Digging deeper, OSINT gathering with SCRAPY

Pau Muñoz, dc170

NcNLabs

Updated: 2018/04/06

¿De que hablaremos?

El uso de la plataforma SCRAPY (**introducción**) para recolectar "inteligencia" a través de la www. También hablaremos sobre herramientas y usos concretos para complementar y potenciar SCRAPY.

Scrapy es una plataforma escrita en python usada para recorrer el html de sitios web buscando y eventualmente tratando/almacenando información.

Overview

1. Instalación
2. Definiciones
3. Ejemplos

Instalación

Instalando SCRAPY

pip install Scrapy

Toda la info:

https://doc.scrapy.org/en/latest/intro/install.html

Definiciones

Consultas

Scrapy funciona descargando contenido html a partir de una URL y trabajando con el mismo. Normalmente scrapy usa selectores para buscar información dentro del código (productos, usuarios, datos personales, correos...)

Selectores

Scrapy funciona mediante selectores. Un selector se encarga de "parsear" el HTML del sitio web buscando una expresión (etiqueta, etiquetas anidadas, etiquetas dentro de etiquetas con propiedades...)

Selectores: CSS

CSS sigue el formato:

response.css('.title::text').extract()

Selectores: XPATH

XPATH sigue el formato:

subforums = ’//td/strong/a/@href’

Cíclo

Para parsear un sitio web scrapy empezará en la página principal y usará una función "recursiva" para ir capturando y tratando todos los "HREF" hasta llegar a una condición de parada.

Crawlers

Podemos automatizar el proceso de crawling creando bots que realicen de manera automática el cíclo anteriormente comentado. También podemos usar la shell interactiva para pruebas.

Ejemplos

SCRAPY SHELL

La shell de scrapy nos permite descargar el html de una página e interactuar con él. Útil para probar filtros.

SCRAPY SHELL

```
2018-04-06 02:03:52 [scrapy.middleware] INFO: Enabled item pipelines:
[]
2018-04-06 02:03:52 [scrapy.extensions.telnet] INFO: Telnet console listening on 127.0.0.1:6023
2018-04-06 02:03:53 [root] DEBUG: Using default logger
2018-04-06 02:03:53 [root] DEBUG: Using default logger
[s] Available Scrapy objects:
[s]   scrapy     scrapy module (contains scrapy.Request, scrapy.Selector, etc)
[s]   crawler    <scrapy.crawler.Crawler object at 0x7fd79db72a10>
[s]   item       {}
[s]   settings   <scrapy.settings.Settings object at 0x7fd79db72990>
[s] Useful shortcuts:
[s]   fetch(url[, redirect=True]) Fetch URL and update local objects (by default, redirects are followed)
[s]   fetch(req)                  Fetch a scrapy.Request and update local objects
[s]   shelp()           Shell help (print this help)
[s]   view(response)    View response in a browser
In [1]: fetch("https://www.reddit.com/r/dankmemes/")
```

SCRAPY SPIDERS

Podemos crear spiders y programar todo el proceso de captura y tratamiento de datos.

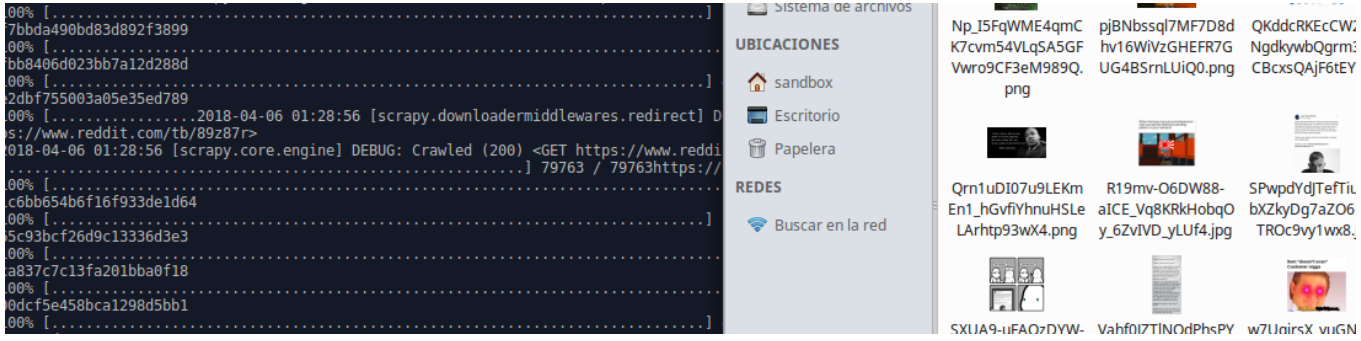
SCRAPY SPIERS

```
start_urls = ['http://www.reddit.com/r/dankmemes//']


def parse(self, response):
    titles = response.css('.title.may-blank::text').extract()
    votes = response.css('.score.unvoted::text').extract()
    times = response.css('time::attr(title)').extract()
    url = response.css('.title.may-blank::attr(href)').extract()
    print "URLS........"
    for u in url:
        print u
    for item in zip(titles,votes,times,url):
        scraped_info = {
            'title' : item[0],
            'vote' : item[1],
            'created_at' : item[2],
            'urls' : item[3],
        }

        yield scraped_info
```

SCRAPY COMPLEMENTOS

Al estar programado en python, podemos complementar nuestros spiders con mil y una librerías. Por ejemplo podemos descargar archivos de manera automática.

SCRAPY COMPLEMENTOS



SCRAPY DEEP WEB

Es muy posible que según que sitio web pueda detectar según que proceso de scraping...Efectivamente, podemos usar scrapy a través de tor y ¡bonus! podemos usar scrapy para explorar tor! (ahmia.fi)

SCRAPY DEEP WEB

```
--------------------------
2018-04-06 01:33:39 [scrapy.core.engine] DEBUG: Crawled (200) <GET http://qzbkws
SUBFORUM: 0day Forum - CVV / Fullz / Dumps
POST TITLE: Mario7777 -Slot Machine Of Dumps-ENJOY-95 % VALID
MAIN URL: http://qzbkwswfv5k2oj5d.onion/thread-10503.html
PAG URL[4]: http://qzbkwswfv5k2oj5d.onion/thread-10503-page-5.html
MD5 HASH OF THE CONTENT: f712d2e4a819ac286e114485077035f3
--------------------------
2018-04-06 01:33:39 [scrapy.core.engine] DEBUG: Crawled (200) <GET http://qzbkws
SUBFORUM: 0day Forum - CVV / Fullz / Dumps
POST TITLE: Buying all your US cvvs bases !! constantly / Покупая все американск
MAIN URL: http://qzbkwswfv5k2oj5d.onion/thread-13853.html
PAG URL[4]: http://qzbkwswfv5k2oj5d.onion/thread-13853-page-5.html
MD5 HASH OF THE CONTENT: 26a2eb706a4e7747b8f3620f1df1778b
--------------------------
2018-04-06 01:33:39 [scrapy.core.engine] DEBUG: Crawled (200) <GET http://qzbkws
SUBFORUM: 0day Forum - CVV / Fullz / Dumps
POST TITLE: http://myccroom.com .ru .pro NEW SHOP BEST CC/CVV2 and DUMPS
MAIN URL: http://qzbkwswfv5k2oj5d.onion/thread-9228.html
PAG URL[3]: http://qzbkwswfv5k2oj5d.onion/thread-9228-page-4.html
MD5 HASH OF THE CONTENT: 1a762cd8690659b7bd1aae80ae36c197
```

Material

- github.com/dc170/osintgathering-Scrapy

THE END