EXPERMINT: 06

● **Aim:** Utilize website crawling OSINT tools to gather a comprehensive list of URLs, internal links, and structure of the website.

● **Theory:**

Website crawling, often referred to as web crawling or web scraping, is the automated process of systematically navigating and collecting data from websites. It involves accessing web pages, extracting information, and following links to other pages within the website. Website crawling is a common practice used for various purposes, including data collection, content indexing, SEO analysis, and competitive research.

Here are the key aspects of website crawling:

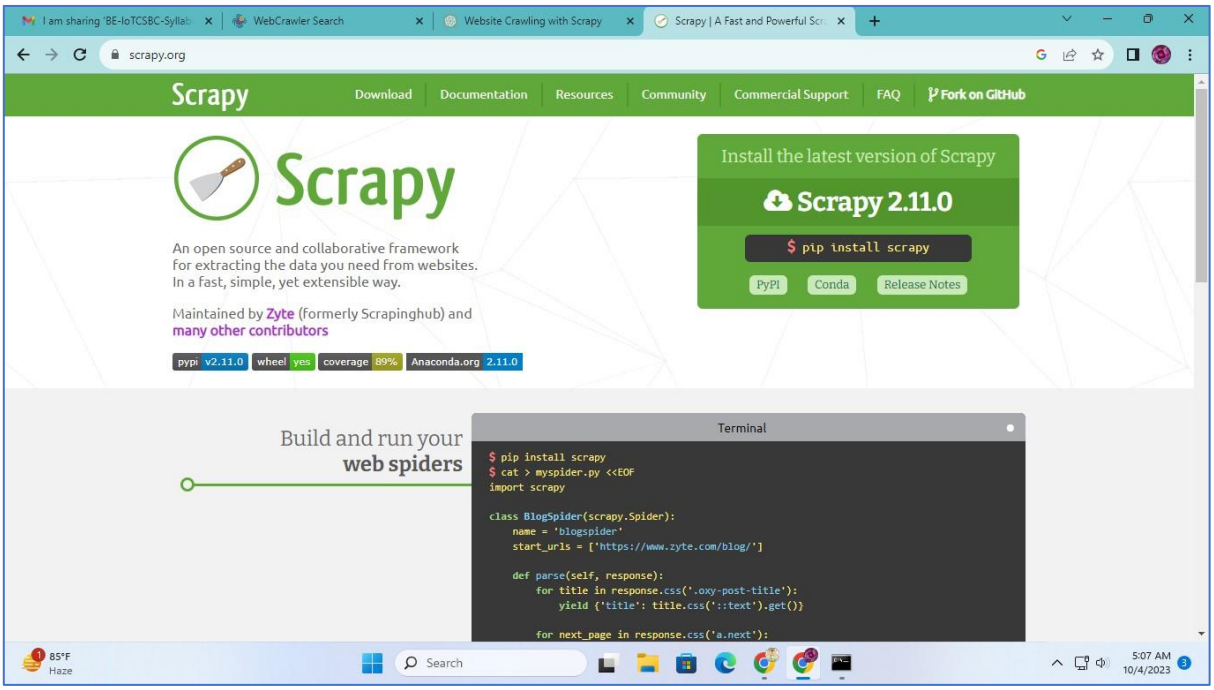
- **Crawler or Spider**: A program or script, known as a crawler or spider, is used to navigate the web and retrieve data from websites. These crawlers follow predefined rules and instructions to access and interact with web pages.
- **URL Discovery**: Crawlers start with one or more initial URLs, often referred to as "seed" URLs. They use these seed URLs to discover other URLs on the website by analyzing the content and following links.
- **Data Extraction**: Crawlers can extract various types of data from web pages, such as text, images, links, metadata, and more. The specific data to extract is defined in the crawler's instructions.
- **Link Analysis**: Crawlers follow internal links on the website, allowing them to navigate to different pages within the same domain. This helps create a comprehensive map of the site's structure.
- **Respect Robots**.txt: Responsible web crawling involves respecting the rules defined in the website's robots.txt file. This file specifies which parts of the site are off-limits to crawlers.
- **Rate Limiting**: To avoid overloading a website's servers and potentially causing disruption, crawlers may employ rate limiting, ensuring that requests are made at a reasonable pace.
- **Data Storage:** Data collected during web crawling is typically stored in a structured format for further analysis, reporting, or indexing. This data can be saved in databases, spreadsheets, or other storage solutions.
- **Continuous Monitoring**: Web crawling can be set up to run periodically to keep the collected data up to date.

Common use cases for website crawling include:

- **Search engine indexing**: Search engines like Google use web crawlers to index and rank web pages.
- **Data scraping:** Extracting data from websites for various purposes, such as competitive analysis, price comparison, or content aggregation.
- **SEO analysis**: Crawling a website to assess its SEO performance, including page load times, broken links, and content quality.
- **Content monitoring**: Keeping track of changes to specific web pages or websites.
- **Market research**: Collecting data from e-commerce websites to analyze products, prices, and trends.

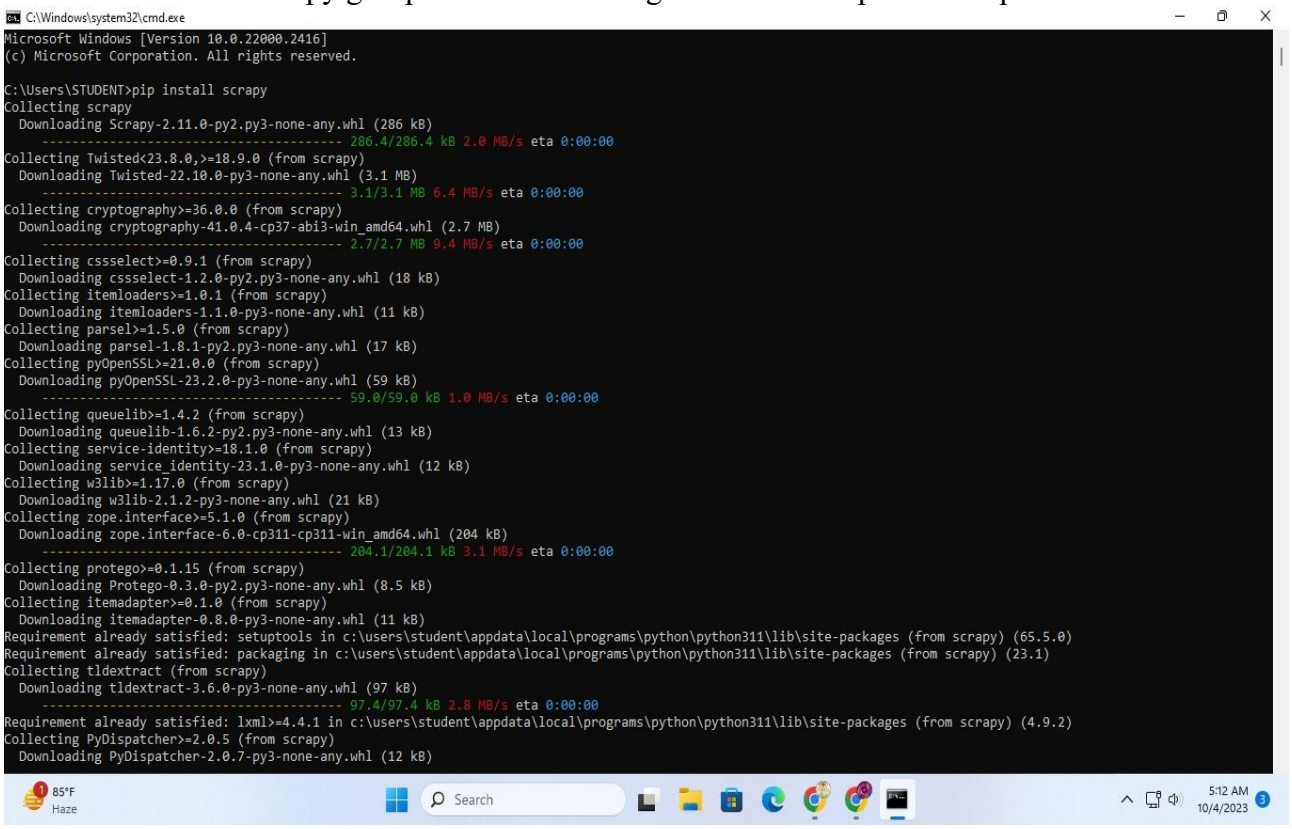
Setting up Scrapy:

- First, you need to install Scrapy using pip: “pip install scrapy”.
- Once Scrapy is installed, you can create a Scrapy project using the command scrapy startproject project_name.



Defining Spiders:

- In Scrapy, spiders are custom Python classes that define how to crawl and scrape data from specific websites.
- You create a spider by defining its start URLs and rules for following links.
- You can use the scrapy genspider command to generate a new spider for a particular domain.



Spider Logic:

- In your spider, you override the start_requests method to specify the initial URLs to crawl.
- You define parsing methods that extract data from the downloaded web pages. These methods typically use XPath or CSS selectors to locate and extract data from the HTML.
- You may also define rules for following links to other pages on the website.

spy

File Edit View

```
import scrapy


class SpySpider(scrapy.Spider):
    name = "spy"
    allowed_domains = ["acpce.org"]
    start_urls = ["https://acpce.org"]

    def parse(self, response):
        pass
```

Sending Requests:

- Scrapy sends HTTP requests to the URLs specified in the start_requests method or discovered during crawling.
- You can customize requests by setting headers, cookies, and other parameters.
- Responses from the server are processed by the parsing methods.

```
C:\Users\STUDENT\osint>scrapy crawl spy
2023-10-04 04:57:42 [scrapy.utils.log] INFO: Scrapy 2.11.0 started (bot: osint)
2023-10-04 04:57:42 [scrapy.utils.log] INFO: Versions: lxml 4.9.2.0, libxml2 2.9.12, cssselect 1.2.0, parsel 1.8.1, w3lib 2.1.2,
.11.4:d2340ef, Jun  7 2023, 05:45:37) [MSC v.1934 64 bit (AMD64)], pyOpenSSL 23.2.0 (OpenSSL 3.1.3 19 Sep 2023), cryptography 41

2023-10-04 04:57:42 [scrapy.addons] INFO: Enabled addons:
[]
2023-10-04 04:57:42 [asyncio] DEBUG: Using selector: SelectSelector
2023-10-04 04:57:42 [scrapy.utils.log] DEBUG: Using reactor: twisted.internet.asyncioreactor.AsyncioSelectorReactor
2023-10-04 04:57:42 [scrapy.utils.log] DEBUG: Using asyncio event loop: asyncio.windows_events._WindowsSelectorEventLoop
2023-10-04 04:57:42 [scrapy.extensions.telnet] INFO: Telnet Password: 2ff1f2e95c110107
2023-10-04 04:57:44 [scrapy.middleware] INFO: Enabled extensions:
['scrapy.extensions.corestats.CoreStats',
 'scrapy.extensions.telnet.TelnetConsole',
 'scrapy.extensions.logstats.LogStats']
2023-10-04 04:57:44 [scrapy.crawler] INFO: Overridden settings:
{'BOT_NAME': 'osint',
 'FEED_EXPORT_ENCODING': 'utf-8',
 'NEWSPIDER_MODULE': 'osint.spiders',
 'REQUEST_FINGERPRINTER_IMPLEMENTATION': '2.7',
 'ROBOTSTXT_OBEY': True,
 'SPIDER_MODULES': ['osint.spiders'],
 'TWISTED_REACTOR': 'twisted.internet.asyncioreactor.AsyncioSelectorReactor'}
2023-10-04 04:57:46 [scrapy.middleware] INFO: Enabled downloader middlewares:
['scrapy.downloadermiddlewares.robotstxt.RobotsTxtMiddleware',
 'scrapy.downloadermiddlewares.httpauth.HttpAuthMiddleware',
 'scrapy.downloadermiddlewares.downloadtimeout.DownloadTimeoutMiddleware',
 'scrapy.downloadermiddlewares.defaultheaders.DefaultHeadersMiddleware',
 'scrapy.downloadermiddlewares.useragent.UserAgentMiddleware',
 'scrapy.downloadermiddlewares.retry.RetryMiddleware',
 'scrapy.downloadermiddlewares.redirect.MetaRefreshMiddleware',
 'scrapy.downloadermiddlewares.httpcompression.HttpCompressionMiddleware',
 'scrapy.downloadermiddlewares.redirect.RedirectMiddleware',
 'scrapy.downloadermiddlewares.cookies.CookiesMiddleware',
 'scrapy.downloadermiddlewares.httpproxy.HttpProxyMiddleware',
 'scrapy.downloadermiddlewares.stats.DownloaderStats']
```

```
2023-10-04 04:57:46 [scrapy.core.engine] INFO: Spider opened
2023-10-04 04:57:46 [scrapy.extensions.logstats] INFO: Crawled 0 pages (at 0 pages/min), scraped 0 items (at 0 items/min)
2023-10-04 04:57:46 [scrapy.extensions.telnet] INFO: Telnet console listening on 127.0.0.1:6023
2023-10-04 04:57:47 [urllib3.connectionpool] DEBUG: Starting new HTTPS connection (1): publicsuffix.org:443
2023-10-04 04:57:47 [urllib3.connectionpool] DEBUG: https://publicsuffix.org:443 "GET /list/public_suffix_list.dat HTTP/1.1" 200 81964
2023-10-04 04:57:47 [scrapy.downloadermiddlewares.redirect] DEBUG: Redirecting (301) to <GET https://www.acpce.org/robots.txt> from <GET https://acpce.org/
2023-10-04 04:57:49 [scrapy.core.engine] DEBUG: Crawled (200) <GET https://www.acpce.org/robots.txt> (referer: None)
2023-10-04 04:57:49 [scrapy.downloadermiddlewares.redirect] DEBUG: Redirecting (301) to <GET https://www.acpce.org/> from <GET https://acpce.org>
2023-10-04 04:57:49 [scrapy.core.engine] DEBUG: Crawled (200) <GET https://www.acpce.org/robots.txt> (referer: None)
2023-10-04 04:57:50 [scrapy.core.engine] DEBUG: Crawled (200) <GET https://www.acpce.org/> (referer: None)
2023-10-04 04:57:50 [scrapy.core.engine] INFO: Closing spider (finished)
2023-10-04 04:57:50 [scrapy.statscollectors] INFO: Dumping Scrapy stats:
{'downloader/request_bytes': 1928,
 'downloader/request_count': 5,
 'downloader/request_method_count/GET': 5,
 'downloader/response_bytes': 36379,
 'downloader/response_count': 5,
 'downloader/response_status_count/200': 3,
 'downloader/response_status_count/301': 2,
 'elapsed_time_seconds': 3.895901,
 'finish_reason': 'finished',
 'finish_time': datetime.datetime(2023, 10, 3, 23, 27, 50, 417304, tzinfo=datetime.timezone.utc),
 'httpcompression/response_bytes': 129355,
 'httpcompression/response_count': 3,
 'log_count/DEBUG': 10,
 'log_count/INFO': 10,
 'response_received_count': 3,
 'robotstxt/request_count': 2,
 'robotstxt/response_count': 2,
 'robotstxt/response_status_count/200': 2,
 'scheduler/dequeued': 2,
 'scheduler/dequeued/memory': 2,
 'scheduler/enqueued': 2,
 'scheduler/enqueued/memory': 2,
 'start_time': datetime.datetime(2023, 10, 3, 23, 27, 46, 521403, tzinfo=datetime.timezone.utc)}
2023-10-04 04:57:50 [scrapy.core.engine] INFO: Spider closed (finished)
```

● Conclusion:

The use of website crawling OSINT tools is a valuable method for systematically gathering data about a website. This data can be crucial for decision-making in the fields of cybersecurity, digital marketing, and web development. It's important to use these tools responsibly, respect the website's terms of service, and ensure that your web crawling activities comply with legal and ethical guidelines.