Black Bird

By: Taja Hicks, Luis Morales,
Abigail Rodriguez Vazquez

Introduction to Project

BLACKBIRD IS AN OSINT TOOL THAT COMBINES USERNAME AND EMAIL SEARCHES ACROSS MORE THAN 600 PLATFORMS. THIS IS DONE USING AI TO PROFILE

BY UTILIZING COMMUNITY-DRIVEN PROJECTS (LIKE WHATSMYNAME), IT LOWERS FALSE POSITIVE RATES AND PROVIDES HIGH-QUALITY RESULTS

SOME MAIN FEATURES INCLUDE SMART FILTERS, POLISHED PDF/CSV EXPORTS, AND MORE

Black Bird Goal/Purpose

- The goal of Blackbird is to use AI to analyze the sites where a username or email is found and return a behavioral and technical profile of the user.

- Helping you understand more about someone, whether it be what language they are learning , what platforms they participate in

- As well as some of the risks they might encounter, all with less effort.

What is Blackbird good for ?

In the aspect of computer security, Blackbird is useful for

1) Data Breach:

You can use Blackbird to better locate the websites or apps you are associated with and help you see if you might have been impacted by a breach.

This can also help you ensure that you use different passwords for different accounts on different platforms

What is Blackbird good for ?

2) Awareness:

SINCE BLACKBIRD RETURNS A BEHAVIORAL AND TECHNICAL PROFILE OF THE USER, WE CAN BECOME MORE AWARE ABOUT OR ONLINE PRESENCE.

THIS AWARENESS CAN HELP YOU BETTER UNDERSTAND SOME OF THE VULNERABILITIES YOU MAY ENCOUNTER AND MITIGATE THEM.

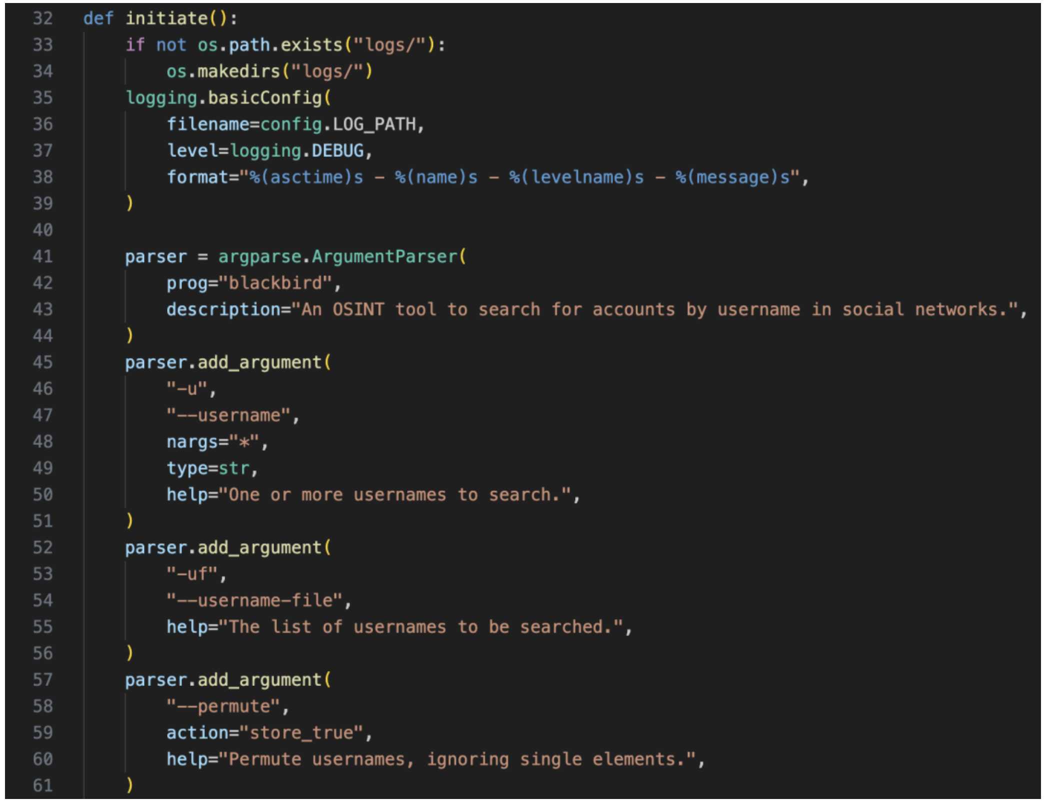
AS WELL AS BEING MORE MINDFUL OF OUR COMPUTER SECURITY AND IMPLEMENTING MORE SECURE LOGIN SYSTEMS SUCH AS TWO-FACTOR AUTHENTICATION.

Getting Started

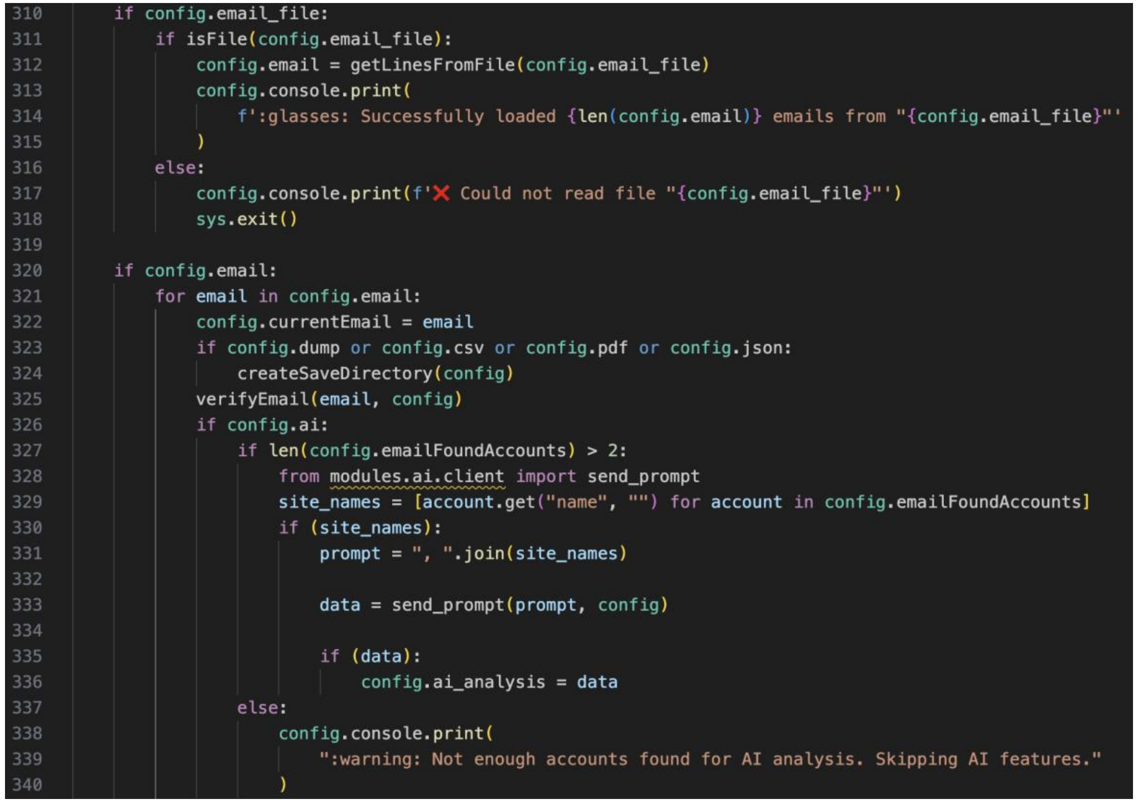
- How to Initialize/Start Blackbird Demo
    - Installation
        - First – Clone the GitHub repository into your local environment
            - !git clone https://github.com/p1ngul1n0/blackbird
        - Next - Navigate to the newly created "blackbird" directory
            - %cd blackbird
        - Then – Install all the Python dependencies listed in the requirements.txt file
            - !pip install -r requirements.txt
        - Finally – After the requirements are installed, you may need to reset your machine to update the changes
            - Once the machine is back up Blackbird is ready to begin
    - Use
        - Once Blackbird has been installed searches will be available
            - !python blackbird.py --name <samplename>

Introducing code

blackbird.py



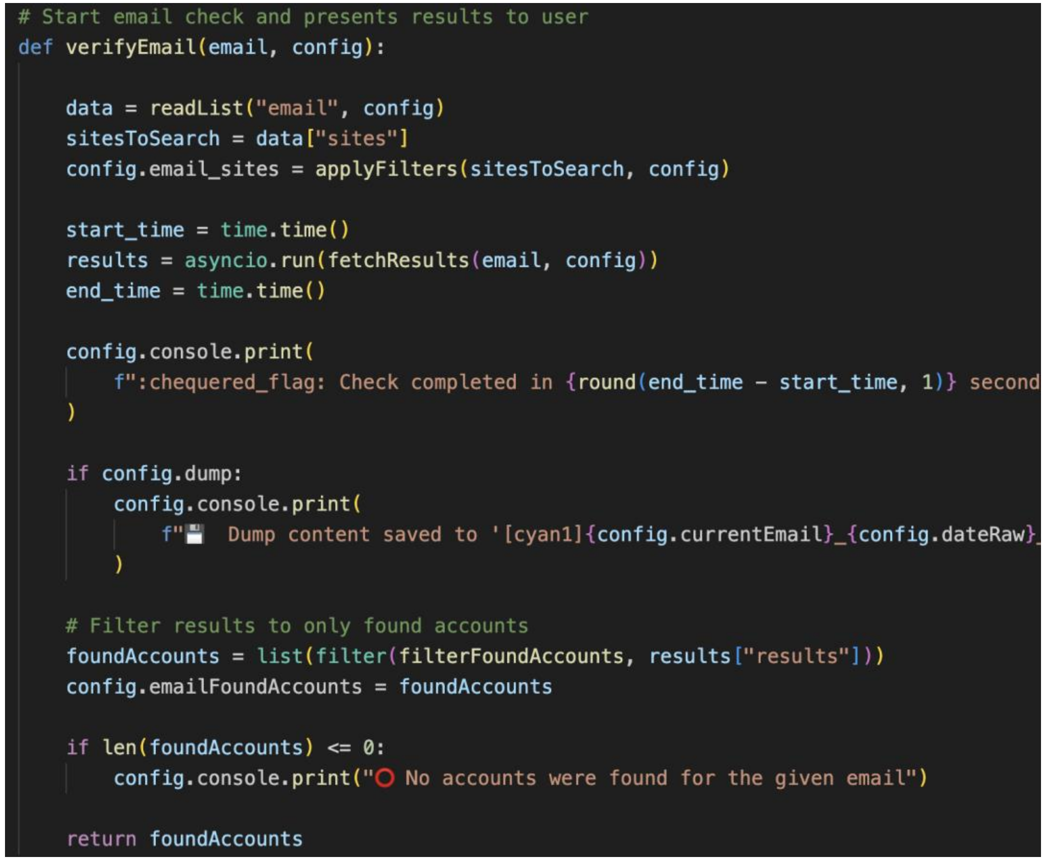
```python
def initiate():
    if not os.path.exists("logs/"):
        os.makedirs("logs/")
    logging.basicConfig(
        filename=config.LOG_PATH,
        level=logging.DEBUG,
        format="%(asctime)s - %(name)s - %(levelname)s - %(message)s",
    )

    parser = argparse.ArgumentParser(
        prog="blackbird",
        description="An OSINT tool to search for accounts by username in social networks.",
    )
    parser.add_argument(
        "-u",
        "--username",
        nargs="*",
        type=str,
        help="One or more usernames to search.",
    )
    parser.add_argument(
        "-uf",
        "--username-file",
        help="The list of usernames to be searched.",
    )
    parser.add_argument(
        "--permute",
        action="store_true",
        help="Permute usernames, ignoring single elements.",
    )
```



```python
    if config.email_file:
        if isFile(config.email_file):
            config.email = getLinesFromFile(config.email_file)
            config.console.print(
                f':glasses: Successfully loaded {len(config.email)} emails from "{config.email_file}"'
            )
        else:
            config.console.print(f'❌ Could not read file "{config.email_file}"')
            sys.exit()

    if config.email:
        for email in config.email:
            config.currentEmail = email
            if config.dump or config.csv or config.pdf or config.json:
                createSaveDirectory(config)
            verifyEmail(email, config)
            if config.ai:
                if len(config.emailFoundAccounts) > 2:
                    from modules.ai.client import send_prompt
                    site_names = [account.get("name", "") for account in config.emailFoundAccounts]
                    if (site_names):
                        prompt = ", ".join(site_names)

                        data = send_prompt(prompt, config)

                        if (data):
                            config.ai_analysis = data
                else:
                    config.console.print(
                        ":warning: Not enough accounts found for AI analysis. Skipping AI features."
                    )
```

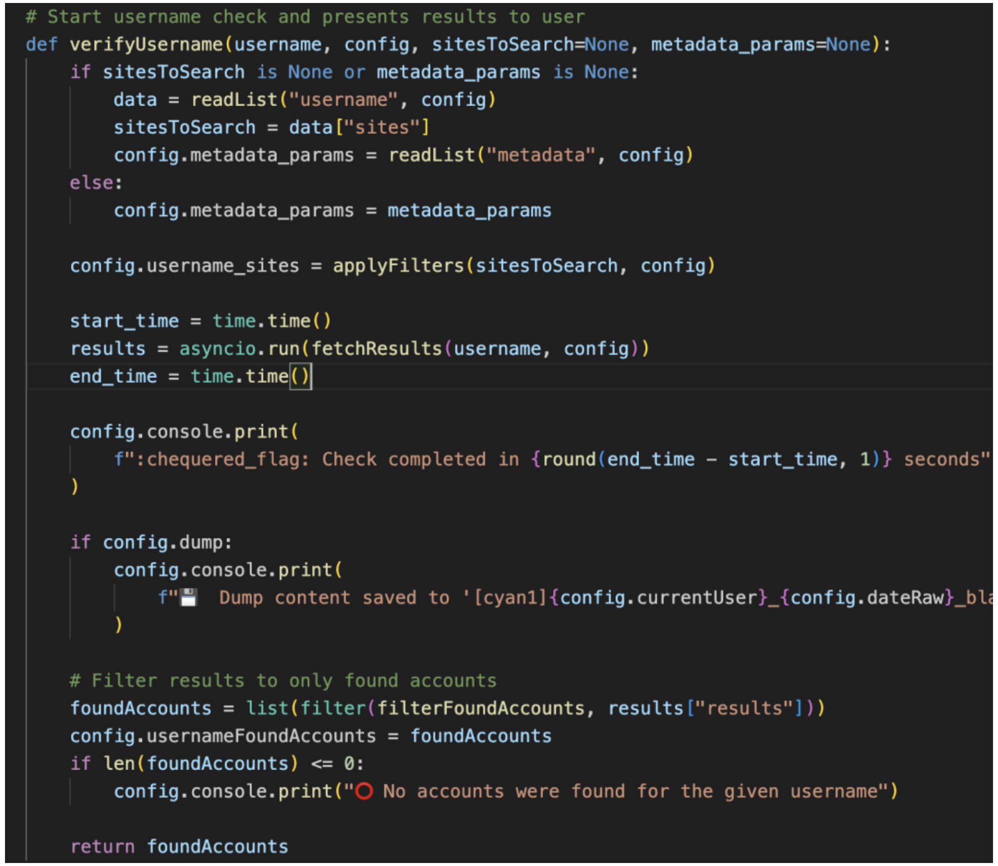
CODE

email.py



```python
# Start email check and presents results to user
def verifyEmail(email, config):

    data = readList("email", config)
    sitesToSearch = data["sites"]
    config.email_sites = applyFilters(sitesToSearch, config)

    start_time = time.time()
    results = asyncio.run(fetchResults(email, config))
    end_time = time.time()

    config.console.print(
        f":chequered_flag: Check completed in {round(end_time - start_time, 1)} second
    )

    if config.dump:
        config.console.print(
            f"💾  Dump content saved to '[cyan1]{config.currentEmail}_{config.dateRaw}_
        )

    # Filter results to only found accounts
    foundAccounts = list(filter(filterFoundAccounts, results["results"]))
    config.emailFoundAccounts = foundAccounts

    if len(foundAccounts) <= 0:
        config.console.print("⭕ No accounts were found for the given email")

    return foundAccounts
```

username.py



```python
# Start username check and presents results to user
def verifyUsername(username, config, sitesToSearch=None, metadata_params=None):
    if sitesToSearch is None or metadata_params is None:
        data = readList("username", config)
        sitesToSearch = data["sites"]
        config.metadata_params = readList("metadata", config)
    else:
        config.metadata_params = metadata_params

    config.username_sites = applyFilters(sitesToSearch, config)

    start_time = time.time()
    results = asyncio.run(fetchResults(username, config))
    end_time = time.time()

    config.console.print(
        f":chequered_flag: Check completed in {round(end_time - start_time, 1)} seconds"
    )

    if config.dump:
        config.console.print(
            f"💾  Dump content saved to '[cyan1]{config.currentUser}_{config.dateRaw}_bla
        )

    # Filter results to only found accounts
    foundAccounts = list(filter(filterFoundAccounts, results["results"]))
    config.usernameFoundAccounts = foundAccounts
    if len(foundAccounts) <= 0:
        config.console.print("⭕ No accounts were found for the given username")

    return foundAccounts
```

Examples of Commands

//Blackbird has 11 core commands that can be used with blackbird.py

- Search for accounts by username.
  - --username <name>
- Search for accounts by email.
  - --email <email>
- List all supported platforms (600+ sites via WhatsMyName integration).
  - --list-sites
- Display all results, not just found accounts.
  - --show-all
- Export results to PDF
  - --pdf
- Port results to CSV.
  - --csv
- Generate an API key for AI-powered profiling.
  - --setup-ai
- Enable AI analysis of found accounts.
  - --ai
- Use a proxy for requests.
  - –proxy <url>
- Launch a local web server for GUI access.
  - --web
- Read results from a saved JSON file
  - -f <filename>

Example Usage

- python blackbird.py --username johndoe
- python blackbird.py --email johndoe@example.com --pdf
- python blackbird.py --list-sites
- python blackbird.py --username johndoe --ai

Examples of Filters

//Blackbird has built-in filters to help find more relevant information from searches

- Filter out NSFW sites
  - --no-nsfw
- Find permutations of the search information
  - --permute
- Cast a wider net for permutations
  - --permuteall
- Search for a list of emails
  - --email-file <email-file>
- Show verbose search results
  - --verbose
- Use more customized filters.
  - --filter <filter>
- Dump HTML content for found accounts
  - --dump
- Enable AI analysis of found accounts.
  - --ai

Example Usage

- python blackbird.py --username johndoe --filter "cat=social"
- python blackbird.py --username johndoe --no-nsfw
- python blackbird.py --list-sites
- python blackbird.py --username johndoe --dump

Blackbird

Conclusion

Basic Uses

- Username Reverse Search
- Email Reverse Search
- Export as PDF, CSV, JSON, or DUMP

AI

- Generates behavioral insights
- Included in exported PDF reports
- Free but subject to a daily quota
- Personal info, such as names, not sent to AI API

Advanced Usage

- Verbose – More detailed information
- Permute – Generate variations of a given username
- Proxy – Routes all HTTP requests through a proxy

Use Cases

- Reverse search:
  - Multiple emails and usernames & export
  - An email and save HTTP response data for analysis
  - A username on Instagram to obtain email & phone number
  - List of usernames in TikTok

sources

- Antoniaci , L. (2020,). Blackbird. https://p1ngul1n0.gitbook.io/blackbird

- Images [slide: 2,3,4,5,10]from Microsoft 365 stock images

- Freepik. (n.d.). Crow Black Crow realistic illustration isolated on transparent background | premium AI-generated PSD. Freepik . https://www.freepik.com/premium-psd/crow-black-crow-realistic-illustration-isolated-transparent-background_222790846.htm

- Crows artwork, Bird Silhouette, dark ravens transparent. PNG Play. (2021, July 24). https://www.pngplay.com/image/185627

- Library, C. (n.d.). *Free blood drip transparent background, download free blood drip transparent background PNG images, free cliparts on Clipart library*. Clipart Library. https://clipart-library.com/free/blood-drip-transparent-background.html

sources

- library, C. (n.d.). *Free blood Splatter Clipart, download free blood splatter clipart PNG images, free cliparts on Clipart library*. Clipart library. https://clipart-library.com/blood-splatter-clipart.html

- FreePik. (n.d.). *Eerie swamp scene with a crow perched on a branch fog rolling over still water and silhouetted trees creating a dark mysterious atmosphere | premium AI-generated image*. https://www.freepik.com/premium-ai-image/eerie-swamp-scene-with-crow-perched-branch-fog-rolling-still-water-silhouetted-trees-creating-dark-mysterious-atmosphere_274742820.htm

- Wallpapers. (n.d.). *[100+] blood moon PNG images*. Wallpapers.com. https://wallpapers.com/blood-moon-png