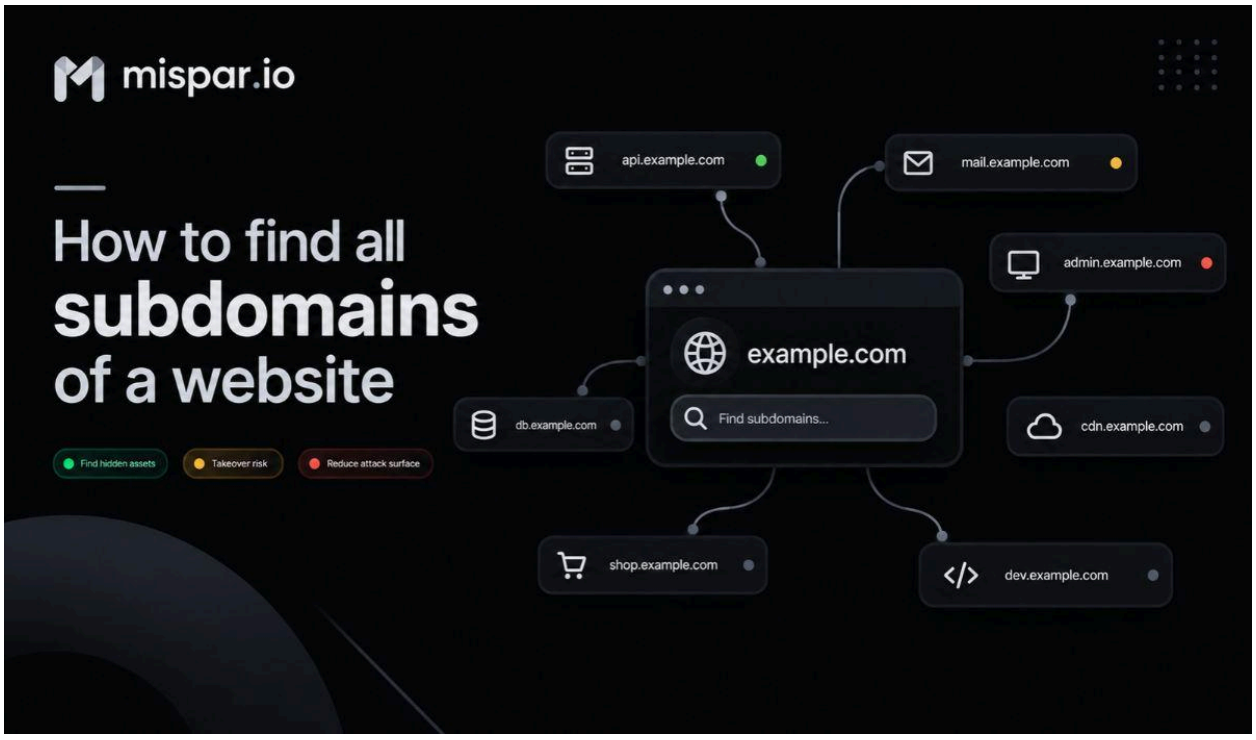
How to Find All Website Subdomains | A DNS-First Guide

To find all subdomains of a website, start with the data DNS and certificates already made public. A subdomain finder collects that data in seconds. You then check the domain's name servers, search for older traces and test which hosts still answer. Together, these steps give you a list you can trust far more than any single tool.Every domain collects extra hostnames over time. A product launch adds one. A developer's test build adds another. A vendor integration adds a third and nobody removes them when the project ends. Knowing how to find all subdomains of a website shows you what is really reachable from the internet, which is the first step in securing it.



This guide is written for MSSPs, MDR providers, penetration testers, bug bounty hunters and IT security teams. It takes a DNS-first view, because DNS is where every subdomain begins.

What Is a Subdomain Finder?

A subdomain finder is a tool that discovers the subdomains of a root domain by reading public sources, mainly DNS records and certificate transparency logs. It returns a list of hostnames,

such as a staging site or a support portal. Security teams use that list to see how much of the organization is exposed online.

A subdomain is a label placed in front of a main domain. Each label creates a new hostname that can point to its own server or cloud service. Some labels lead to busy production systems. Others lead to forgotten pages that nobody maintains.

The same job goes by several names, including subdomain scanner, subdomain lookup, subdomain checker and subdomain enumeration. The aim never changes: the fullest list of hostnames you can build for a single domain.

How a Subdomain Finder Works

A subdomain finder pulls hostnames from several public places, merges them and removes duplicates. Some sources hold data that already exists, which is passive collection. Others need the tool to send queries, which is active collection. Strong tools lean on passive data first.

Passive Collection

Passive collection reads what other systems have already recorded. Examples include certificate transparency logs, search engine results, internet-wide scan data and web archives. The tool never contacts the target's servers, so this is the safest way to begin reconnaissance.

Active Collection

Active collection asks DNS about guessed names. The tool takes a wordlist of common labels, adds each one to the root domain and checks whether it resolves. This is brute-forcing. It can reveal hosts that have no public record lists, but it creates traffic, so run it only on domains you own or have written permission to test.

Certificate Transparency Logs

Certificate transparency logs are public records of issued TLS certificates. Since each certificate names the hostnames it covers, the logs work as a steady source of subdomains, including old ones that still appear in the record.

Name Servers and Zone Data

Every domain has authoritative name servers that hold its zone, the full set of DNS records. Name servers copy that data to each other with a process called a zone transfer. Understanding this process matters for subdomain discovery, because a misconfigured server that shares its zone with anyone gives away every hostname in one request.

Wildcard DNS

Wildcard DNS makes every possible name under a domain resolve to one answer. It confuses guessing tools, because every guess looks like a match. A reliable finder checks for a wildcard first and filters those answers.

Where Every Tool Stops

No finder sees everything. A host with no public certificate, no public DNS trail and an unusual name can stay out of sight. The Mispar tool page says so plainly, calling the list a starting point rather than a full audit of your attack surface.

How to Find All Subdomains of a Website: Step by Step

The most dependable method is a short, repeatable workflow. Confirm your permission, run a baseline scan, search certificate logs, review the name servers and DNS records, check search engines and archives, then validate every name you found.

Step 1: Confirm Scope and Approval

List the root domains you are covering and who approved the work. Reading public data is routine. Active testing against systems you do not manage can create legal and contractual problems, so get written approval first and file it with your project notes.

Step 2: Run a Baseline Scan

A free online subdomain finder is the quickest answer to how to find all subdomains of a website at the start, because it shows the visible footprint in seconds.

The Mispar Subdomain Scanner is one option. According to its tool page, it checks public DNS records, certificate logs and known naming patterns. It is free, runs in a browser and needs no signup. The page also states that discovery uses public DNS records only and does not touch the target server directly and it asks users to get authorization before scanning infrastructure they do not manage.

Save the output. Every later step adds to it or confirms it.

Step 3: Search Certificate Logs

Open a certificate log search site such as crt.sh and search the root domain with a wildcard. You will see certificates issued for names under the domain, including old ones. Read the main name and the alternative names on each certificate, then collect the unique hostnames.

Two gaps apply. A wildcard certificate hides individual names behind one entry. Private certificates never reach public logs at all.

Step 4: Review Name Servers and DNS Records

Find the domain's name servers, then read its records closely. Address records, CNAME records, mail records and text records all hold hostnames. A CNAME may show that a team linked a subdomain to an outside platform. A text record may name a service the company uses.

Then try a zone transfer against each name server, but only where your scope allows it. Most servers refuse and refusal is the right behavior. If one agrees, you have found a real misconfiguration and probably the complete host list. Report it.

Step 5: Check Search Engines and Archives

Search engines index pages that the main site never links to. Use the site operator with a wildcard before the root domain, then exclude names you already know so that new ones appear. Repeat until nothing changes.

Web archives add history. The Wayback Machine can show subdomains retired long ago that may still have DNS entries. Public code repositories also help, because developers sometimes commit configuration files that mention internal hostnames.

Step 6: Add Passive Automation

Tools such as Subfinder, Amass in passive mode, Sublist3r and Assetfinder query many sources at once. The OWASP Web Security Testing Guide names Sublist3r and Amass among its enumeration tools. Run at least two, because their sources differ. Several of them return more when you add API keys.

If your scope allows active testing, finish with a wordlist run using a tool like Gobuster or ffuf. Test for wildcard DNS first and keep the query rate low.

Step 7: Validate and Triage

Merge all lists, remove duplicates and probe each name to see whether it resolves and responds. A tool like https does this in bulk. For each live host, ask who owns it, what it runs, whether it is still needed and whether its DNS record points to a service that has been shut down.

Finding Subdomains in Specific Setups

The method is the same everywhere. Tools and menus differ.

On Windows, nslookup and the Resolve-DnsName command in PowerShell handle basic record checks and the Windows Subsystem for Linux runs the Linux tools. In Kali Linux, the

package repositories include many reconnaissance tools, though names and versions change over time.

For a domain you own, your DNS dashboard is the best source. In cPanel, look in the Domains area. In GoDaddy, open the DNS management page. In Cloudflare, open the DNS Records tab. A dashboard only shows what it manages, so compare it against the outside methods.

WordPress does not control DNS, so a standard install has no subdomain list in the admin area. Your DNS provider or host holds that data. On a multisite network that uses subdomains, the Sites screen in Network Admin lists each site. For small custom checks, a short Python script with the dnspython library can test common labels against a root domain.

Why Security Teams and MSSPs Need a Subdomain Finder

Security teams need a subdomain finder because exposure begins with hosts nobody tracks. It reveals forgotten servers, shadow IT and takeover candidates early. For MSSPs, it also turns a vague risk conversation into a concrete list of a client's public hosts.

- Attack surface mapping. Each subdomain is a possible entry point, so a complete list is the base of any exposure review.
- Shadow IT discovery. Teams sign up for outside services and connect subdomains without telling security.
- Client onboarding. An outside-in asset list is one of the fastest deliverables for a new MSSP client.
- Testing scope. Penetration testers and bug bounty hunters use subdomain lists to widen coverage, within program rules.

Key Features to Look For in a Subdomain Finder

A strong subdomain finder uses several public sources, admits its limits and makes results easy to review. Use this list to compare tools.

- Several sources. Certificate logs, DNS data and naming patterns together beat any one alone.
- Passive by design. A tool that avoids direct contact with the target is safer for early reconnaissance.
- Honest coverage. Be careful with any claim to find every subdomain.
- Low friction. A browser-based tool with no signup suits quick checks and the Mispar page states that its scanner needs none.
- Wildcard handling. The tool should spot wildcard DNS and filter false names.
- Liveness hints. Knowing which names respond saves triage time.

Comparison Table: Ways to Find Subdomains

The table compares the main options. Passive options read existing data, active options send queries, manual options suit one-off work and automated options suit repeat work. Most teams combine several.

Option	Type	Effort	Best For	Main Limit
Web-based scanner (Mispar Subdomain Scanner)	Public-record discovery	Very low	Fast baseline; free, online, no signup	Public records and common patterns only, so unusual unlisted names may not appear
Certificate log search	Passive, manual	Low	Names from issued certificates	Wildcard and private certificates hide names
Search engines and archives	Passive, manual	Low to medium	Forgotten pages and retired hosts	Shows only indexed or archived content
DNS record review	Passive, manual	Medium	Links to third-party services	Shows only names present in readable records
Zone transfer attempt	Active, manual	Low	Testing name server configuration	Well-run servers refuse the request
Passive automation tools	Passive, automated	Medium	Repeatable multi-source runs	Setup time; some sources work better with API keys
Wordlist brute force	Active, automated	Medium	Names that appear in no public record	DNS traffic, wildcard false hits, needs authorization
Commercial attack surface platform	Mixed, automated	Varies	Ongoing monitoring across many domains	Cost and coverage vary by vendor

Treat the table as a menu. A web scanner, a certificate search and one passive tool will show most of what public data holds. Add active tests only when you are cleared and need more depth.

Zone Transfers: The Shortcut That Rarely Works

A zone transfer is the one method that can hand you every hostname in a single request, which is why testers always try it. In practice, well-run name servers refuse. A server that is unable or unwilling to share its zone returns an error code and most return the standard refusal.

That refusal is good news. Sites that do not want to disclose their full zone data can restrict transfers to authorized secondary servers and signing the transfers with a shared key adds proof of who is asking. If a server does answer, report it as a misconfiguration, because anyone could copy the zone.

Interesting Facts and Key Stats

These facts come from IETF documents, OWASP guidance and the Mispar tool page. They are stated in general terms, with no invented figures.

- RFC 5936, published by the IETF in 2010, defines the DNS zone transfer protocol known as AXFR, the standard way servers copy a full zone.
- An IETF draft that clarified AXFR notes that the protocol gives read-only public access to complete zone data and that sites wanting to avoid disclosure may restrict transfers to authorized secondary servers.
- The same draft says that a server unable or unwilling to provide a transfer must reply with an error code other than the no-error code and it treats a closed connection as equivalent to a refusal.
- The draft also recommends using TSIG, a shared-key signing method, to authenticate transfers.
- RFC 9103 explains that zone transfers travel in clear text, which lets an eavesdropper collect a zone's contents and that TSIG limits who can request a transfer but adds no encryption. It defines zone transfer over TLS to fix that.
- The Mispar tool page states that the scanner is free, needs no signup and uses public DNS records only, without touching the target server directly.

Conclusion

To find all subdomains of a website, start with public DNS and certificate data, then widen the search. Use a subdomain finder for a quick baseline, add certificate logs, review name servers and DNS records, check search engines and archives, run passive tools and test guesses only when you are authorized. Then probe what is live and review anything unfamiliar.No method gives a perfect list and honest tools admit it. What matters is the routine: scan regularly, compare against the last run and retire what nobody needs. For MSSPs, that routine turns a plain hostname list into repeatable value for every client. To learn how Mispar supports MSSPs, visit the main site.

Frequently Asked Questions (FAQs)

What is the quickest way to find all subdomains of a website?

Run a web-based subdomain finder first, because it returns a baseline in seconds. Then add a certificate log search and one passive tool. Together they cover most of what public data shows.

Why do most DNS zone transfers fail?

Most name servers are set to refuse transfer requests from anyone except authorized secondary servers. That is the secure setup. If a transfer works for a stranger, it is a misconfiguration worth reporting.

Is a free subdomain finder reliable?

A free subdomain finder is reliable for names in public records, such as DNS data and certificate logs. It can miss unusual unlisted names, so confirm important results with a second source.

What is the difference between passive and active enumeration?

Passive enumeration reads data that already exists, without contacting the target. Active enumeration sends DNS queries to test guessed names. Passive is safer to start with and active needs authorization.

Why can a wildcard DNS record cause false results?

A wildcard makes every possible name resolve to the same answer, so a guessing tool thinks every name exists. Test first with a name that cannot exist and filter out any matching answers.

What should I do once I have the list?

Check each host against what your team expects. Find the owner, confirm it is still needed and look at where its DNS record points. Retire unused hosts and delete records for services that no longer exist.