Operation Iron Veil: Hardened & Sharpened Subdomain Enumerator

Presented by: Karim Jaber

Date: July 11, 2025

Agenda

- Introduction to the Tool
- Key Features
- How It Works: Core Phases
- Installation and Dependencies
- Usage Guide with Examples
- Comparison to Other Tools (e.g., Amass)
- Ethical Considerations and Limitations
- Future Improvements
- Demo Highlights
- Q&A

Introduction to the Tool

What is Operation Iron Veil?

A Python-based, militarized subdomain enumeration and verification tool designed for security researchers, penetration testers, and bug bounty hunters.

Purpose:

Maximize subdomain discovery while minimizing false positives. Combines passive OSINT, active DNS techniques, probing, scanning, and recursion.

Key Goals:

- Discover subdomains "without missing any" (aspirational, given limitations like private DNS)
- Verify live hosts and extract fingerprints
- Emphasize ethical use: Always get permission!

Inspiration: Built to be resilient, stealthy, and comprehensive—hardened against failures, sharpened for precision.

Key Features

1

Passive Recon Sources

crt.sh, HackerTarget, VirusTotal (API optional), Wayback Machine, BufferOver.run, AlienVault OTX, DNSdumpster.

2

Active Discovery

Wordlist brute-force, permutation generation (typos, prefixes/suffixes, numbers), DNS zone transfers.

3

Wildcard Filtering

Smart IP and content-based filtering to reduce noise.

4

Live Verification

HTTP/S probing for status codes, titles, headers, content hashes, and TLS cert SANs/CN (uncovers hidden subs).

1

Port Scanning

Optional check on common ports (e.g., 80, 443, 3389) for live hosts.

2

Recursion

Depth-limited exploration of sub-subdomains via CNAME/NS or multi-level finds.

3

Stealth & Resilience

Proxy/UA rotation, DNS resolver rotation, backoff retries, rate limiting.

4

Outputs & Thread Safety

TXT (live subs), JSONL (detailed), CSV (structured). Efficient multi-threading with locks.

How It Works: Core Phases

The tool orchestrates enumeration in structured phases:

- **Passive Reconnaissance**

  Gather subdomains from OSINT sources without alerting the target.

- **Active DNS Discovery**

  Brute-force, permutations, and zone transfers for deeper finds.

- **Live Host Verification**

  Probe HTTP/S to confirm liveliness and extract metadata/certs.

- **Port Scanning**

  Quick checks on common ports for additional intel (optional).

- **Recursive Enumeration**

  Dive into sub-subdomains (depth-limited to avoid overload).

- **Final Reporting**

  Summarize results, export data, and log details.

Installation and Dependencies

Clone the Repo:

```
git clone https://github.com/exfil0/IronVeil.git
cd IronVeil
```

Install Dependencies:

```
pip install -r requirements.txt
```

Key libs: requests, dnspython, backoff, cryptography.

Package as CLI (Recommended):

```
pip install -e .
```

Now run with ironveil [options].

Project Structure Highlights:

- src/ironveil/: Core code (enumerator.py, phases/, utils/)
- tests/: Unit tests
- docs/: Advanced usage
- examples/: Sample wordlists/proxies

Usage Guide with Examples

Basic Command:

```
ironveil -d example.com
```

CLI Options:

Option	Description	Default
-d DOMAIN	Target domain (required)	N/A
-w WORDLIST	Wordlist path	Creates mini default
-o OUTPUT	Output base file	Domain-based default
-t THREADS	Threads	20
--timeout TIMEOUT	Timeout (s)	10
-v	Verbose	False
-r RECURSION	Recursion depth	0
-p PROXIES	Proxies file	None
--no-probe	Disable probing	False
--port-scan	Enable port scan	False
--rate-limit RATE	Delay per thread (s)	0.0

Examples:

- **Basic:** ironveil -d example.com -w subdomains.txt -v
- **Full:** ironveil -d example.com -w subdomains-top1million.txt -o results.txt -r 1 --port-scan --rate-limit 0.2 -p proxies.txt
- **Passive Only:** ironveil -d example.com --no-probe
- **With API:** export VEIL_VT_API_KEY=your_key; ironveil -d example.com

Comparison to Other Tools

Vs. Amass (OWASP Tool):

- **Amass:** More mature with 87+ sources, faster (Go-based), graph DB for mapping. Great for high-volume discovery.
- **Iron Veil:** Integrated verification (probing, certs, ports) in one tool; better wildcard filtering; Python for easy customization.
- **Verdict:** Amass for breadth; Iron Veil for depth + verification. Combine them!

Vs. Subfinder or Sublist3r:

Faster for quick scans, but Iron Veil adds recursion, probing, and outputs.

Strengths of Iron Veil:

- Stealth (proxies/rate limits)
- Modularity for extensions

Ethical Considerations and Limitations

Ethical Use

- Legal reconnaissance only—get permission for third-party scans
- Rate limiting to respect services; proxies ethically sourced
- No guarantees: Misses private subs; aspirational completeness

Limitations

- IPv6 support limited (focuses on IPv4)
- No vuln scanning or screenshots
- Python slower for massive scales vs. Go tools

Risks

- Probing/scanning may alert targets or violate ToS

⊗ Always obtain proper authorization before conducting security testing on any systems you don't own.

Future Improvements

Add more passive sources

e.g., Shodan API

Enhance IPv6 probing/port scanning

Integrate vuln checks

e.g., CVE lookup via APIs

Docker support

For easy deployment

Tech fingerprinting

Wappalyzer-like

Community contributions

Extend phases/utils!

Demo Highlights

Live Demo (Imagined):

Run on example.com:

- Passive yields basics
- Active adds more
- Probing verifies live sites
- Recursion finds sub-subs

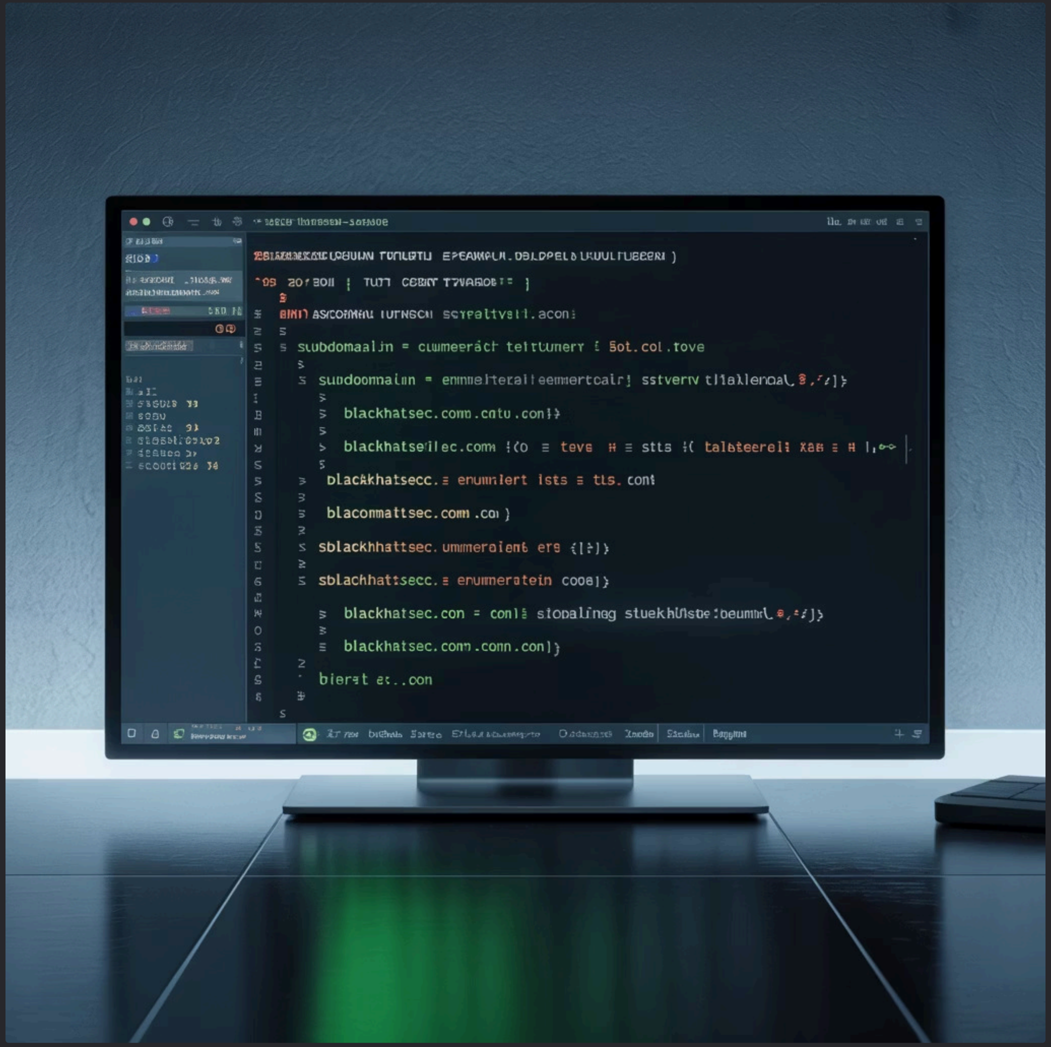
Outputs:

- JSONL with details
- CSV for analysis

Key Output Snippet:

Example JSONL entry:

```
{"domain": "www.example.com",
"is_resolved": true,
"primary_ip": "93.184.216.34",
"live_status_code": 200,
...}
```



Conclusion & Q&A

Summary:

Operation Iron Veil is a powerful, ethical tool for subdomain recon—hardened for reliability, sharpened for accuracy.

Get Started:

- Clone from GitHub
- Contribute via PRs!

Thank You!

Questions?