# Automating Recon

Cheat Sheet

While manual reconnaissance can be effective, it can also be time-consuming and prone to human error. Automating web reconnaissance tasks can significantly enhance efficiency and accuracy, allowing you to gather information at scale and identify potential vulnerabilities more rapidly.

## Why Automate Reconnaissance?

Automation offers several key advantages for web reconnaissance:

- **Efficiency**: Automated tools can perform repetitive tasks much faster than humans, freeing up valuable time for analysis and decision-making.
- **Scalability**: Automation allows you to scale your reconnaissance efforts across a large number of targets or domains, uncovering a broader scope of information.
- **Consistency**: Automated tools follow predefined rules and procedures, ensuring consistent and reproducible results and minimising the risk of human error.
- **Comprehensive Coverage**: Automation can be programmed to perform a wide range of reconnaissance tasks, including DNS enumeration, subdomain discovery, web crawling, port scanning, and more, ensuring thorough coverage of potential attack vectors.
- **Integration**: Many automation frameworks allow for easy integration with other tools and platforms, creating a seamless workflow from reconnaissance to vulnerability assessment and exploitation.

## Reconnaissance Frameworks

These frameworks aim to provide a complete suite of tools for web reconnaissance:

- FinalRecon: A Python-based reconnaissance tool offering a range of modules for different tasks like SSL certificate checking, Whois information gathering, header analysis, and crawling. Its modular structure enables easy customisation for specific needs.
- Recon-ng: A powerful framework written in Python that offers a modular structure with various modules for different reconnaissance tasks. It can perform DNS enumeration, subdomain discovery, port scanning, web crawling, and even exploit known vulnerabilities.
- theHarvester: Specifically designed for gathering email addresses, subdomains, hosts, employee names, open ports, and banners from different public sources like search engines, PGP key servers, and the SHODAN database. It is a command-line tool written in Python.
- SpiderFoot: An open-source intelligence automation tool that integrates with various data sources to collect information about a target, including IP addresses, domain names, email addresses, and social media profiles. It can perform DNS lookups, web crawling, port scanning, and more.
- OSINT Framework: A collection of various tools and resources for open-source intelligence gathering. It covers a wide range of information sources, including social media, search engines, public records, and more.

### FinalRecon

FinalRecon offers a wealth of recon information:

- **Header Information**: Reveals server details, technologies used, and potential security misconfigurations.
- **Whois Lookup**: Uncovers domain registration details, including registrant information and contact details.
- **SSL Certificate Information**: Examines the SSL/TLS certificate for validity, issuer, and other relevant details.
- **Crawler**:
  - HTML, CSS, JavaScript: Extracts links, resources, and potential vulnerabilities from these files.
  - Internal/External Links: Maps out the website's structure and identifies connections to other domains.
  - Images, robots.txt, sitemap.xml: Gathers information about allowed/disallowed crawling paths and website structure.
  - Links in JavaScript, Wayback Machine: Uncovers hidden links and historical website data.
- **DNS Enumeration**: Queries over 40 DNS record types, including DMARC records for email security assessment.
- **Subdomain Enumeration**: Leverages multiple data sources (crt.sh, AnubisDB, ThreatMiner, CertSpotter, Facebook API, VirusTotal API, Shodan API, BeVigil API) to discover subdomains.
- **Directory Enumeration**: Supports custom wordlists and file extensions to uncover hidden directories and files.
- **Wayback Machine**: Retrieves URLs from the last five years to analyse website changes and potential vulnerabilities.

Installation is quick and easy:

Automating Recon

```
MisaelMacias@htb[/htb]$ git clone https://github.com/thewhiteh4t/FinalRecon.git
MisaelMacias@htb[/htb]$ cd FinalRecon
MisaelMacias@htb[/htb]$ pip3 install -r requirements.txt
MisaelMacias@htb[/htb]$ chmod +x ./finalrecon.py
MisaelMacias@htb[/htb]$ ./finalrecon.py --help

usage: finalrecon.py [-h] [--url URL] [--headers] [--sslinfo] [--whois]
                     [--crawl] [--dns] [--sub] [--dir] [--wayback] [--ps]
                     [--full] [-nb] [-dt DT] [-pt PT] [-T T] [-w W] [-r] [-s]
                     [-sp SP] [-d D] [-e E] [-o O] [-cd CD] [-k K]

FinalRecon - All in One Web Recon | v1.1.6

optional arguments:
  -h, --help  show this help message and exit
  --url URL   Target URL
```

Table of Contents



My Workstation

OFFLINE

Start Instance

∞ / 1 spawns left

```
  --headers   Header Information
  --sslinfo   SSL Certificate Information
  --whois     Whois Lookup
  --crawl     Crawl Target
  --dns       DNS Enumeration
  --sub       Sub-Domain Enumeration
  --dir       Directory Search
  --wayback   Wayback URLs
  --ps        Fast Port Scan
  --full      Full Recon

Extra Options:
  -nb         Hide Banner
  -dt DT      Number of threads for directory enum [ Default : 30 ]
  -pt PT      Number of threads for port scan [ Default : 50 ]
  -T T        Request Timeout [ Default : 30.0 ]
  -w W        Path to Wordlist [ Default : wordlists/dirb_common.txt ]
  -r          Allow Redirect [ Default : False ]
  -s          Toggle SSL Verification [ Default : True ]
  -sp SP      Specify SSL Port [ Default : 443 ]
  -d D        Custom DNS Servers [ Default : 1.1.1.1 ]
  -e E        File Extensions [ Example : txt, xml, php ]
  -o O        Export Format [ Default : txt ]
  -cd CD      Change export directory [ Default : ~/.local/share/finalrecon ]
  -k K        Add API key [ Example : shodan@key ]
```

To get started, you will first clone the `FinalRecon` repository from GitHub using `git clone https://github.com/thewhiteh4t/FinalRecon.git`. This will create a new directory named "FinalRecon" containing all the necessary files.

Next, navigate into the newly created directory with `cd FinalRecon`. Once inside, you will install the required Python dependencies using `pip3 install -r requirements.txt`. This ensures that `FinalRecon` has all the libraries and modules it needs to function correctly.

To ensure that the main script is executable, you will need to change the file permissions using `chmod +x ./finalrecon.py`. This allows you to run the script directly from your terminal.

Finally, you can verify that `FinalRecon` is installed correctly and get an overview of its available options by running `./finalrecon.py --help`. This will display a help message with details on how to use the tool, including the various modules and their respective options:

| Option | Argument | Description |
|---|---|---|
| `-h`, `--help` | | Show the help message and exit. |
| `--url` | URL | Specify the target URL. |
| `--headers` | | Retrieve header information for the target URL. |
| `--sslinfo` | | Get SSL certificate information for the target URL. |
| `--whois` | | Perform a Whois lookup for the target domain. |
| `--crawl` | | Crawl the target website. |
| `--dns` | | Perform DNS enumeration on the target domain. |
| `--sub` | | Enumerate subdomains for the target domain. |
| `--dir` | | Search for directories on the target website. |
| `--wayback` | | Retrieve Wayback URLs for the target. |
| `--ps` | | Perform a fast port scan on the target. |
| `--full` | | Perform a full reconnaissance scan on the target. |

For instance, if we want `FinalRecon` to gather header information and perform a Whois lookup for `inlanefreight.com`, we would use the corresponding flags (`--headers` and `--whois`), so the command would be:

Automating Recon

```
MisaelMacias@htb[/htb]$ ./finalrecon.py --headers --whois --url http://inlanefreight.com

 ______  __   __   __   ______   __
/\  ___\/\ \ /\ "-.\ \ /\  __ \ /\ \
\ \  __\\ \ \\ \ \-.  \\ \  __ \\ \ \____
 \ \_\   \ \_\\ \_\\"\_\\ \_\ \_\\ \_____\
  \/_/    \/_/ \/_/ \/_/ \/_/\/_/ \/_____/
 ______   ______   ______   ______   __   __
/\  == \ /\  ___\ /\  ___\ /\  __ \ /\ "-.\ \
\ \  __< \ \  __\ \ \ \____\ \ \/\ \\ \ \-.  \
 \ \_\ \_\\ \_____\\ \_____\\ \_____\\ \_\\"\_\
  \/_/ /_/ \/_____/ \/_____/ \/_____/ \/_/ \/_/

[>] Created By   : thewhiteh4t
 |---> Twitter   : https://twitter.com/thewhiteh4t
 |---> Community : https://twc1rcle.com/
[>] Version      : 1.1.6

[+] Target : http://inlanefreight.com

[+] IP Address : 134.209.24.248

[!] Headers :

Date : Tue, 11 Jun 2024 10:08:00 GMT
Server : Apache/2.4.41 (Ubuntu)
Link : <https://www.inlanefreight.com/index.php/wp-json/>; rel="https://api.w.org/", <https://www.inlanefreight.com/in
Vary : Accept-Encoding
Content-Encoding : gzip
Content-Length : 5483
Keep-Alive : timeout=5, max=100
Connection : Keep-Alive
Content-Type : text/html; charset=UTF-8

[!] Whois Lookup :

   Domain Name: INLANEFREIGHT.COM
   Registry Domain ID: 2420436757_DOMAIN_COM-VRSN
   Registrar WHOIS Server: whois.registrar.amazon.com
   Registrar URL: http://registrar.amazon.com
   Updated Date: 2023-07-03T01:11:15Z
   Creation Date: 2019-08-05T22:43:09Z
   Registry Expiry Date: 2024-08-05T22:43:09Z
   Registrar: Amazon Registrar, Inc.
   Registrar IANA ID: 468
   Registrar Abuse Contact Email: abuse@amazonaws.com
```


Registrar Abuse Contact Phone: +1.2024422253
Domain Status: clientDeleteProhibited https://icann.org/epp#clientDeleteProhibited
Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited
Domain Status: clientUpdateProhibited https://icann.org/epp#clientUpdateProhibited
Name Server: NS-1303.AWSDNS-34.ORG
Name Server: NS-1580.AWSDNS-05.CO.UK
Name Server: NS-161.AWSDNS-20.COM
Name Server: NS-671.AWSDNS-19.NET
DNSSEC: unsigned
URL of the ICANN Whois Inaccuracy Complaint Form: https://www.icann.org/wicf/
[+] Completed in 0:00:00.257780
[+] Exported : /home/htb-ac-643601/.local/share/finalrecon/dumps/fr_inlanefreight.com_11-06-2024_11:07:59
Previous
Next
Mark Complete & Next
Powered by HACKTHEBOX