TOOL TO ANALYZE CYBER PRESENCE OF INDIVIDUALS ON PUBLIC INTERNET

Capstone Project – Summer Semester, 2017

Technical Design Report

Contributors

Project Advisor:

Vance McMillan Saunders, M.S.
Computer Science
Instructor
Wright State University

Project Designer:

Survya Pratap Singh
M.S. Cyber Security
Student
Wright State University
Singh.143@wright.edu

Date: July 25, 2017

Department of Computer Science and Engineering
Wright State University
Fairborn, OH 45324

Contents

Abstract

In the era of technology, where having one's virtual identity is as important as their physical identity, more and more people are getting connected to various networking sites for maintaining their online profile. Most of the internet users share their entire life online either knowingly or unknowingly, thus making these websites a rich repository of valuable personal data. These sites allow users to publish details about themselves and their lives and connect to their friends and colleagues. Commonly users do not think or are not even aware of the risks when they share something online. The availability of personal information online is an opportunity for identity thieves, scam artists, and cyber stalkers to use the information that people themselves have voluntarily provided in a way harmful for the owner of the information [1].

We all scatter small bits of our personal information on several websites, but since it is spread out and requires effort to access, this affords us a certain level of privacy. Aggregating all this information into one place and making it so easily accessible can bring up serious privacy concerns. While information present on these websites may not create completely new cyber threats, but they do substantially amplify the risk of existing ones. This scattered information present on websites is becoming an ideal source for gathering information or performing reconnaissance for the targeted individual and it may make you feel that you are always being watched by someone you do not know [2].

The paper discusses the detail design and development of a tool, which will have the capability of aggregating the individual's online scattered information and educate them about their cyber presences and cyber-attacks or scams that can be launched using aggregated information.

Introduction

The probability of carrying out the successful cyber-attack depends very much on the quality of reconnaissance or data gathered against the target. There are tons of social media websites, which are being used by individuals for sharing information either knowingly or unknowingly, which has a potential to compromise their privacy. There are around 1.6 billion social media network users worldwide and it is one of the most popular ways for the online user to spend their time, share information and stay connected with friends and families [3]. The presence of private data on social media sites or less secured websites has motivated cyber criminals to make these websites as their

hunting ground for targeting the individuals by launching cyber-attacks such as spear-phishing, social spam, link-jacking, and like-jacking [2].

Most of the well-known companies have started using the data available on social media sites to track the personal and social behavior pattern of individuals and provide them with the targeted advertisement and offer them with improved personalized content, which in privacy aspect is not ethical [1]. One of the well-known professional social networking website called as LinkedIn, where user not only provide information about themselves but also reveal critical information about company and technology being used. Cybercriminal, use this information to perform reconnaissance or information gathering for launching cyber-attack against targeted companies. Online large-scale networks (e.g. LinkedIn, Facebook, and Twitter) have raised many important privacy issues, as employees can use this SNSs to store some sensitive and confidential data [2].

Disclosing and Sharing private information is a necessity to be visible on SNS (Social networking sites), but at the same time, malicious attacks can be largely facilitated using this freely available data [3]. It seems that the awareness of these threats is very low among users and, generally, users do not really care about the possible implications and risks associated with data sharing. This era has witnessed an increase in the number of security incidents related to the exploitation of the human factor on SNSs [2]. The technique most widely used by attackers that focuses on the human element is commonly referred as a social engineering attack. It is defined as the art of deceiving or tricking people to help attackers reach their goals, to gain information from them, or to persuade them to perform an action that will benefit the attacker in some way. Phishing is a form of social engineering in which the attacker attempts to fraudulently retrieve legitimate users' confidential or sensitive information by mimicking electronic communication from a trustworthy source. The publicly available data has motivated a cyber attacker to launch targeted phishing attack called as spear phishing attack.

Having a tool or software, which can educate the individual about their cyber presence on public internet will help them to decide what not to share on the internet. Even though there are various online portals and applications such as peekyou.com, beenverified.com, and Maltego, which has some capability of analyzing an individual information present publicly on the internet, but those services are either paid or requires a license to operate thus normal internet user feels hesitant to

use them. The paper discusses Google web crawling approach to crawl the public websites and how using google search one can aggregate the data regarding specific individuals.

The aim is to build a user-friendly and easy to use the tool, where all the aggregated data will be presented and provide to the individual with the ability to see their publicly and easily available information.

Problem Statement

Internet users have the habit of scattering information online, without paying much attention to the cyber risk associated with the shared information. Sharing information on less secure websites or website which may leak shared information, poses a greater cyber threat and contributes to increasing the overall cyber-attack surface area of an individual [2].

Googlebot, which is a Google Search engine web crawler [17], crawl the publicly available websites and its content for indexing the web pages and for providing the users with accurate search results. Most of the websites do not have well defined "robot.txt" file and will allow Googlebot to crawl all the information present on the website, thus sharing information on such websites exposes your information to be publicly available with google search [17]. Google also provides the specialized search technique called as "Google dorks" for finding the specific information based on the keywords and it mostly used by hackers for performing reconnaissance [15] activity for their target.

As this information is scattered in bits and pieces over the internet and required efforts to search the content, this provides the user with a certain level of privacy. But if all the scattered information is aggregated into one place can bring up serious privacy concerns.

Providing the users with simple and easy to use application, with the capability of aggregating all the publicly shared content at one place and presenting it to the individuals for educating them about their cyber presence.

Literature Review

The internet search engines such as Google and Yahoo are designed for searching the accurate result available on the internet. To achieve the search result accuracy, they use the web crawler to crawl the publicly available website on the internet and store the content available on those

websites. A Web crawler is a program that is tasked for browsing the World Wide Web in a methodical, automated manner or in an orderly fashion. Web crawling is an efficient way for collecting the data and keeping up with, the rapidly expanding Internet [9], as huge number of new websites are added daily on the internet. Web crawlers follow the 4 sets of policies for crawling the publicly available websites. Selection policy which states what pages has to be downloaded from the websites, re-visit policy which defines the rechecking of web pages for new changes, politeness policy which states how to avoid overloading the websites and a parallelization policy which describe the coordination between distributed crawlers[9].

Googlebot, which is a web crawling bot used by Google, sometimes also referred as a spider, is used for crawling the new and updated web pages to be added to the Google search index[10]. Googlebot uses a huge set of distributed computers or bots for fetching up the billions of pages available on the web. It uses the specialized algorithm process for deciding which sites to crawl, how often websites should be crawled and how many pages to fetch from each website. Websites allow this crawler to crawl their websites so that they are easily assembled to the user by search engines. Googlebot's crawl process begins with a list of web page URLs, generated from previous crawl processes and augmented with Sitemap data provided by webmasters. As Googlebot visits each of these websites it detects links (SRC and HREF) on each page and adds them to its list of pages to crawl. New sites, changes to existing sites, and dead links are noted and used to update the Google index [10]. The Google crawler consists of five functional components running in different processes. A URL server process reads URLs out of a file and forwards them to multiple crawler processes. Each crawler process runs on a different machine, is single-threaded, and uses asynchronous I/O to fetch data from up to 300 Web servers in parallel. The crawlers transmit downloaded pages to a single Store Server process, which compresses the pages and stores them to disk. The pages are then read back from disk by an indexer process, which extracts links from HTML pages and saves them to a different disk file. A URL resolver process reads the link file, derelativizes the URLs contained therein, and saves the absolute URLs to the disk file that is read by the URL server.[7]. There are several ways which can be used by the website for controlling the behavior of crawler and prevent it from accessing private links or private data. “Robot.txt” is a file which contains the information about pages which should and should not be crawled by the spiders[18]. Including robot meta tag in the head section of the HTML can prevent spiders from crawling it[18].

```
<!DOCTYPE html>
<html><head>
<meta name="robots" content="noindex" />
(…)
</head>
<body>(…)</body>
</html>
```

Websites lacking proper setup of "robot.txt" file are vulnerable to data leakage, as all of its content would be getting crawled by the spiders. Sharing of the phone numbers, email or address on such websites will expose it to the entire Internet user as that information will be crawled by search engines.

Search engines are a rich repository of publicly available information over the internet, and with a proper search, technique attacker can easily gather information about their targets. Google Dorks is a technique for pin pointing the exact search result based on the keywords and has the ability to uncover some of the incredible information such as phone number, email id and address and other critical files present publicly on the internet[15]. Google dorks have several operators which are used craftily by attackers to gather scattered information about their targets. Searching anything on google with double quotation mark will search for that specific keyword on the entire internet and will give you the more accurate results[15].

One can imagine that what attacker is capable of with all the information that has been crawled by the search engine spider and the specialized search technique. The same technique has been implemented in the designing of my tool, which has the capability of running the specialized search queries on the Google search engine to find information about particular individuals.

Tool Design Approach

The design approach of this tool is pretty simple, this tool minimizes the effort of aggregating the scattered information, about the particular individual over the internet and present it to the user in an easy to read format.

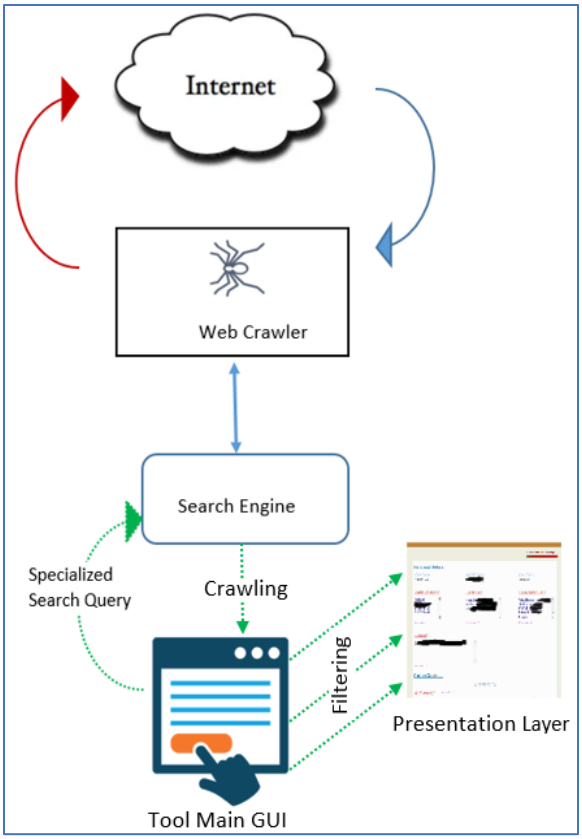

Fig:1- High-Level Application Design

As most of the information is already being crawled by search engines spiders, performing a specialized search query on the search engine, will give more accurate search result about individuals. The tool utilizes this idea and instead of crawling the entire internet for finding the user information it crawls the search result provided by the search engine. Once the search result is crawled tool will perform data filtering based on the user input and will aggregate the required information. Once the information is gathered it displays the information to the user in an easy to read format. It also allows the user to save the gathered result in HTML format for future reference.

Technology used

This is a GUI based tool and is entirely built in Python 3.4, it also utilizes various Python API for achieving the desired result[16]. Below is the list of Python API and their usage in the tool development.

- Requests: This is an API for sending HTTP/HTTPS request to the server and based on the requests it receives the response from the server[20]. Most of the search engine is designed in such a way that, if HTTP request is sent using the code, it will get discarded by search engine server and response of 403 will be received. To overcome this challenge, I have used a predefined header with browser information, which is appended to HTTP request. Below is the snippet of HTTP request being sent by using Requests API.

  r = requests.get(URL,headers={"User-Agent":"Mozilla/5.0 (X11; Linux i686) AppleWebKit/537.17 (KHTML, like Gecko) Chrome/24.0.1312.27 Safari/537.17"})
- Beautiful Soup: It is a Python library and is used for extracting the data from HTML or XML parsed file[5]. Once the tool receives the HTML response from the server, for the searched query, it gets parsed by Python lxml API. Beautiful soup provides the functionality of navigating, searching, and modifying the parse HTML tree based on tags [5].
- Lxml: It is used for parsing the HTML response, is the first Python XML library that demonstrates high-performance characteristics and includes native support for XPath 1.0, XSLT 1.0, custom element classes, and even a pythonic data-binding interface. It is built on top of two C libraries: libxml2 and libxslt[19].
- Tkinter: This is the Python module for creating the graphical user interface[13]. This module is supported by both Unix and windows based operating system
- Pillow: It's an image library, used in the tool for processing the image and converting it to base64 string for optimizing the storing of the image by the tool. The tool is designed in a way that images are never stored on the user's system[14].
- Yattag
- It's a Python API for generating HTML file from Python code. It is used in the tool for saving the gathered information in the form of HTML file[12].

This tool has a total of six classes, designed for specific functionality. Below is the brief description for each class.

- MainAppClass.py: This is the main class of the project, which has the main method. The program starts its execution by calling the main method of this class.
- CreateAppGUI.py: This is the class which contains a method for all the graphical user interface and its functionality.

- CustomeNotebook.py: This class also deals with GUI and is responsible for creating TAB based window frame in the application.
- Animation.py: This class is required for displaying the animated part of the application such as GIF images.
- BackEndProcess.py: This is main class for crawling, parsing, filtering and displaying the user information.
- AppVariables.py: It has information about all the important variables of the application.

The entire development of the tool can be categorized into 3 different parts which are Information Gathering, Information parsing, and presentation.

Information Gathering Approach

Information gathering is a 4 step process starting from user input to fetching up the data from Google search engine.

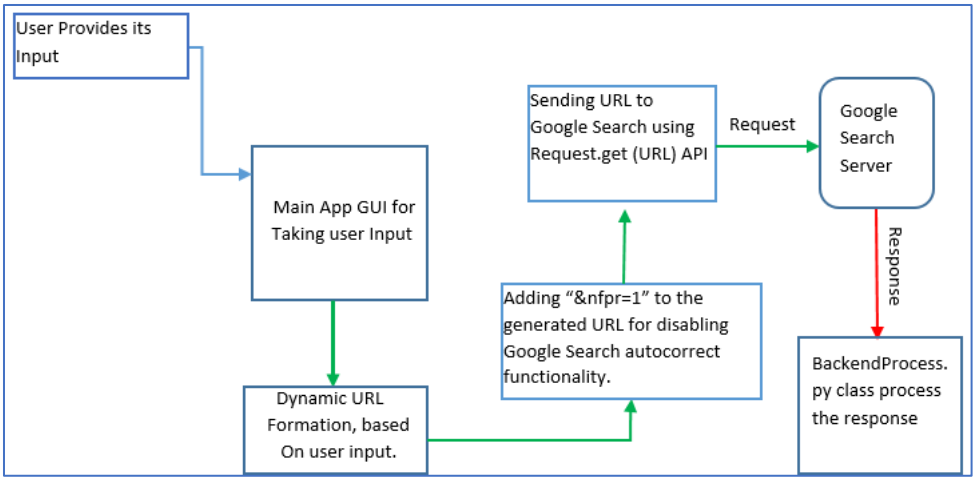

Fig:2- Information Gathering steps.

User Input & GUI

There are two main GUI windows created for this tool. The first GUI is where the user will provide the input or the details of the individual to be searched.First name and last name are mandatory fields which have to be provided and rest of the field is optional. Based on the provided input, the tool will create dynamic Google search URL and will send HTTPS request to Google Server.

Gathered information is presented in the separate GUI windows, where the information will be distributed into different categories and based on the gathered information tool will display the cyber presence of individuals.

URL formation

Based on the information provided by the user, the application will create a maximum of 11 unique search URLs and a minimum of 1 URL[21]. The data provided by the user is appended to google base URL which is 'https://www.google.com/search?q=' along with quotation mark around the data. Ex. User Data URL - https://www.google.com/search?="firstname+LastName"&nfpr=1
Ex. Image URL -
https://www.google.co.in/search?q="FirstName+LastName"&source=lnms&tbm=isch&nfpr=1
Adding double quotation marks around the user provided data will search the entire internet for a specific keyword. "&nfpr=1" is added at the end of the URL for disabling the Google search auto correct feature[21].

Request and Response

After the URL formation is done requests.get() API is used along with browser header information to send https get request message to google server. Google search engine will search for a specific keyword as URL contains double quotes("user input data") around the user data. If the request is successful, the server will send the first page of google search to the application as a https response message[20].

Information Parsing and Filtering approach

Data Parsing Searched Result

Once the response is received from the server, it is parsed to HTML template by using HTML.parser API of python and the parsed HTML is used by beautiful soup object for extracting the content. During my research, I found that there is 3 main HTML tags/class(r, cite, span) which is present in google search source code and can be used by beautiful soup for iterating over that classes to get the entire content of the search result. Below is the snapshot for google search in the browser and its source content. The tool utilizes the source code for fetching up the data for the searched result[5].

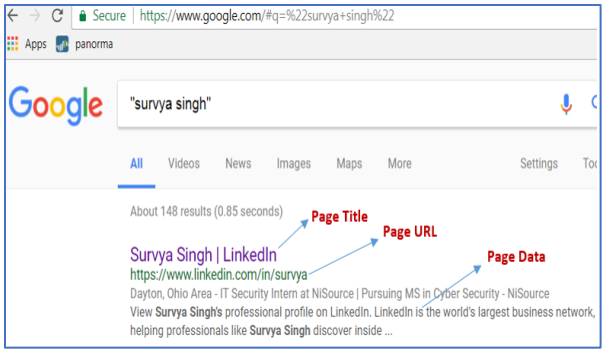

Fig: 3- Google Search Result in Browser

Fig 4- Google Search Source Code

Data Filtering Content

After parsing the search result, the beautiful soup is used for iterating over the content based on the "class=r", 'cite' and 'span' tags. The title, URL, and data for each of the iterated result are passed to ParseFirstHandInfoI() method for filtering the content. This method will search for a maximum of 12 and minimum of 5 unique combinations of first name, middle name and last name in the result and based on it the result will be filtered.

Images Parsing

Parsing the images URL from Google search source code is little different from parsing the data. Google does not have images URL as a link in the source code, but it does provide a metadata about images which contains all the information about the image. Below is the source code displaying the metadata in Google source code, which was used by beautiful soup to iterate over the parsed HTML result.

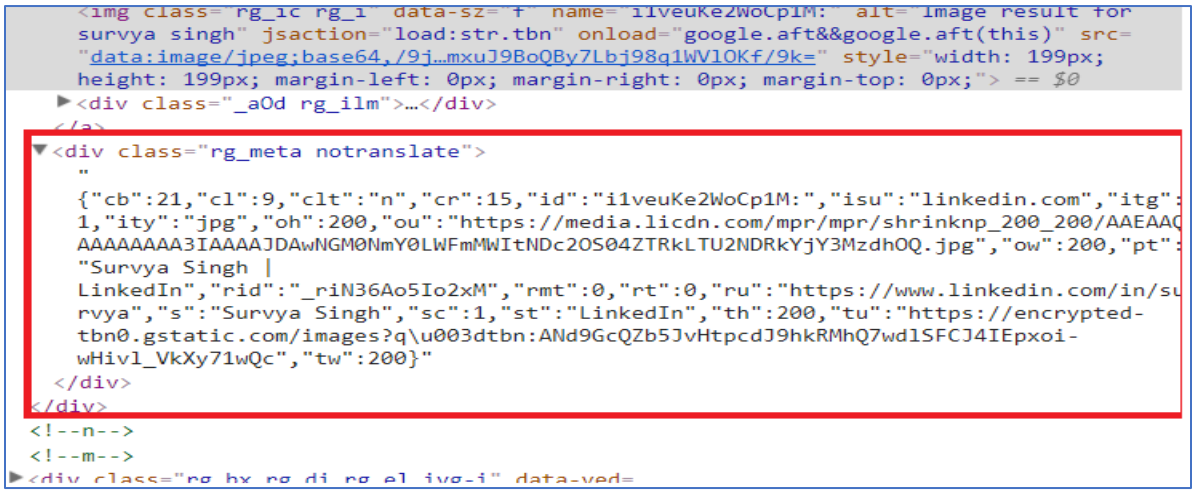

Fig.4 – Image Metadata used in parsing from Source Code

Once the image URL has been fetched, the tool uses requests[20] and pillow[14] API for reading the image from the captured link and store it in the form of base64 encoded string. The tool does not store the images on the system, which makes it more efficient and faster while processing the images from base64 strings.

Link Criticality

Python regular expression has been used for determining the criticality of links, based on the content provided by that data. There is 3 method which checks for the presence of phone number, email id, and address in the data provided by a link.

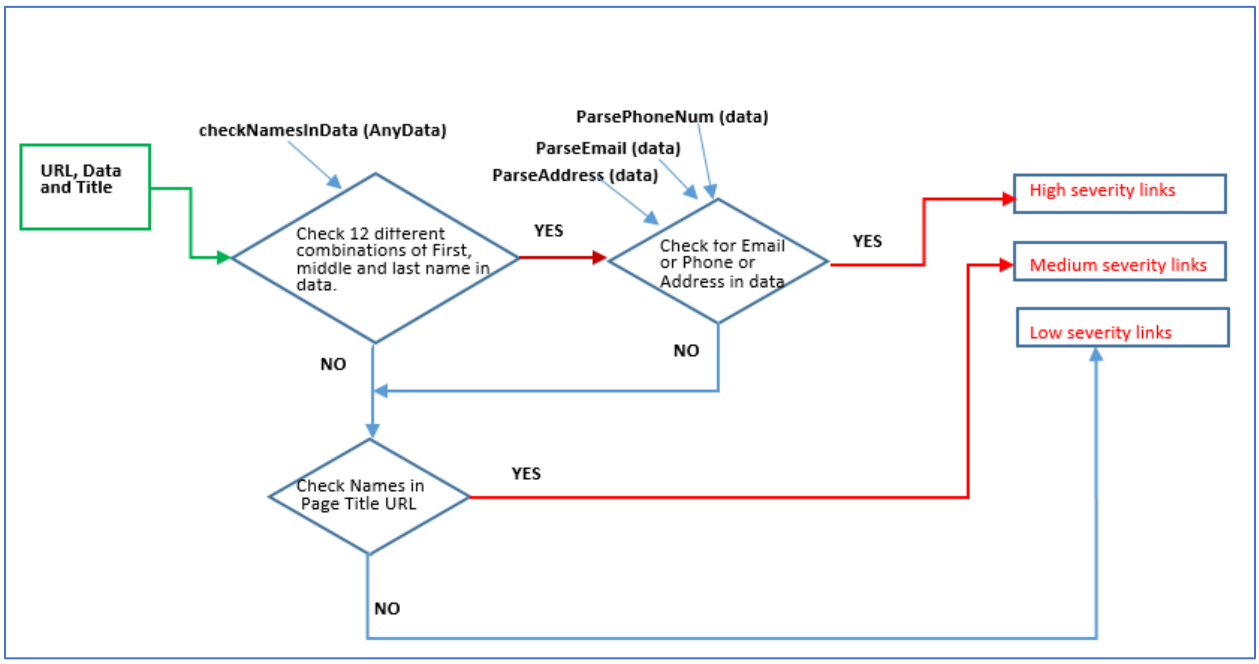

Fig. 5- Deciding Link Severity algorithm

As we can see in Fig.5, that first the data is checked for the presence of first, last and middle name with 12 different combinations, if a match if found, it will be again checked for the presence of email, phone number, and address. If that 3 information is detected in the data, the link will be considered as the high severity link. If a match is not found then, the title of the link will be checked for the presence of an individual name and if a match is found link will be considered as medium severity link. If there is no match for email, phone, address, name either in data or in the title, the link will be considered as the low severity link. Links are stored in 3 different dictionaries with the pair of key: value to avoid the presence of a duplicate link.

Information Presentation approach

Personal Information

This part of the section will display user first, middle and last name along with phone number, email ID's, address and individual social media user-id. If information like phone number, email id, and address is not found during the search process, "N/A" will be displayed.

Online Content

This part will have information about link severity which is high, medium and low. Data from high and medium severity links will be displayed to the user. Images which were found during the search are decoded from base64 string and are displayed, this tool is designed to display only 16 images, but it will display all the images links it found during the search.

Profile Status

There is 5 profile status, which tells the user about their cyber presence which is Very-low, low, medium, high and very-high.Very-high status is given when phone, email id, and address is detected In the data, high status is assigned when either phone/email/address is detected, medium is assigned when there is more than 2 high severity link, medium is assigned when there is more than 5 medium severity link, low is assigned when there is more than 6 low severity link, if none of this condition satisfies very-low status is assigned to user profile.

Storing Result to File

Once the aggregated information is displayed to the user, the tool provides the functionality of saving the aggregated result in the system. At present it supports the saving of file in HTML format, the tool uses Yattag API[12] for generating the HTML file in python. Generated HTML file will not contain images, but the links will be provided in the file.

Future Enhancement

This tool can be considered as a framework which can be expanded further for fetching data from other search engines like Yahoo and Bing. Fetching data from some of the well known social media websites like LinkedIn, Facebook and Twitter will have potential to gather more information about a particular individual. Giving the user the functionality of searching the images of an individual based on provided pictures can also be added, which will require face recognition technique implementation.

Adding the functionality of showing the demo of spear-phishing attack by using the gathered information, will educate user with the cyber risk associated with the publicly available content on the internet.

Conclusion

As an internet user, people have a habit of scattering bits and pieces of information on a large number of websites, without having any knowledge of security feature implemented by that particular website. Scattered information over the internet doesn't seem to have cyber-risk associated with it, but when the same information is aggregated at one location it does increase the surface area of cyber risk. Cyber attacker or criminals gathered this scattered information from the internet during the reconnaissance phase, and use if for launching a successful spear-phishing[2] attack against their target. Even if the user wants to know about their cyber presence or information they have scattered over the internet, they have to invest a huge amount of time in searching the content.

Providing the user with a GUI based user-friendly tool, which can search the scattered information present on the internet, and aggregate it at one place which can be used to educate them about their cyber presence on the public internet. Based on the aggregated information user can take action to remove critical information such as email, address or phone number from the data leaking websites. Even though this tool will search the publicly available data about the user from the white internet, but the user should always be cautious scattering information all over the internet, especially on the unsecured website.

References

[1] Yaniv Altshuler, 1, 2 Nadav Aharony, 1 Yuval Elovici, 2, 3 Alex Pentland, 1 and Manuel Cebrian1, 4. Stealing Reality: When Criminals Become Data Scientists (n.d.): n. pag. Web.

[2] Mario Silic. The dark side of social networking sites: Understanding phishing risks (n.d.): n. pag. Web. <http://www.sciencedirect.com/science/article/pii/S0747563216301029>.

[3] Raymond Heatherly. Preventing Private Information Inference Attacks on Social Networks (n.d.): n. pag. Web. <http://ieeexplore.ieee.org/abstract/document/6226400/>.

[4] https://www.digitalocean.com/community/tutorials/how-to-crawl-a-web-page-with-scrapy-and-python-3

[5] Beautiful Soup Documentation [https://www.crummy.com/software/BeautifulSoup/bs4/doc/]
[6] David Rosenblum. What Anyone Can Know: The Privacy Risks of Social Networking Sites (n.d.): n. pag. Web. <http://ieeexplore.ieee.org/document/4218550/>.
[7] Allan Heydon. Mercator: A Scalable, extensible Web crawler (n.d.): n. pag. Web. http://dl.acm.org/citation.cfm?id=598733
[8] Mike Thelwall. A web crawler design for data mining (n.d.): n. pag. Web. <http://journals.sagepub.com/doi/abs/10.1177/016555150102700503>.
[9] S.S. Dhenakaran1 and K. Thirugnana Sambanthan2. "WEB CRAWLER - AN OVERVIEW." (n.d.): n. pag. Web. <http://www.csjournals.com/IJCSC/PDF2-1/Article_49.pdf>.
[10] Googlebot. Web. https://support.google.com/webmasters/answer/182072?hl=en
[11] https://www.lifewire.com/remove-personal-information-from-internet-3482691
[12] Html template generator Yattag API https://pypi.python.org/pypi/yattag
[13] Tkinter API https://docs.python.org/2/library/tkinter.html
[14] PIL API https://pillow.readthedocs.io/en/4.2.x/
[15]Understanding Google Dorks and How Hackers Use Them. Web. https://www.hackingloops.com/google-dorks/
[16] Python 3.4 documentation. Web. https://docs.python.org/3.4/
[17]Crawling and Indexing: The Web. https://www.google.com/intl/es419/insidesearch/howsearchworks/crawling-indexing.html
[18] Robots meta tag and X-Robots-Tag HTTP header specifications. Web. https://developers.google.com/search/reference/robots_meta_tag?hl=en
[19]LXML HTML parser. Web. https://pypi.python.org/pypi/lxml/3.4.4
[20]Python Requests API. Web. http://docs.python-requests.org/en/master/
[21]Google Search Parameter. Web http://yoast-mercury.s3.amazonaws.com/uploads/2007/07/google-URL-parameters.pdf