UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE – UFRN

DEPARTAMENTO DE COMUNICAÇÕES

ENGENHARIA DE TELECOMUNICAÇÕES

PROGRAMAÇÃO ORIENTADA A OBJETO

WEB CRAWLER

Eriberto de Souto Silva
Victor Costa

2 INTRODUÇÃO

- O que é Web Crawler?
- Qual a sua utilidade ?
- Como ele funciona ?
- Aplicações?
- Exemplo códificado?

3 O QUE É WEB CRAWLER?

- **Crawler**, também conhecido como Spider ou Bot, é um robô usado pelos buscadores para encontrar e indexar páginas de um site.
- Um **rastreador da rede**, em inglês *web crawler*, é um programa de computador que navega pela rede mundial de uma forma metódica e automatizada. (wikipédia)

4 QUAL A SUA UTILIDADE ?

- Os Web Crawlers são principalmente utilizados para criar uma cópia de todas as páginas visitadas para um pós-processamento por um motor de busca que irá indexar as páginas baixadas para prover buscas mais rápidas.
- Crawlers também podem ser usados para tarefas de manutenção automatizadas em um Web Site, como checar os links ou validar o código HTML. Os Crawlers também podem ser usados para obter tipos específicos de informações das páginas da Web, como minerar endereços de email (mais comumente para spam)

5 COMO ELE FUNCIONA ?

- Ele captura informações das páginas e cadastra os links encontrados, possibilitando encontrar outras páginas e mantendo sua base de dados atualizada.
- Os dados coletados podem ser analisados e posteriormente alguma ação pode ser feita com esses dados.

6 EXEMPLOS DE WEB CRAWLERS

- **Yahoo! Sluro** é o nome do Crawler do Yahoo!
- **Msnbot** é o nome do Crawler do Bing – Microsoft.
- **Googlebot** é o nome do Crawler do Google.
- **Methabot** é um Crawler com suporte a scripting escrito em C.
- **Arachnode.net** é um Web Crawler open-source usando a plataforma .NET e escrito em C#
- **DuckDuckBot** é o Web Crawler do DuckDuckGo.

7 BIBLIOTECAS

- Existem muitas maneiras de se criar um crawler utilizando bibliotecas prontas disponíveis.
- Biblioteca Request
- Urllib2
- BeautifulSoup
- Framework Scrapy

8 BIBLIOTECA REQUEST

- A biblioteca *requests* fará uma solicitação *GET* ao servidor, que fará o download dos conteúdos HTML da página solicitada para nós.

```
import requests

page = requests.get("http://dataquestio.github.io/web-scraping-pages/simple.html")
page
```

9 BIBLIOTECA BEAUTIFULSOUP

- É a ferramenta que retira informações de uma página web, podendo extrair tabelas, listas, parágrafos ou filtrar informações de páginas.

```
from bs4 import BeautifulSoup
soup = BeautifulSoup(page.content, 'html.parser')
```

```
print(soup.prettify())
```

```
<!DOCTYPE html>
<html>
 <head>
  <title>
   A simple example page
  </title>
 </head>
 <body>
  <p>
   Here is some simple content for this page.
  </p>
 </body>
</html>
```

10 URLLIB2

- O urllib2 é um módulo do Python que define funções e classes(Objetos) que manipulam URL(s), seja ela uma URL simples baseado no protocolo HTTP, autenticação Digest, envio de dados GET e POST download de arquivos, cookies, sessão etc.

11 FRAMEWORKS SCRAPY

- Mecanismo de extração de conteúdo das páginas (*scraping*), e de **navegação** de páginas relevantes para a extração (*crawling*) dos dados, utilizando o conceito de Spider.
- Um *Scrapy spider* é responsável por definir como seguir os links “navegando” por um site (o que chamamos de *crawling*) e como extrair as informações das páginas em estruturas de dados Python.

12 FRAMEWORKS SCRAPY

- O Scrapy propõe que você crie algumas classes que representem os itens que você pretende extrair das páginas. Por exemplo, podemos extrair os preços e detalhes de produtos de uma loja virtual, que irão representar uma classe.

13 VULNERABILIDADE XSS

- Uma vulnerabilidade XSS é o que acontece quando um site exibe a entrada do usuário.
- Um site com essa vulnerabilidade, pode receber diversos tipos de ataque com base na relação de confiança entre o usuário e a plataforma, como redirecionar para outro site, com intuito de roubar informações, ou até mesmo baixar alguma ameaça para quem seja executada no sistema.

14 MÉTODOS DE PREVENÇÃO

- Encoding e Validation
- Bibliotecas Anti-XSS
- Utilizar o Content Security Policy (CSP)
- Usando a flag HttpOnly
- X-XSS-Protection no Header
- Biblioteca XSScrapy

BIBLIOTECA XSSCRAPY

- XSScrapy é um scanner XSS que funciona usando o scrapy para criar uma aranha da web para baixar o HTML de todas as páginas em um determinado nome de domínio.
- A Scrapy encontra URLs seguindo automaticamente todos os links no site até que tenha verificado cada URL. Uma vez que o HTML para uma página é baixado, o XSScrapy pesquisa automaticamente a página para vulnerabilidades do XSS.

16 BIBLIOTECA XSSCRAPY

- Isso é feito procurando por uma série de pontos comuns de injeção e injetando a corda
- 9zqjxel'"(){}<x>:9zqjxel;9

BIBLIOTECA FIMAP

- Fimap é uma pequena ferramenta programado em python que pode encontrar, preparar, auditar e explotar automaticamente erros de Remote File Inclusion em aplicações web.

FIMAP: CARACTERÍSTICAS

- Verifica um único URL, lista de URLs ou resultados do Google de forma totalmente automática
- Pode identificar e explorar os erros de inclusão de arquivos.
- Tenta automaticamente elevar sufixos com Nullbyte e outros métodos como Dot-Truncation
- Injeção de arquivo remoto.
- Injeção de logfile. (FimapLogInjection)
- Teste e explore vários erros:

FIMAP: CARACTERÍSTICAS

- Tem um modo de exploração interativa que:
- - pode gerar um shell em sistemas vulneráveis.
- - pode gerar um invólucro reverso em sistemas vulneráveis.
- - pode fazer tudo o que você adicionou no seu payload-dict dentro do config.py
- Adiciona suas próprias cargas e caminhos ao arquivo config.py.
- Tem um modo de colheita que pode coletar URLs de um determinado domínio para posterior pentesting.

FIMAP: CARACTERÍSTICAS

- Pode usar proxys.
- Digitaliza as variáveis GET e POST.
- Tem uma pegada muito pequena.
- Pode atacar também servidores do Windows! (WindowsAttack)
- Tem uma pequena interface de plugin para escrever plugins de exploitmode (PluginDevelopment)
- Cookie and Header digitalização e exploração.

21 REFERÊNCIAS

- https://www.edools.com/automatizando-tarefas-com-web-crawlers-mechanize/
- https://imasters.com.br/desenvolvimento/aprendendo-sobre-web-scraping-em-python-utilizando-beautifulsoup/?trace=1519021197&source=single
- https://www.crummy.com/software/BeautifulSoup/bs4/doc/#kinds-of-objects
- https://www.vooo.pro/insights/guia-para-iniciantes-de-web-scraping-em-python-usando-beautifulsoup/
- http://pythonclub.com.br/material-do-tutorial-web-scraping-na-nuvem.html
- http://www.gilenofilho.com.br/usando-o-scrapy-e-o-rethinkdb-para-capturar-e-armazenar-dados-imobiliarios-parte-i/
- http://www.nacaolivre.com.br/python/python-usando-urllib2/
- http://kaoticcreations.blogspot.com.br/2011/08/automated-lfirfi-scanning-exploiting.html