Page 15 / Creepy Crawlles

Creepy Crawlies

Web crawling is vast and intricate, but you don't have to embark on this journey alone. A plethora of web crawling tools are available to assist you, each with its own strengths and specialties. These tools automate the crawling process, making it faster and more efficient, allowing you

Popular Web Crawlers

- crawler called Spider. Spider excels at mapping out web applications, identifying hidden content, and uncovering potential vulnerabilities.
- 2. OWASP ZAP (Zed Attack Proxy): ZAP is a free, open-source web application security scanner. It can be used in automated and manual modes and includes a spider component to crawl web applications and identify potential
- 3. Scrapy (Python Framework): Scrapy is a versatile and scalable Python framework for building custom web
- 4. Apache Nutch (Scalable Crawler): Nutch is a highly extensible and scalable open-source web crawler written in requires more technical expertise to set up and configure, its power and flexibility make it a valuable asset for large-scale reconnaissance projects.

Adhering to ethical and responsible crawling practices is crucial no matter which tool you choose. Always obtain permission before crawling a website, especially if you plan to perform extensive or intrusive scans. Be mindful of the website's server resources and avoid overloading

Scrapy

We will leverage Scrapy and a custom spider tailored for reconnaissance on inlanefreight.com. If you are interested in more information on crawling/spidering techniques, refer to the "Using Web Proxies" module, as it forms part of CBBH as well.

Installing Scrapy

Before we begin, ensure you have Scrapy installed on your system. If you don't, you can easily install it using pip, the Python package installer:



This command will download and install Scrapy along with its dependencies, preparing your environment for building our spider.

ReconSpider

First, run this command in your terminal to download the custom scrapy spider, ReconSpider, and extract it to the current working directory.



With the files extracted, you can run ReconSpider.py using the following command:



Replace inlanefreight.com with the domain you want to spider. The spider will crawl the target and collect valuable information.

results.ison

structure of the JSON file produced:

```
Code: json
"emails": [
    "lily.floid@inlanefreight.com",
    "cvs@inlanefreight.com"
    "https://www.themeansar.com",
    "https://www.inlanefreight.com/index.php/offices/",
"external_files": [
    "https://www.inlanefreight.com/wp-content/uploads/2020/09/goals.pdf",
     https://www.inlanefreight.com/wp-includes/js/jquery/jquery-migrate.min.js?ver=3.3.2",
```





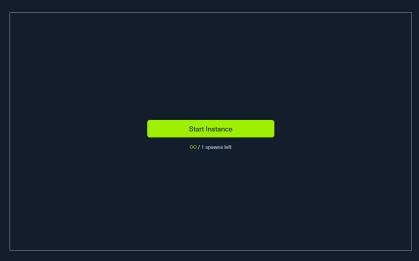


Each key in the JSON file represents a different type of data extracted from the target website:

Description
Lists email addresses found on the domain.
Lists URLs of links found within the domain.
Lists URLs of external files such as PDFs.
Lists URLs of JavaScript files used by the website.
Lists form fields found on the domain (empty in this example).
Lists URLs of images found on the domain.
Lists URLs of videos found on the domain (empty in this example).
Lists URLs of audio files found on the domain (empty in this example).
Lists HTML comments found in the source code.

By exploring this JSON structure, you can gain valuable insights into the web application's architecture, content, and potential points of interest





Waiting to start...

