ChatGPT

Deep Search of Personal Social Media Information: Tools, Libraries, and Techniques

Understanding Your Digital Footprint

Your **digital footprint** is the collection of all information about you available online – including social media profiles, posts, and personal data on various websites. To discover *"how much of myself is out there,"* we can employ Open-Source Intelligence (OSINT) methods. This involves using advanced search queries, people-search websites, and specialized tools to gather data from public sources. The goal is to perform an **extensive, exhaustive search** based on your first name, last name, and location, uncovering every possible mention or profile that might be associated with you.

Search Engine Techniques (Name & Location Queries)

Start with search engines like Google or Bing, using smart queries to narrow results to your name and city: - **Quoted Name + City** – For example, searching `"John Smith" Nashville` will return more targeted results for a John Smith in Nashville [1]. Quoting your full name ensures an exact match, and adding your city filters results to that location. This helps separate your information from results about other people with the same name. - **Google Dorks** – These are advanced search parameters that can dig up hidden or specific data [2]. For instance, you can search within specific social sites or exclude certain content: - **Site-specific search:**
`"<Your Name>" site:instagram.com` – finds Instagram pages where your name appears [2].
`"<Your Name>" site:twitter.com` – finds Twitter profiles or tweets mentioning your name.
- **Exclude known profiles:** If you have an account (e.g., *instagram.com/YourName*), you can exclude it to find other references like comments:
`"<Your Name>" -site:instagram.com/YourUsername site:instagram.com` [2]. This shows where your name appears on Instagram *other than* your own profile (such as comments or tags).
- **Include location keywords:** You can also add your city or state as a keyword in such queries (e.g., `"<Your Name>" "<Your City>" site:facebook.com`). This seeks pages on Facebook mentioning both your name and city. - **Multiple Search Engines** – Don't rely on Google alone. Other search engines (Bing, DuckDuckGo, Yahoo, Yandex) may surface different results [3]. Each index has unique coverage, so a thorough search means querying several engines with similar strategies.

Using these techniques will uncover news articles, public social media posts, or directory pages containing your name. Be sure to try variations of your name (middle name or nickname) and different spellings, as well as including and excluding your location to cast a wide net.

People Search Websites and Databases

There are many specialized people-search websites that aggregate public records and social media information. By inputting your name (and sometimes city/state) into these sites, you might discover phone numbers, addresses, relatives, and social profiles linked to your identity [4]. Examples include:

- **Spokeo** – Searches by name, phone, email, or username and compiles social profiles and public records [5].
- **TruthFinder / BeenVerified** – Paid services that provide comprehensive background reports, including social network accounts, if available.
- **FastPeopleSearch / TruePeopleSearch** – Free lookup tools for names, often showing addresses and associated social media.
- **X-Ray** – An OSINT tool (online) that can query multiple sources at once [5].

Keep in mind that these services often have opt-out features. While they can reveal where your personal data appears, the information may not always be up-to-date or might include other people with similar names. Use them to supplement your search, but verify any profile actually belongs to you. (For example, check if the age, location, or photos match your details.)

OSINT Frameworks and Tools for Social Media

Several open-source tools and frameworks can automate the process of finding social media accounts and personal data. These tools can save time by querying dozens of sites and databases in one go:

- **Sherlock** – A popular OSINT tool that **checks username availability on 300+ social media platforms** [6]. If you commonly use a particular handle/username, Sherlock will hunt for accounts on forums, websites, and social networks using that username. *Usage:* You provide a username, and Sherlock returns links to any found profiles [7]. This is useful if your social handles are consistent (though it works by username, not real name).
- **Maigret / Namechk / Instant Username** – Similar to Sherlock, these tools and websites search across hundreds of sites for a given username [8]. They help ensure you haven't missed an obscure platform where you might have an account. *(Namechk and Instant Username are web services, whereas Maigret is a command-line tool.)*
- **Tookie-OSINT** – An advanced OSINT tool that finds social media accounts based on various inputs (primarily usernames) [9]. Tookie is akin to Sherlock and can uncover profiles by scanning many platforms. It's known for a user-friendly interface and additional settings to broaden the search [10].
- **SpiderFoot** – A comprehensive OSINT automation tool that integrates dozens of data sources (DNS, breaches, social media, etc.). However, SpiderFoot's **strength is in domains, IPs, emails, etc., and it has limited social profile coverage by default** [11]. It can still be configured with APIs for sites like Twitter or have modules added, but out-of-the-box it may miss social accounts [12]. SpiderFoot is useful if you want to include technical footprint (e.g., your personal website, past emails, data breaches involving your info) in addition to social media.
- **Recon-ng** – A modular recon framework (similar to Metasploit but for OSINT). It has modules to search for profiles using the **WhatsMyName** database, which is a collection of websites and their username formats [13]. Recon-ng can also tap into APIs for Facebook, LinkedIn, etc., if you supply API keys. This tool requires comfort with command-line interfaces but can automate searches across many sites.

- **SpiderFoot + Recon-ng with *WhatsMyName*** – The *WhatsMyName* project is essentially a list of websites and how to query them for a username. Both SpiderFoot and Recon-ng have modules to utilize this list [13]. This means they can automate the same kind of username search that Sherlock/ Namechk do. If you provide a list of usernames or even parts of your name as potential usernames, these tools can check those across many platforms.
- **Maltego** – A visual link-analysis tool. With the Community Edition (free), you can use **transforms** (integrations) to search for people by name, email, or phone. Maltego can, for example, find social media links given an email or do reverse image search on profile photos. It's more interactive than script-based; you drag nodes on a graph. If you prefer a GUI and seeing relationships (like which accounts share the same username or email), Maltego is powerful. However, full use may require API keys or the paid version for certain transforms.
- **DaProfiler** – An OSINT tool *specifically designed to collect information about yourself* (the name hints at "Data Profiler"). **DaProfiler can gather addresses, social media accounts, email addresses, phone numbers, and jobs associated with your identity** [14]. It was created to help individuals find traces of their personal data online in order to exercise data privacy rights (like GDPR removal requests). This tool might use several of the above techniques under the hood: searching social networks, querying people-search databases, etc., all centered around your name. *(Note: DaProfiler's documentation indicates it's geared towards persons in certain regions – one fork mentions France – but the concept is applicable generally [15].)*
- **Other Notable Tools**:
- **theHarvester** – Originally for finding emails and subdomains, it also searches search engines and key sites for a given name or email [16]. It's less focused on social media profiles, but if you use it with your full name, it might pick up public mentions or related domains.
- **Holehe** – Checks if a given **email address** is associated with accounts on ~120 websites by testing logins or password reset forms [17]. If you use the same email for all your social accounts, running Holehe on your email can quickly list where that email is registered (Twitter, Instagram, etc.) without logging in to each. (This can reveal accounts you forgot about.)
- **HaveIBeenPwned** – Not exactly a social media finder, but it tells you if your email (or phone) appeared in any data breaches [18]. This can indirectly highlight old accounts tied to that email. For example, a breach might show you had an account on some forum or site years ago.

Each of these tools requires varying levels of technical skill. Many are on GitHub with installation instructions. For instance, Sherlock is a simple Python script you can run with a username as argument [7], whereas SpiderFoot has a web interface you can run locally, and Recon-ng involves an interactive console. It's wise to start with one or two tools (like Sherlock for usernames, and DaProfiler for a broad sweep) before moving to more complex frameworks.

Beware of false positives: if your name or username is common, these tools will find accounts that belong to other people with the same name/usernames. You will need to manually verify which discovered profiles are actually yours. As OSINT experts note, *you should expect some irrelevant results when someone else uses the same handle or name* [19]. Always cross-check details (photos, bios, etc.) to confirm identity.

Python Libraries for Web Scraping and Crawling

To go truly deep, you might need to custom-build a web crawler or scraper that scours the web for your personal info. Python has an excellent ecosystem for web scraping and OSINT automation:

- **Requests** – A simple HTTP library to fetch web pages. Use it to download HTML from search results or profile pages as needed.
- **Beautiful Soup** – An HTML parsing library that makes it easy to extract information from fetched pages. For example, you could fetch a Google search results page and use BeautifulSoup to pull out all the result links, or scrape specific social media profile fields from a page's HTML.
- **Selenium** – A browser automation tool. Some social sites (like LinkedIn or Facebook) heavily rely on dynamic content or require login to view detailed info. Selenium can simulate a real browser, navigate to a page, and let you scrape content after the page fully loads (or after logging in with your credentials, if you choose). This is heavier and slower than Requests/BS4, but sometimes necessary for pages that don't show content to non-browsers.
- **Scrapy** – A powerful web crawling framework. If you plan to crawl through many links (for example, systematically follow all Google result pages for your name, or traverse all posts on a forum where you suspect your name appears), Scrapy provides a robust structure for large-scale crawls. It handles queuing URLs, parsing data, and can be extended with pipelines to save results.
- **Python OSINT libraries** – There are libraries specifically made for scraping certain platforms, which can save you time versus writing your own scrapers:
- **snscrape**: A versatile scraper for social networks. It can gather **profiles, posts, or search results from sites like Twitter (X), Facebook, Instagram, Reddit, etc.** via a Python API [20] [21]. For instance, you can use snscrape to search Twitter for your name (even without an API key) and collect recent tweets mentioning you. Or use it to get all posts from a Reddit user if you find an account that might be yours.
- **Twint**: An advanced Twitter scraping tool (Python-based) that doesn't require the official API [22]. With Twint, you can scrape a user's tweets, followers, or search Twitter by keywords. *Note:* Twitter's site changes have sometimes broken Twint, so ensure you get the latest updates or consider snscrape as an alternative for Twitter.
- **Instaloader**: A tool to scrape Instagram **profiles, posts, and metadata** (including public and private profiles you have access to) [23]. In a script, you could use Instaloader to download all pictures and captions from an Instagram profile or to get the list of followers/following. If you have an Instagram account, it can also scrape private profiles you follow (with your login). This helps gather what information *your own* profile is exposing, or to confirm the content of any impostor accounts that might be pretending to be you.
- **facebook-scraper**: A Python library (available on PyPI) that can scrape public Facebook content – posts from public pages or profiles, group posts, etc. – **without needing the official API or login** [24]. It's limited to publicly visible data, but if your Facebook profile is public or you have public posts/comments, this could retrieve them. It's also useful to monitor public groups or pages in your town for mentions of your name. *(Note: Much of Facebook is private, so this has limits. For more in-depth Facebook data, using their Graph API with a user token or Selenium automation might be necessary.)*
- **PRAW (Reddit API)**: If you suspect there's content on Reddit about you (maybe your name was mentioned in a thread), you can use Reddit's API via PRAW to search. Reddit's search can be accessed via their API to find submissions or comments containing a keyword (your name). This requires API credentials (free to obtain) and respects rate limits but is straightforward to use.

- **Other APIs**: Many platforms provide some official API or at least a search endpoint. For example, you can use the **GitHub API** to search for your name (perhaps you want to find if your name or email is exposed in any public code or in commit logs). Or use the **YouTube Data API** to find videos with your name in the title/description. If you have distinct personal interests, searching those in combination with your name might reveal forum accounts or old blog posts.

Using these libraries, you can build a comprehensive script. For example, you could write a Python script that: (1) uses Google's Custom Search API or **Bing Web Search API** to get web results for your name, (2) scrapes each result page for any contact info or new leads (like usernames or emails tied to you), (3) queries social media scrapers (snscrape/Twint/Instaloader) for each platform to fetch details of any discovered accounts, and then (4) compiles all this information into a report.

Python is extremely flexible for OSINT. As one guide notes, you can combine libraries like **BeautifulSoup**, **requests**, and various APIs to automate OSINT tasks, and even schedule these scripts to run regularly [25]. This means you can systematically collect data without manual clicking.

Ethical tip: Ensure you respect each site's Terms of Service while scraping. Too-aggressive scraping can trigger anti-bot measures. In fact, Google may temporarily block you with CAPTCHAs if you automate too many search queries quickly [26]. To mitigate this, implement **rate limiting** (pausing between requests), use API endpoints when possible (like Google Custom Search API which allows a certain number of queries per day), or integrate proxy servers to distribute requests. Always comply with legal guidelines – scraping public data for personal use is generally fine, but avoid accessing any account data you're not authorized to see.

Continuous Monitoring and Alerts

Finding your current online information is one step – but you also asked about a script that **continuously does this deep search**. This is like setting up your personal digital footprint watchdog. Here's how to implement it:

1. **Automate the Search Workflow:** Once you have a script or set of tools that collects the information (as described above), wrap it into a repeatable function. For example, create a Python script that performs all the searches (Google dorks, OSINT tools, scraping various sites) and outputs the findings (perhaps into a file or database). Ensure this script is idempotent – running it again should yield the same results if nothing new is found, so you can compare changes over time.
2. **Schedule Regular Scans:** Use a scheduler like cron (on Linux/Mac) or Task Scheduler (on Windows) to run your script at desired intervals (daily, weekly, etc.) [27]. For instance, a weekly scan could check for any new web pages or social media posts that mention your name. Scheduling ensures **24/7 monitoring**, so you don't have to remember to manually rerun the search [28] [29].
3. **Track and Store Results:** Each run, save the results (e.g., as a dated report or in a database). Over time, you will accumulate records of your online mentions. By storing past results, your script can compare the current findings with the previous run. If something is newly found (a new Twitter mention or a new public profile appearing), you can flag that.
4. **Alerts for New Information:** Integrate an alerting mechanism. This could be as simple as the script emailing you if it finds a *new* mention of your name, or a new account that wasn't there before. You could use Python libraries like `smtplib` to send an email, or integrate with messaging APIs (e.g., send a Slack message or SMS via Twilio). The key is to notify you when there is a change so you can promptly review it.

5. **Refine and Update:** The internet is dynamic. New social media platforms emerge, old ones change their search features. Keep your toolset updated – for example, if you find a new OSINT tool on GitHub that covers a platform your current script misses, consider adding it. Also, adjust for false positives: you might initially get irrelevant data (for instance, someone else with your same name in a different country). Over time, you can refine your queries or filtering to focus on the data truly about you.

By following these steps, you essentially build your own **personal OSINT alert system**. This is similar in concept to Google Alerts (which emails you when new pages with certain keywords appear), but your custom approach can be far more exhaustive and tailored to multiple sources.

Example: Suppose your script runs every Sunday. This week, it searches and finds nothing new except a Tumblr blog post that mentions your name in an unrelated context. The script flags this as new, sends you an alert, and you can then check if it's actually about you or another person with your name. If it's not relevant, you might tweak your search to exclude that site in the future. If it *is* you (maybe a friend mentioned you in their blog), you've just discovered another piece of your digital footprint.

Finally, ensure you handle the data responsibly. You are essentially "doxing" yourself – which is fine since it's your data – but protect the results, especially if they include sensitive info like addresses or phone numbers. If you decide to share this data or use it for a project, remember it contains personally identifiable information. Keep your monitoring script secure as well (store API keys safely, etc.).

Conclusion

Using the combination of **advanced search techniques, OSINT tools, Python scraping libraries, and automation**, you can compile an extremely comprehensive view of your online presence. We covered how to pinpoint social media accounts (even across hundreds of sites using tools like Sherlock), how to scour the web for name and location mentions, and how to leverage Python to automate and continuously monitor these searches. By setting up a continuous crawl and alert system, you'll be immediately informed of new instances of your personal information appearing online [30] [31].

This deep search process will likely reveal old accounts you may have forgotten, posts where you were tagged, and entries in public databases. It can be eye-opening to see the *extent* of one's digital footprint. Always remember to use this information for good – for example, you might decide to tighten privacy on certain accounts or request removal of data from people-search sites once you discover it. Also stay within legal and ethical bounds: gather only publicly available data and respect any usage policies of the platforms you query [32].

In summary, by diligently using the tools and methods described, you'll achieve an **extensive and exhaustive view of your social media information online** – and with continuous monitoring in place, you'll keep that knowledge up-to-date as new information surfaces about you.

Sources: The strategies and tools discussed above are drawn from OSINT best practices and reputable sources, including the Molfar OSINT people-search guide [2] [5], community-curated OSINT tool repositories [14] [9], and expert tutorials on deep web searching and automation [1] [25]. These sources provide further details on using Google dorks, specialized search services, and Python scripts to conduct thorough personal information searches. Each tool (Sherlock [6], SpiderFoot [11], etc.) and library (snscrape

[20], Instaloader [23], etc.) mentioned is well-documented on GitHub or developer sites for implementation details. By consulting those references and the guidelines above, you can confidently build a powerful system to discover and monitor your complete social media footprint.

---

[1] Finding Key Decision Makers
https://callproof.com/find-key-decision-makers/

[2] [3] [4] [5] [8] [13] [16] [18] [19] [26] How to find information on anyone: The best OSINT tools for people search – Molfar
https://molfar.com/en/blog/how-to-find-information-on-anyone-the-best-osint-tools-for-people-search

[6] [7] Sherlock - Hunt Username on Social Media Kali Linux Tool - GeeksforGeeks
https://www.geeksforgeeks.org/linux-unix/sherlock-hunt-username-on-social-media-kali-linux-tool/

[9] [14] osint-tools · GitHub Topics · GitHub
https://github.com/topics/osint-tools

[10] Tookie, an advanced OSINT social media tool - Reddit
https://www.reddit.com/r/OSINT/comments/1c79knz/tookie_an_advanced_osint_social_media_tool/

[11] [12] Any name/address automated OSINT tools like spiderfoot? : r/OSINT
https://www.reddit.com/r/OSINT/comments/l606r1/any_nameaddress_automated_osint_tools_like/

[15] dalunacrobate/DaProfiler - GitHub
https://github.com/dalunacrobate/DaProfiler

[17] Holehe OSINT - Email to Registered Accounts - GitHub
https://github.com/megadose/holehe

[20] Snscrape - A social networking service scraper in Python - GitHub
https://github.com/JustAnotherArchivist/snscrape

[21] Snscrape: What it is, How to Use it, & More! - RapidSeedbox
https://www.rapidseedbox.com/blog/snscrape

[22] twintproject/twint: An advanced Twitter scraping & OSINT tool written ...
https://github.com/twintproject/twint

[23] Instaloader — Download Instagram Photos and Metadata
https://instaloader.github.io/

[24] facebook-scraper - PyPI
https://pypi.org/project/facebook-scraper/

[25] [27] [28] [29] [31] [32] Automating OSINT for Threat Intelligence: Scaling Your Workflow — Part 3/4 | by Adil Shafiq | Medium
https://medium.com/@gh4reeb/automating-osint-for-threat-intelligence-scaling-your-workflow-part-3-e192c2c9a5dd

[30] OSINT Techniques: Complete List for Investigators (2025)
https://shadowdragon.io/blog/osint-techniques/