TheHarvester — Professional Cheat Sheet

> **Compact professional reference** for **TheHarvester** — OSINT email, subdomain and host discovery tool. Quick commands, data sources, output formats, tuning, integration tips, and OPSEC notes for red-team and reconnaissance workflows.

---

1) At-a-glance

- **Tool:** `theHarvester` — gathers emails, hostnames, subdomains, virtual hosts, open ports and people names from public sources. Useful for external reconnaissance and attack surface mapping.
- **Primary uses:** discovery of email addresses, subdomains and hosts, harvesting targets for social engineering, initial recon for penetration tests.
- **Note:** Results depend on source coverage and API keys for some providers (e.g., Shodan, Google Custom Search, Bing API).

---

2) Installation / update

```
# Kali (preinstalled usually)
sudo apt update && sudo apt install theharvester

# From source (latest)
git clone https://github.com/laramies/theHarvester.git
cd theHarvester
pip3 install -r requirements.txt
# run: python3 theHarvester.py -h
```

---

3) Basic usage pattern

```
python3 theHarvester.py -d <domain> -b <source> [options]
# or if installed system-wide
theHarvester -d example.com -b google
```

- `-d` target domain (or company name for some sources). - `-b` data source (see list below). Use `-b all` to run many sources.

---

4) Supported sources (common)

- `bing`, `google`, `yahoo`, `baidu` — search engines (API keys sometimes required).
- `shodan` — internet device data (API key recommended).
- `virustotal` — file/URL reputation and passive DNS (API key).
- `crtsh` — certificate transparency for subdomains.
- `certspotter`, `censys` — certificate & host discovery (API keys).
- `linkedin`, `twitter`, `facebook` — people/social (limited by API/changes).
- `exalead`, `bingapi`, `googleapi` — API-backed search (require keys).
- `dnsdumpster`, `threatcrowd`, `otx` (AlienVault), `fullcontact` — additional OSINT sources.

  Use `-b all` to attempt multiple sources; supplying API keys improves coverage and reduces rate-limit issues.

---

5) Important flags & options

- `-d <domain>` : domain to search.
- `-b <source>` : data source or `all`.
- `-l <limit>` : limit number of results per source (avoid huge runs).
- `-S` : active search (less common; uses additional checks).
- `-f <file>` : save results to an HTML file (report).
- `-o <file>` : save raw output to a file (JSON if `-f` requested or use `-o` and `-f`).
- `-v` : verbose.
- `-n` : only DNS check (passive DNS).
- `-p` : enable passive DNS lookups (when supported).
- `--source <file>` : use a list of sources from file.
- `--csv` : output as CSV (if supported by build).
- `--issue` : show issues or warnings found during run.
- `-h` / `--help` : show help and available modules.

---

6) Practical examples

6.1 Basic domain harvest using Bing

```
theHarvester -d example.com -b bing -l 200 -f example_bing.html
```

6.2 Use multiple sources (all) and save as HTML

```
theHarvester -d example.com -b all -l 500 -f all_sources.html
```

6.3 Use Shodan + VirusTotal (requires API keys set in config)

```
theHarvester -d example.com -b shodan -l 100 -f shodan.html
theHarvester -d example.com -b virustotal -l 200 -f vt.html
```

6.4 Enumerate subdomains only and output CSV

```
theHarvester -d example.com -b crtsh -l 500 --csv subdomains.csv
```

6.5 Run headless and output raw JSON to parse later

```
python3 theHarvester.py -d example.com -b all -l 300 -o raw_output.json
```

---

7) API keys & config

- **API keys** for services like Shodan, VirusTotal, Censys, Google/Bing APIs greatly improve results and reliability.
- Keys are usually configured in `~/.theHarvester.cfg` or `config.py` depending on version — check the repo docs.
- Respect API rate limits and quotas; cache results where possible.

---

8) Integration & workflows

- **Post-processing:** import results into SpiderFoot, Maltego, or your asset inventory.
- **Chain tools:** feed discovered subdomains to `gobuster`, `nmap`, `ffuf` and `nikto` for deeper probing.
- **Automate:** wrap theHarvester in scripts to run per domain and store outputs in a central index (Elasticsearch or SQLite).

---

9) Tuning & performance tips

- **Limit results** (`-l`) for large targets to avoid API exhaustion.
- **Source selection:** prefer `crtsh`, `shodan`, `virustotal`, `bing` for subdomain coverage.
- **Run incrementally:** start with passive sources (`crtsh`, `certspotter`) before doing heavier search engine/API calls.
- **Update theHarvester:** OSINT source endpoints change — keep your copy updated to maintain compatibility.

---

10) Troubleshooting & common pitfalls

- **API changes:** search engine and OSINT API layouts change often — check project issues if a source stops returning data.
- **Missing results:** verify API keys, check rate limits, and run `crtsh`/`certspotter` manually to compare.
- **False positives:** validate email addresses and subdomains before using them in engagement (some results are historic or typo variants).

---

11) Reporting & OPSEC

- **Sensitive data:** harvested email lists and exposed endpoints are sensitive; protect outputs.
- **Legal:** OSINT collection is generally legal, but target-focused automated scraping can violate terms of service; ensure authorized scope.
- **Responsible disclosure:** if you find exposed credentials or sensitive data, coordinate with the target organization for disclosure.

---

12) One-liners (copy-paste)

```
# Basic harvest
theHarvester -d example.com -b google -l 200 -f example_google.html

# All sources, limited results
theHarvester -d example.com -b all -l 300 -f example_all.html

# Shodan (API key required)
theHarvester -d example.com -b shodan -l 200 -f shodan.html

# Output JSON for automation
theHarvester -d example.com -b all -l 500 -o example.json
```

Further reading

- Official repo & docs: https://github.com/laramies/theHarvester
- Combine with SpiderFoot, Maltego, and passive DNS services for comprehensive reconnaissance.

This cheat sheet is for professional OSINT and reconnaissance in authorized engagements only.