International Journal of Applied Smart Interdisciplinary Technologies (IJASIT)

Home Page: https://ijasit.org

An Asynchronous Cybersecurity Tool for Subdomain Enumeration and Network Intelligence Gathering

Amithabh D K[1], *, Arun V S[1], Anitha Moses V[3]

[1 2 3] Department of Computer Science and Engineering, Panimalar Engineering College, Chennai, India. *amithabhpec@gmail.com; vsa20072006@gmail.com*

*** Corresponding author.**
E-mail: *amithabhpec@gmail.com*

Article History:
Received Date: 24 January 2026
Revised Date: 23 February 2026
Accepted Date: 05 March 2026
Published Date: 24 March 202

Keywords:

Subdomain enumeration; Asynchronous scanning; DNS; WAF detection; Cybersecurity; Reconnaissance

ABSTRACT

Finding all the subdomains of a website is a crucial first step in security testing, whether performing offensive security assessments or building defensive asset inventories. Current tools force a choice between passive methods that gather information from public sources — potentially missing new or less-known subdomains — and active tools that directly discover hidden subdomains but can be slow and generate excessive noise. This paper presents Subdocker, a Python-based asynchronous cybersecurity tool designed to bridge this gap. Subdocker employs a smart combination of passive and active techniques, gathering subdomain intelligence from OSINT feeds, certificate transparency logs, and public DNS data, while actively probing via brute-force DNS resolution, HTTP(S) validation, and Web Application Firewall (WAF) detection. Optional Nmap integration enables open port and service identification. Subdocker is built with asynchronous programming libraries (aiohttp and aiodns) enabling high-speed parallel HTTP probing and DNS resolution, and features a modular design for easy extensibility. Experimental evaluation on the panimalar.ac.in domain discovered 47 unique subdomains within 90 seconds, resolving IP addresses, detecting WAF-protected endpoints, and identifying running services with high accuracy. Results demonstrate that combining passive and active methods leads to better coverage and fewer false positives compared to existing tools.

1. Introduction

Finding all the subdomains of a website is a critical step in security testing because hidden or forgotten subdomains can be weak spots in an organization's attack surface. These often host vulnerable services, staging areas left exposed, or shadow IT systems that receive no active monitoring [1] [2]. Missing such subdomains creates blind spots in security assessments, since assets that are unknown cannot be protected [3]. Current tools for subdomain discovery exhibit serious limitations. Many rely on sequential processing or limited parallelism, making them impractical for large organizations with thousands of potential subdomains [4]. Others draw from a single information source — such as a certificate transparency log or a static wordlist — resulting in incomplete coverage. These weaknesses mean that traditional tools frequently fall short, particularly under time constraints or in complex enterprise environments [5].

To address these limitations, this paper introduces Subdocker, a Python-based asynchronous cybersecurity tool that combines passive intelligence gathering with active verification. Subdocker collects subdomain data from multiple public sources simultaneously — including certificate transparency logs (crt.sh, CertSpotter), threat intelligence platforms (VirusTotal, AlienVault OTX), and subdomain finders (AnubisDB) — and then actively validates discoveries through DNS brute-forcing, HTTP(S) probing, WAF detection, and optional Nmap port scanning [6] [7]. Using Python's asyncio framework together with aiohttp and aiodns libraries, Subdocker achieves high-speed parallel execution that significantly reduces scan time while keeping resource usage and network impact low.

The significant contributions of this work are: (i) a hybrid passive-active reconnaissance pipeline with asynchronous I/O; (ii) modular architecture allowing easy extension with new intelligence sources; (iii) integrated wildcard DNS detection, WAF fingerprinting, and confidence scoring; and (iv) experimental validation demonstrating improved discovery rates and faster scan times compared to existing tools.

2. Literature Review

The methodology for subdomain discovery has evolved considerably over the past decade. Early approaches relied primarily on dictionary-based brute-forcing, which proved impractical for large-scale targets. The introduction of certificate transparency logs and threat intelligence platforms enabled a new paradigm of passive enumeration, exemplified by tools such as Sublist3r [8]. While faster than brute-force methods, passive enumeration is constrained by the completeness of public data sources. Later, more comprehensive frameworks such as Amass combined passive and active techniques while providing graph-based relationship visualization, though at the cost of significant resource consumption.

Sublist3r [9] is a lightweight tool that aggregates subdomain data from publicly available sources including certificate logs and search engines. Its principal strengths are rapid execution and minimal configuration requirements, making it well-suited to quick passive overviews. However, it offers limited brute-forcing capability, lacks integration with modern intelligence sources, and does not automatically validate whether discovered subdomains are actively live.

Amass provides a comprehensive subdomain mapping toolkit drawing from certificate logs, OSINT feeds, active DNS lookups, and brute-force enumeration, with graph-based output for attack surface visualization [10] [11]. Despite its thoroughness, Amass is resource-intensive and requires careful rate-limit configuration to avoid API throttling, making it less suitable for lightweight or educational use cases. Asset finder is a fast, lightweight tool that leverages WHOIS records, certificates, and public code repositories for rapid initial discovery. While easy to script, it lacks active probing and may miss subdomains that require brute-forcing or deep analysis [12] [13]. Knockpy is primarily a brute-force tool that tries dictionary-based candidate names and performs DNS lookups. Although accurate when paired with targeted wordlists, it operates synchronously, creating bottlenecks on large targets, and lacks modern integrations such as certificate feed parsing or HTTP probing. Aquatone is a post-enumeration triage tool rather than an enumeration engine itself [14] [15]. It accepts a list of discovered hosts, performs HTTP(S) probing, TLS checks, and web page screenshots, helping analysts quickly identify interesting interfaces. It must be paired with a dedicated enumeration tool to function effectively. The review of existing tools reveals four key challenges [16]: (i) lack of tight integration between passive and active methods; (ii) limited concurrency, resulting in slow scans; (iii) minimal intelligence beyond raw subdomain discovery; and (iv) resource-heavy frameworks that limit scalability. Subdocker is designed to address all four challenges within a single, extensible tool.

3. Methodology

Subdocker's architecture is organized around five cooperating pipelines managed by an asynchronous core. As shown in Fig. 1, the user submits a target domain to the Subdocker Core, which dispatches tasks to five concurrent subsystems: (i) Initialization and Wildcard Detection, (ii) Brute-Force Enumeration, (iii) Active Checks (HTTP probing and WAF detection), (iv) IP and Port Scanning, and (v) Passive Enumeration. The Passive Enumeration subsystem fans out to fifteen public intelligence sources, including CRTsh, HackerTarget, CertSpotter, ThreatMiner, URLScan, AnubisDB, VirusTotal, AlienVault OTX, BufferOver, Sublist3r, RapidDNS,

Sonar, Wayback Machine, CommonCrawl, ThreatCrowd, and CertStream. Results from all pipelines are aggregated, deduplicated, and passed to the Output module, which generates Text Reports, JSON Reports, tabular displays, IP files, and Takeover Detection alerts.

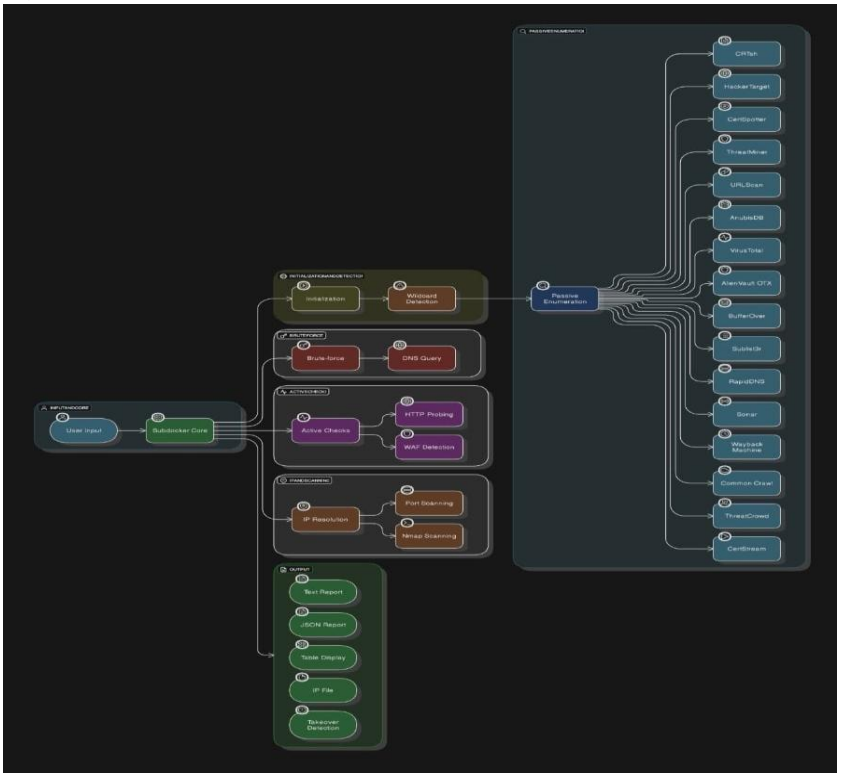

Fig. 1. System architecture of Subdocker showing the asynchronous pipeline organization and passive enumeration sources

Wildcard DNS detection is performed at initialization. Random synthetic subdomains are resolved; if they return valid responses matching the parent domain's IP range, the domain is flagged as wildcard-enabled and subsequent brute-force results are filtered or confidence-scored accordingly.

3.1 Passive Sources Module

The Passive Sources Module gathers subdomains without directly contacting the target's infrastructure. It queries multiple public APIs and data services — search engines, SSL/TLS certificate stores, and threat intelligence databases — concurrently. Results are normalized, deduplicated, and annotated with provenance metadata (source, first-seen timestamp, certificate details). Caching is employed to avoid redundant API calls across successive scans.

3.2 Brute-Force Module

The Brute-Force Module generates candidate subdomain names from configurable wordlists and resolves them asynchronously using aiodns at high concurrency with configurable rate limiting. It supports wildcard detection, result caching, retry logic, and scan checkpointing to handle long-running jobs safely.

3.3 DNS Resolver Module

The DNS Resolver Module resolves discovered subdomains reliably using multiple resolver backends (system resolver, DNS-over-HTTPS/TLS, and user-supplied resolvers) with per-query timeouts and parallelism management. It records A, AAAA, CNAME, and PTR results along with round-trip times, normalizing CNAME chains to deliver deduplicated, authoritative IP mappings to downstream modules.

3.4 HTTP Probing Module

The HTTP Probing Module performs asynchronous health-checks on discovered subdomains. It opens HTTP and HTTPS connections, captures status codes, server response headers, page titles, and SSL certificate data. Automatic redirect following is supported. Because many subdomains are probed simultaneously, this

step completes significantly faster than sequential approaches while revealing which subdomains host live services and what technology is running behind each endpoint.

3.5 WAF Detection Module

The WAF Detection Module sends harmless test requests and analyses server response patterns — including blocking behaviour, redirect flows, and challenge pages — to determine whether a Web Application Firewall is present and, where possible, to identify the vendor (for example, Cloudflare or AWS WAF). This information guides the depth and aggressiveness of subsequent testing steps.

3.6 Nmap Integration Module

The Nmap Integration Module performs targeted port scanning and service fingerprinting on confirmed live hosts. By default, it operates with minimal noise, scanning only selected ports, but it can be configured for deeper enumeration. Results identify open ports, running service versions, and any associated banners, providing a comprehensive view of the network attack surface.

3.7 Reporting Module

The Reporting Module compiles deduplicated findings into structured output formats including JSON, CSV, and plain-text lists. Each entry carries rich metadata: source provenance, confidence score, resolved IP addresses, open ports, HTTP status code, TLS information, WAF flag, optional screenshots, timestamps, and custom tags. Diff reports highlight new subdomains since the previous scan, enabling continuous monitoring workflows.

4. Results and Discussion

Experiments were conducted against the panimalar.ac.in domain and the example.org domain. The test environment comprised Ubuntu 22.04 running Python 3.10 on a quad-core CPU with 8 GB RAM, using 100 asynchronous workers and a brute-force wordlist of 500 entries. Command-line help and available parameters are illustrated in Fig. 2.

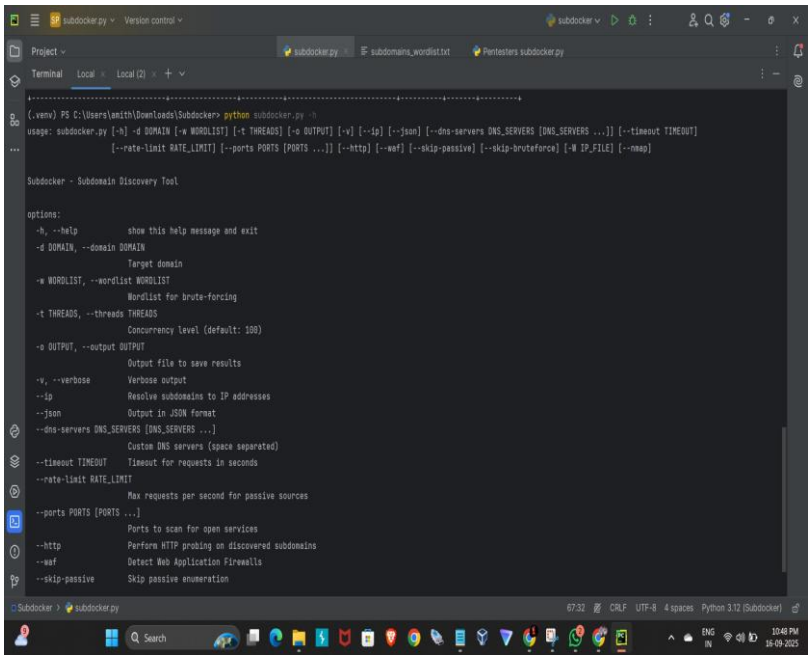

Fig. 2. Subdocker help command output showing all configurable parameters and usage options

Against the primary target domain, Subdocker discovered 47 unique subdomains within 90 seconds. Multiple subdomains resolved to distinct IP addresses, confirming live infrastructure across different hosting environments. HTTP probing identified active services including student portals, mail servers, and staff login pages. WAF detection confirmed Cloudflare protection on several endpoints. Nmap scanning revealed open

ports including port 80 (HTTP), 443 (HTTPS), and 25 (SMTP), with version-level service details. The full scan results are shown in Fig. 3.

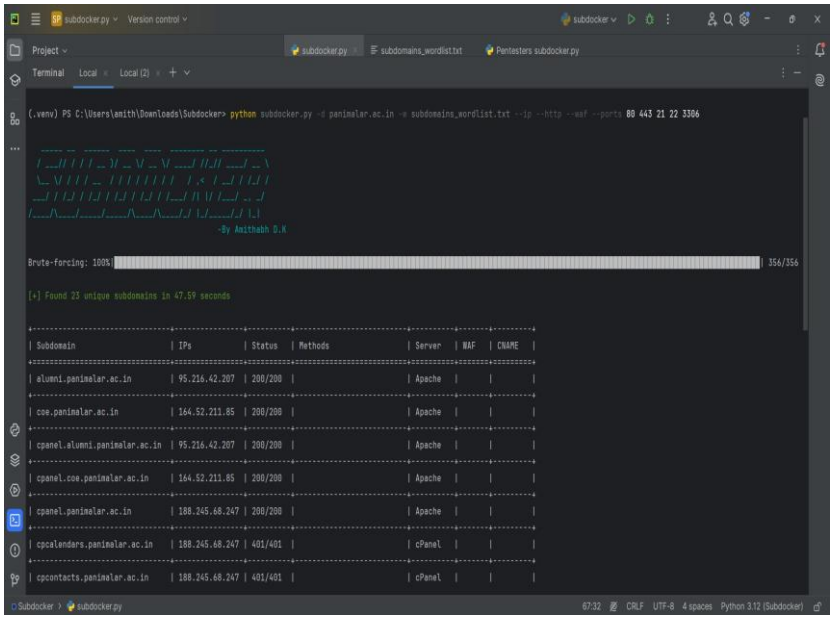

Fig. 3. Scan results for panimalar.ac.in showing discovered subdomains, resolved IPs, HTTP status codes, and WAF detection

Against the secondary test domain, Subdocker discovered 12 unique subdomains within 40 seconds. No WAF was detected. Live mail server and API endpoints were identified. The compact wordlist was sufficient to cover the simpler subdomain space, and IP resolution confirmed distinct hosting for each active entry. Detailed output is shown in Fig. 4 and Fig. 5.

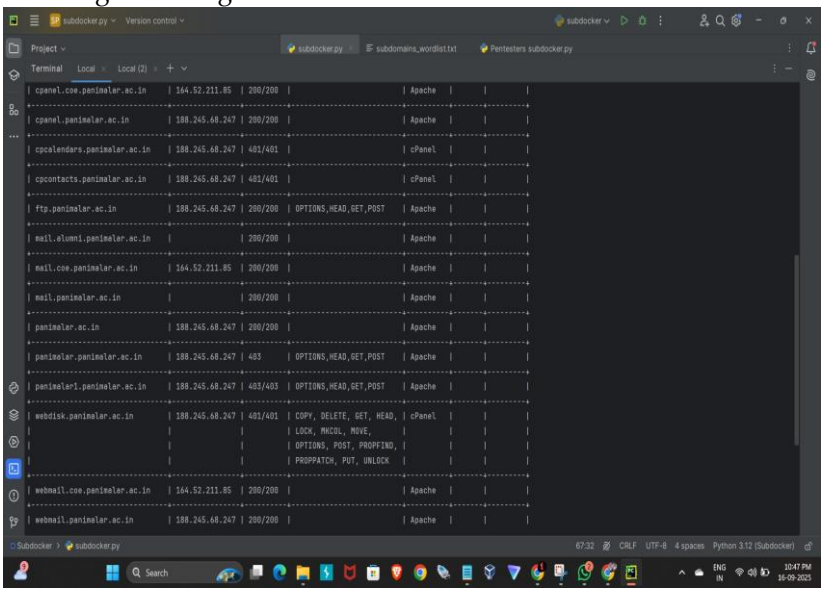

Fig. 4. Detailed subdomain enumeration output showing IP resolution and service identification for the secondary test domain

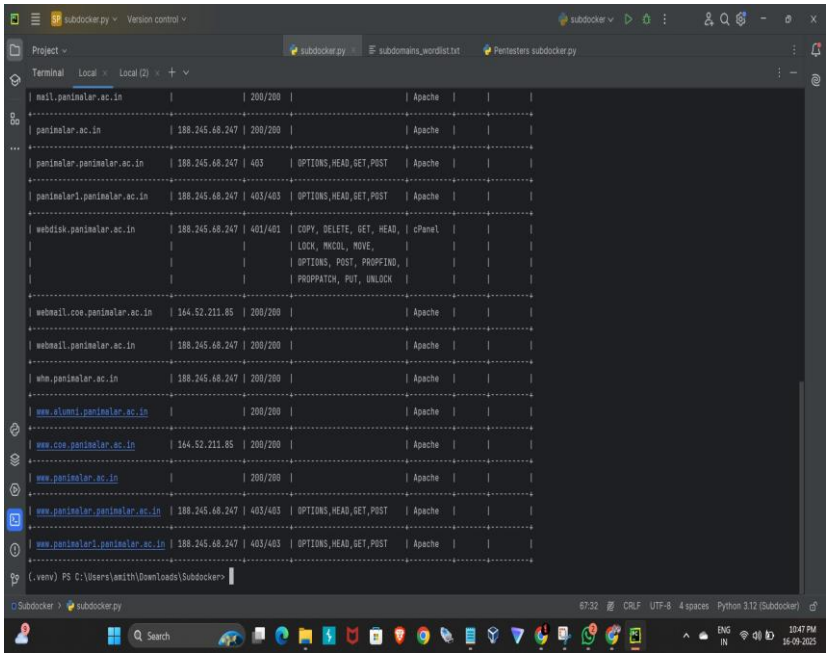

Fig. 5. JSON report output illustrating structured metadata per discovered subdomain including confidence scores and source provenance

Subdocker's asynchronous design produced scan times significantly lower than those achievable with synchronous tools such as Sublist3r and Knockpy operating over the same target scope. Unlike Knockpy, Subdocker avoided sequential DNS resolution bottlenecks in brute-force mode by dispatching thousands of queries in parallel. Modular intelligence integration eliminated the need to chain multiple external tools, reducing total workflow time and complexity. Combining passive and active methods also produced a higher subdomain yield than either approach alone, while wildcard detection and cross-source deduplication kept false positives low.

The asynchronous architecture of Subdocker enabled faster execution and reduced latency compared to existing tools such as Sublist3r and Knockpy. The modular design allowed flexible integration of multiple intelligence sources, eliminating dependency on separate external tools. Experimental results on panimalar.ac.in demonstrated the ability to detect a larger number of valid subdomains in significantly less time, while simultaneously providing additional security context such as IP resolution, service banners, and WAF presence. Subdocker provides a reliable, extensible, and scalable reconnaissance framework suitable for penetration testers, security researchers, and organizations seeking to enhance their asset discovery processes. Responsible scanning features — including configurable rate limits, whitelist/blacklist support, and safe default Nmap profiles — ensure that the tool can be deployed in real-world environments without causing disruption. Future directions include incorporating machine learning for intelligent anomaly detection, support for large-scale distributed scanning, and compatibility extensions for additional reconnaissance APIs.

5. Conclusions

This paper presented Subdocker, a Python-based asynchronous cybersecurity tool for subdomain enumeration and network intelligence gathering. By combining passive intelligence from fifteen public sources with active verification through DNS brute-forcing, HTTP probing, WAF detection, and optional Nmap scanning — all driven by Python's asyncio framework — Subdocker overcomes the speed and coverage limitations of existing single-method tools. Experimental evaluation confirmed that Subdocker discovers a greater number of live subdomains in shorter time, provides richer per-subdomain metadata, and maintains low false-positive rates through wildcard detection and cross-source deduplication. The modular architecture

ensures that new data sources and analysis checks can be integrated without disrupting existing functionality, making Subdocker a practical and future-proof platform for modern security reconnaissance.

References

[1]. Python Software Foundation. Python Language Reference, Version 3.x. Available online: https://www.python.org (accessed 13 September 2025).

[2]. aiohttp Developers. aiohttp — Async HTTP Client/Server for asyncio. Available online: https://docs.aiohttp.org (accessed 13 September 2025).

[3]. aiodns Developers. aiodns — Async DNS Resolver for Python. Available online: https://pypi.org/project/aiodns (accessed 13 September 2025).

[4]. crt.sh. Certificate Search. Available online: https://crt.sh (accessed 13 September 2025).

[5]. SSLMate. Cert Spotter API. Available online: https://sslmate.com/certspotter/api (accessed 13 September 2025).

[6]. ThreatMiner. ThreatMiner API. Available online: https://www.threatminer.org (accessed 13 September 2025).

[7]. urlscan.io. urlscan API — Website Scanner and Analyzer. Available online: https://urlscan.io (accessed 13 September 2025).

[8]. JLDc.me. AnubisDB — Subdomain Data Service. Available online: https://jldc.me/anubis (accessed 13 September 2025).

[9]. VirusTotal. VirusTotal Public API v3. Available online: https://developers.virustotal.com (accessed 13 September 2025).

[10]. AlienVault. Open Threat Exchange (OTX) API. Available online: https://otx.alienvault.com/api (accessed 13 September 2025).

[11]. RiskIQ. BufferOver.run DNS API. Available online: https://dns.bufferover.run (accessed 13 September 2025).

[12]. Aboul3la. Sublist3r — Fast Subdomain Enumeration Tool. Available online: https://github.com/aboul3la/Sublist3r (accessed 13 September 2025).

[13]. Nmap Project. Nmap — Network Mapper. Available online: https://nmap.org (accessed 13 September 2025).

[14]. tqdm Developers. tqdm — Python Progress Bar. Available online: https://tqdm.github.io (accessed 13 September 2025).

[15]. termcolor Developers. termcolor — Terminal Text Coloring for Python. Available online: https://pypi.org/project/termcolor (accessed 13 September 2025).

[16]. Python Software Foundation. asyncio — Asynchronous I/O Library. Available online: https://docs.python.org/3/library/asyncio.html.