

## Required APIs:

1. **Prospero API** (For fetching email addresses)
    - API URL: `https://api.prospeo.io/domain-search`
    - **Signup Instructions:**
      - Go to [Prospero](#).
      - Sign up for an account and obtain an API key.
      - Use the API key in your tools for fetching emails.
  2. **WhoisXML API** (For WHOIS, Subdomains, IP Geolocation, Netblock, SSL, and IP Lookups)
    - API URL:
      - [WhoisXML API](#): Provides WHOIS, IP geolocation, subdomain information, and netblock details.
    - **Signup Instructions:**
      - Go to [WhoisXML API](#).
      - Sign up for an account.
      - Upon registration, you'll be given an API key.
      - Use the API key in your tools for WHOIS, IP geolocation, netblock, etc.
  3. **WhatCMS API** (For Detecting CMS on websites)
    - API URL: `https://whatcms.org/API/Tech`
    - **Signup Instructions:**
      - Go to [WhatCMS API](#).
      - Sign up and obtain an API key for CMS detection.
  4. **SSL Certificates API (WhoisXML API)** (For SSL Certificate Information)
    - API URL: `https://ssl-certificates.whoisxmlapi.com/api/v1`
    - **Signup Instructions:**
      - Go to WhoisXML API SSL page.
      - Sign up and obtain an API key.
- 

## Dependencies to Install on Linux:

You can install the required dependencies using `pip` for Python. These dependencies are required for API interactions and other functionality.

1. **requests** - Used for making HTTP requests to APIs.

Install with:

```
bash
CopyEdit
pip install requests
```

2. **subprocess** - For running system commands like `nslookup`.

- This is part of Python's standard library, so no installation is required.
  - 3. **re** - For regular expressions.
    - This is part of Python's standard library, so no installation is required.
  - 4. **json** - For handling JSON data.
    - This is part of Python's standard library, so no installation is required.
- 

## Setting Up the Project (Linux Guidelines):

### 1. Clone or Create a New Directory:

Create a directory for your project and add all your Python scripts into it.

```
bash
CopyEdit
mkdir domain_tools_project
cd domain_tools_project
```

### 2. Install Python (if not installed):

Ensure you have Python installed. If not, install it using:

```
bash
CopyEdit
sudo apt update
sudo apt install python3 python3-pip
```

### 3. Install Dependencies:

Install `requests` for making HTTP requests. Other dependencies (`subprocess`, `re`, `json`) are part of Python's standard library.

```
bash
CopyEdit
pip install requests
```

### 4. API Key Configuration:

- Store your API keys securely in your code. **Never** hard-code your API keys directly in your scripts for production. Consider using environment variables or a `.env` file.

Example of setting up API keys using environment variables:

```
bash
CopyEdit
export PROSPEO_API_KEY="your_prospeo_api_key"
export WHOISXML_API_KEY="your_whoisxml_api_key"
export WHATCMS_API_KEY="your_whatcms_api_key"
export SSL_CERT_API_KEY="your_ssl_cert_api_key"
```

Then, in your Python script, access these keys like this:

```
python
CopyEdit
import os
prospero_api_key = os.getenv('PROSPEO_API_KEY')
whoisxml_api_key = os.getenv('WHOISXML_API_KEY')
```

5. **Testing:**

Run each script separately by providing appropriate inputs like domain names or IP addresses. Make sure your API keys are working, and the responses from the APIs are as expected.

6. **Error Handling:**

Consider adding robust error handling to deal with issues such as rate limits, API failures, or missing data.

7. **Security:**

Always secure API keys, especially in production. Avoid publishing or sharing code with hardcoded keys. Consider using API key management services for enhanced security.

8. **Documentation:**

Keep documentation up to date. Include instructions for others to sign up for the APIs, install dependencies, and configure the environment.

---

## Other Important Guidelines:

- **Rate Limiting:** Many of the APIs (like WhatCMS, WhoisXML, etc.) may have rate limits. Be mindful of this, especially during development. You may need to request higher limits if you're doing extensive scanning.
- **API Quotas:** Ensure you are aware of any free tier limits for each API. Some APIs have usage caps depending on your subscription.
- **Error Handling:** In production or large-scale use, implement a system to handle errors gracefully (e.g., logging, retries, or user notifications in case of failure).
- **Test thoroughly:** After setting up, run each tool with sample domains to verify everything works as expected before using it on multiple domains.

---

## Example of Using the Scripts:

1. **Run the Main File:**

After configuring the API keys and dependencies, run the main file to start the domain information retrieval process:

```
bash
CopyEdit
python main.py
```

2. **Provide a Domain:**

When prompted, enter the domain (e.g., `example.com`).

3. **See Results:**

The results from different tools (emails, CMS detection, WHOIS, SSL info, etc.) will be printed in the terminal.