol.com.
```

What if you want to grab MX records? Just use -t mx, as shown here:

```
[research@securitytrails.com ~]$ host -t mx securitytrails.com
securitytrails.com mail is handled by 10 aspmx3.googlemail.com.
securitytrails.com mail is handled by 5 alt2.aspmx.l.google.com.
securitytrails.com mail is handled by 10 aspmx2.googlemail.com.
securitytrails.com mail is handled by 1 aspmx.l.google.com.
securitytrails.com mail is handled by 5 alt1.aspmx.l.google.com.
```

The -t option can be used to request any recognized query type such as: CNAME, NS, SOA, TXT, DNSKEY, AXFR, etc. If no query type is specified, host by default will look for A, AAAA and MX records.

Having said that, let's try to perform a full DNS transfer:

```
[research@securitytrails.com ~]$ host -t axfr securitytrails.com ns08.domaincontrol.com
Trying "securitytrails.com"
```

```
Host securitytrails.com not found: 9(NOTAUTH)
; Transfer failed.
[research@securitytrails.com ~]$
```

No luck :), our DNS server doesn't allow AXFR transfers by default.

In the case of a successful DNS transfer, you should be able to get the full DNS zone for the given domain name, as you see below—notice this time we are using -l option, which is another way to list all DNS records from a domain name—while testing the vulnerable site zonetransfer.me:

```
[research@securitytrails.com ~]$ host -l zonetransfer.me nsztm1.digi.ninja
Using domain server:
Name: nsztm1.digi.ninja
Address: 81.4.108.41#53
Aliases:
zonetransfer.me has address 5.196.105.14
zonetransfer.me name server nsztm1.digi.ninja.
zonetransfer.me name server nsztm2.digi.ninja.
14.105.196.5.IN-ADDR.ARPA.zonetransfer.me domain name pointer www.zonetransfer.me.
asfdbbox.zonetransfer.me has address 127.0.0.1
canberra-office.zonetransfer.me has address 202.14.81.230
dc-office.zonetransfer.me has address 143.228.181.132
deadbeef.zonetransfer.me has IPv6 address dead:beaf::
email.zonetransfer.me has address 74.125.206.26
home.zonetransfer.me has address 127.0.0.1
internal.zonetransfer.me name server intns1.zonetransfer.me.
```

```
internal.zonetransfer.me name server intns2.zonetransfer.me.
intns1.zonetransfer.me has address 81.4.108.41
intns2.zonetransfer.me has address 167.88.42.94
office.zonetransfer.me has address 4.23.39.254
ipv6actnow.org.zonetransfer.me has IPv6 address 2001:67c:2e8:11::c100:1332
owa.zonetransfer.me has address 207.46.197.32
alltcpportsopen.firewall.test.zonetransfer.me has address 127.0.0.1
vpn.zonetransfer.me has address 174.36.59.154
www.zonetransfer.me has address 5.196.105.14
[research@securitytrails.com ~]$
```

## DNSenum

DNSEnum is a great script specifically designed for DNS recon activities. Written in Perl, it can help you create a full DNS map of any domain name on the Internet.

Available in most distros, including Ubuntu, Fedora, and of course, Kali Linux, it offers an easy syntax for all who are performing reconnaissance tasks.

What can I do with DNSenum?

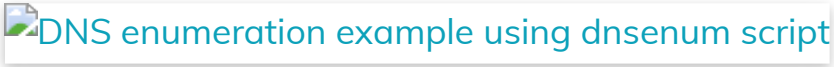
Fetch NS, MX, AXFR and A records, as well as remote BIND version from the DNS server. Apart from that, it also allows you to perform Google scraping using Google dorks such as allinurl: -www site:domain, launch brute force subdomain reconnaissance attacks using word lists, and get a full list of C class domain network ranges. It also allows WHOIS queries on each one of them and reverse lookups against netranges.

How does it work?

Let's look at an example where we'll be avoiding reverse lookup (–noreverse) and saving the output into a file.xml (-o) while querying securitytrails.com:

```
[root@research ~]# dnsenum --noreverse -o file.xml securitytrails.com
```

Expected output:




As shown in the result, DNSenum was able to get A records for the main hostname, as well as NS and MX records, while performing a DNS zone transfer by querying the AXFR record.

DNSenum also allows you to use the Google search engine to "scrape" the results and get a list of subdomains. We'll now combine this with the "-p" argument, which specifies the number of pages searched on Google that will be processed (by default 5 pages). The "-s" option defines the maximum number of subdomains that will be extracted from Google (default is 15).

```
dnsenum --dnsserver ns3.p16.dynect.net github.com -p 10 -s 50
```

## Nmap

Nmap was our #1 choice when we reviewed the best port scanners, but it's really more than that. This time it will help us reveal DNS information from a remote domain name.

By using the dns-brute script, Nmap will attempt to enumerate DNS hostnames by brute forcing popular subdomain names. In this case, we did it against microsoft.com and this was the result:

```
[root@research ~]# nmap -T4 -p 53 --script dns-brute microsoft.com
Starting Nmap 7.70 ( https://nmap.org ) at 2019-10-02 11:56 -03
Nmap scan report for microsoft.com (40.113.200.201)
Host is up (0.21s latency).
Other addresses for microsoft.com (not scanned): 40.76.4.15 104.215.148.63 40.112.72.205 13.77.161.179
PORT STATE SERVICE
53/tcp filtered domain
Host script results:
| dns-brute:
| DNS Brute-force hostnames:
| admin.microsoft.com - 13.107.9.156
| admin.microsoft.com - 2620:1ec:4:0:0:0:0:156
| ads.microsoft.com - 23.100.75.192
| alerts.microsoft.com - 40.112.72.205
| apps.microsoft.com - 23.44.181.123
| id.microsoft.com - 13.107.6.190
| info.microsoft.com - 192.28.149.178
| test.microsoft.com - 104.211.31.212
|_ news.microsoft.com - 192.237.225.141
```

```
Nmap done: 1 IP address (1 host up) scanned in 10363.72 seconds
[root@research ~]
```

While this isn't the most complete DNS recon option, it does help as an alternative source of information during your data collection process.

## DNS Recon

DNSRecon is another great script that can help you discover DNS data from any given domain name.

It allows you to enumerate all types of DNS records, including A, AAAA, SPF, TXT, SOA, NS and MX, and also includes a brute force technique for grabbing subdomain and host A and AAAA records based on a wordlist.

A cool thing we noticed is that it supports checking for cached A and AAAA DNS records on the DNS servers, as well as local DNS enumeration capabilities.

How can I perform DNS exploration with DNSRecon?

The easiest way is by using the -d parameter, as you see below:

```
dnsrecon -d domain.com
```

Here we performed this dns enumeration against linkedin.com, and this was the result:

dns enumeration against linkedin.com

As shown, it was able not only to fetch multiple records (MX, A, AAAA, TXT, SRV and NS), but also to find some exposed bind versions from their DNS servers.

### Fierce

Fierce is another great DNS reconnaissance tool. Written in Perl, Fierce offers numerous options for performing DNS enumeration by scanning domains in just minutes. Its syntax is pretty easy, as you can see:

```
fierce -dns securitytrails.com
```

Expected output:

```
root@securitytrails-kali:~# fierce -dns securitytrails.com
DNS Servers for securitytrails.com:
ns08.domaincontrol.com
ns07.domaincontrol.com
Trying zone transfer first...
Testing ns08.domaincontrol.com
Request timed out or transfer not allowed.
Testing ns07.domaincontrol.com
Request timed out or transfer not allowed.
Unsuccessful in zone transfer (it was worth a shot)
Okay, trying the good old fashioned way... brute force
Checking for wildcard DNS...
Nope. Good.
Now performing 2280 test(s)...
```

```
151.139.243.5 blog.securitytrails.com
151.139.243.5 ftp.securitytrails.com
Subnets found (may want to probe here using nmap or unicornscan):
151.139.243.0-255 : 2 hostnames found.
Done with Fierce scan: http://ha.ckers.org/fierce/
Found 2 entries.
Have a nice day.
root@securitytrails-kali:~#
```

Fierce was able to discover a few subdomains, along with NS records, and attempted to run a DNS transfer, which obviously failed.

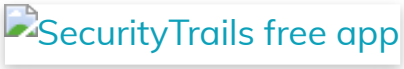
By adding the –wide, you can also extend the fierce scan to the entire class C after finding any matching hostnames in that class C. This can take a lot of time to finish, especially on networks with a lot of hosts, so keep that in mind.

### SecurityTrails advanced DNS enumeration

DNS record detection is one of the core features in all of our products, from the free app to our passive DNS API to our enterprise-grade products.

In particular, our free app offers great results when it comes to building a DNS map of all possible DNS records from a given domain name, as shown here:

For this purpose, the SecurityTrails free app offers both current and historical DNS records for A, AAAA, NS, MX, SOA and TXT records, letting you build an existing DNS structure quickly.




It also includes full subdomain enumeration capabilities, which allows you to discover all existing subdomains for the target domain. In this case, linkedin.com showed that it has around 1235 subdomains, quite an extensive list.

Where other DNS enumeration tools fail to retrieve all the existing subdomains, or require to launch brute-force techniques, our intelligence tool does it all, in a matter of seconds.


However, if you're ready to jump into the real thing, that's where our enterprise-grade SurfaceBrowser product comes in.

SurfaceBrowser is no mere DNS exploration tool, but a complete OSINT utility that will give you a look at any attack surface area by combining DNS information, IP addresses, associated domains, SSL certificates and SSL ports.

Unlike the free app, SurfaceBrowser doesn't limit the amount of DNS information, and gives you instant access to full DNS enumeration like no other tool on the Internet.

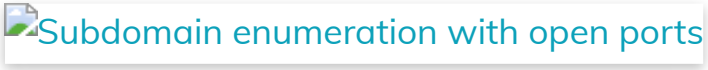
Want to see how it works? Check it out:

- First, log in from https://securitytrails.com/app/sb/ (if you don't have an account, schedule a demo with our sales team).
- Now, analyze a domain—let's say sputniknews.com, a popular russian news service.

Let's look at the result:


First things first: you'll see the most important options to choose from, whether you want a Summary by Hosting company, Summary by IP, or by Open Ports.

From the SurfaceBrowser interface you can get the full list of DNS records, including subdomains, along with open ports on each one of them. With this information you don't need to perform a full port scan, as the results are already there. And if you need it, you can download all the data with just a few clicks for offline analysis.




Now ask yourself this: What if you could do more than a current DNS enumeration? What if you could detect present DNS records, while drawing an entire picture of past A, AAAA, MX, NS, SOA and TXT DNS records? That's where SurfaceBrowser does its magic:

Our historical DNS records are able to bring you the latest DNS records as well as the oldest ones, so you can browse all the changes over the years.

## DNS enumeration API

If you don't want to rely on third party tools, and you have your own developers who can write code, then there's another way to perform DNS enumeration—by querying a DNS enumeration API.

Our passive DNS API allows you to fetch full DNS record information from any domain name in the world, and in the same manner as SurfaceBrowser, the data is available in an instant.

How does it work?

Just sign up to get your free API key. Once you've done that, you can query our API data by launching a simple HTTP request. Here's an example with curl from the terminal:

```
curl --request GET \
--url 'https://api.securitytrails.com/v1/history/oracle.com/dns/a?apikey=api_key_here''
```

Of course, our API supports a wide range of programming languages, such as PHP, Python, Javascript, Go, NodeJS and much more. This is another example of full DNS historical records, this time using Python:

```
import requests
url = "https://api.securitytrails.com/v1/history/oracle.com/dns/a"
querystring = {"apikey":"your_api_key"}
```

```
response = requests.request("GET", url, params=querystring)
print(response.text)
```

And if you need to perform a full subdomain enumeration, it's easy with the list-subdomains endpoint:

Subdomain DNS search using a passive DNS API

## Conclusion

DNS enumeration can be easily performed with terminal-based commands or web-based interfaces.

From an attacker's point of view, finding DNS records is a critical part of the intel-reconnaissance process. It's found in all penetration testing activities.

And from a blue team perspective, knowing what's behind your DNS servers is a valuable piece of information for reducing your attack surface area before the bad guys start chasing you.

---

Get access to the ultimate enterprise-grade DNS reconnaissance tool and explore the #1 cyber security OSINT toolkit: SurfaceBrowser™. Book a demo with our sales team today!

## Sign up for our newsletter!

Email

name@company.com

**Subscribe**

‹ PREVIOUS

## Related Posts

### What's My DNS? How to Find Domain DNS Records with our DNS Checker

Exploring the best ways to answer the question: What's my DNS?

### Cybersecurity Red Team Versus Blue Team — Main Differences Explained

We've previously explored the Top 20 OSINT Tools available, and today we'll go through the list of top-used Kali Linux software.

### Cybersecurity Fingerprinting Techniques and OS-Network Fingerprint Tools

We explore what a fingerprint is in cyber security, different types of

fingerprint techniques, and some of the most popular fingerprinting tools in use.