Lab – Find Social Media Accounts Using Sherlock

Overview

In this lab, you learn how to find social media accounts using a person's username. With Sherlock, we can instantly hunt down social media accounts created with a unique screen name on many online platforms simultaneously.

Lab Requirements

- **One virtual install of CIS Linux**
  - **Or….**
- **One Virtual Install of Kali Linux.**
- **Internet connection.**

For this lab demonstration, since Sherlock comes preinstalled with CSI Linux, I will be using CSI Linux. If you prefer to use Kali Linux, you will first need to install Sherlock using the following commands.

From your Kali desktop, open a terminal, and at the prompt, type the following command.

```
git clone https://github.com/sherlock-project/sherlock.git
```

Change directory location over to the newly downloaded directory.

```
cd sherlock
```

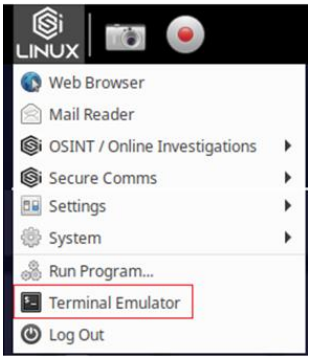
Build the application using pip3.

```
sherlock$ pip3 install -r requirements.txt
```

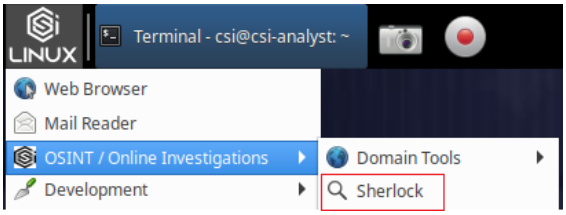
```
~$ git clone https://github.com/sherlock-project/sherlock.git
~$ cd sherlock
~/sherlock$ pip3 install -r requirements.txt
```

The rest of the lab will be shown using CSI Linux, but the steps shown, work with Kali Linux.

From your CSI desktop, click on the application quick launch, and from the context menu, Launch a new terminal.



We could use Sherlock's shortcut in the application directory marked OSINT/Inline Investigation, but it will still open a terminal.



At the prompt, type, **`cd /opt/sherlock`**

Press enter.

```
csi@csi-analyst:/$ cd /opt/sherlock
csi@csi-analyst:/opt/sherlock$
```

At the prompt type, **`ls`** to see the files and directories that are available. Note there is another subdirectory we need to have access to.

```
csi@csi-analyst:/$ cd /opt/sherlock
csi@csi-analyst:/opt/sherlock$ ls
CODE_OF_CONDUCT.md  images              README.md           sherlock
CONTRIBUTING.md     infosecwriter.txt   removed_sites.json  site_list.py
docker-compose.yml  LICENSE             removed_sites.md    sites.md
Dockerfile          __pycache__         requirements.txt    src
csi@csi-analyst:/opt/sherlock$
```

At the prompt, type, **`cd sherlock`** to change directory location to the folder marked, **`sherlock`**.

Press enter.

```
csi@csi-analyst:/opt/sherlock$ ls
CODE_OF_CONDUCT.md  images              README.md           sherlock
CONTRIBUTING.md     infosecwriter.txt   removed_sites.json  site_list.py
docker-compose.yml  LICENSE             removed_sites.md    sites.md
Dockerfile          __pycache__         requirements.txt    src
csi@csi-analyst:/opt/sherlock$ cd sherlock
csi@csi-analyst:/opt/sherlock/sherlock$
```

Again, at this prompt, type ls to see the contents of the directory.

```
csi@csi-analyst:/opt/sherlock/sherlock$ ls
__init__.py  notify.py    resources  sherlock.py  tests
__main__.py  __pycache__  result.py  sites.py
csi@csi-analyst:/opt/sherlock/sherlock$
```

The file sherlock.py is the executable for the program. Since Sherlock is built using Python3, we must launch the program using the **python3** command.

For Kali users, if you have any issue launching the program, make sure you have Python3 installed along with pip3.

Let us first look at help for Sherlock. At the prompt type, **python3 sherlock.py -h**

```
csi@csi-analyst:/opt/sherlock/sherlock$ python3 sherlock.py -h
usage: sherlock.py [-h] [--version] [--verbose] [--folderoutput FOLDEROUTPUT]
                   [--output OUTPUT] [--tor] [--unique-tor] [--csv]
                   [--site SITE_NAME] [--proxy PROXY_URL] [--json JSON_FILE]
                   [--timeout TIMEOUT] [--print-all] [--print-found]
                   [--no-color] [--browse] [--local]
                   USERNAMES [USERNAMES ...]

Sherlock: Find Usernames Across Social Networks (Version 0.12.9)
```

As you can see, there are lots of options here, including options for using Tor. While we will not be using Tor today, this feature can come in handy when you do not want anyone to know who is researching their social media presence.

Having a social presence on the Internet comes with its perils. As long as we have a pretty good idea of the individual's username used most often, we can look it across many different social media sites.

In this next example, we are trying to get a picture of an individual's life on the Internet. The individual has a propensity to using the username of deadcow.

If the individual is creating a social media account using the name deadcow, we should be able to find them using Sherlock.

At the prompt, type in the following command.

python3 sherlock.py deadcow --print-found

Press enter.

Our user, deadcow, has quite a life on the Internet.

```
csi@csi-analyst:/opt/sherlock/sherlock$ python3 sherlock.py deadcow --print-found
[*] Checking username deadcow on:
[+] 9GAG: https://www.9gag.com/u/deadcow
[+] AllTrails: https://www.alltrails.com/members/deadcow
[+] Archive.org: https://archive.org/details/@deadcow
[+] AskFM: https://ask.fm/deadcow
[+] Audiojungle: https://audiojungle.net/user/deadcow
[+] Badoo: https://badoo.com/profile/deadcow
[+] Bandcamp: https://www.bandcamp.com/deadcow
[+] BitBucket: https://bitbucket.org/deadcow/
[+] Blogger: https://deadcow.blogspot.com
[+] BodyBuilding: https://bodyspace.bodybuilding.com/deadcow
[+] Bookcrossing: https://www.bookcrossing.com/mybookshelf/deadcow/
[+] BuzzFeed: https://buzzfeed.com/deadcow
[+] CNET: https://www.cnet.com/profiles/deadcow/
[+] Career.habr: https://career.habr.com/deadcow
[+] Chess: https://www.chess.com/member/deadcow
[+] Clozemaster: https://www.clozemaster.com/players/deadcow
[+] Codecademy: https://www.codecademy.com/profiles/deadcow
[+] DailyMotion: https://www.dailymotion.com/deadcow
[+] DeviantART: https://deadcow.deviantart.com
[+] Discogs: https://www.discogs.com/user/deadcow
[+] Disqus: https://disqus.com/deadcow
[+] Docker Hub: https://hub.docker.com/u/deadcow/
[+] Duolingo: https://www.duolingo.com/profile/deadcow
[+] Ello: https://ello.co/deadcow
```

At the prompt, type **`ls`**. In our previous command, we used the **`-- print find`** to the command syntax. This saves the results as a text file to the sherlock directory.

```
csi@csi-analyst:/opt/sherlock/sherlock$ ls
deadcow.txt  __init__.py  __main__.py  notify.py  __pycache__  resources
csi@csi-analyst:/opt/sherlock/sherlock$
```

To see the list of user accounts found using the name deadcow, we can print out the text file's content to the terminal using the **`cat`** command.

At the prompt, type **`cat deadcow.txt`**.

We see the username deadcow is detected or found on 122 websites.

```
https://www.toster.ru/user/deadcow/answers
Total Websites Username Detected On : 122
csi@csi-analyst:/opt/sherlock/sherlock$
```

Not all username searches are going to be this easy. You can search for a username of an individual using their name as it appears in their email address, their first initial_last name, and just about any other combination, you can think of.

When searching for an individual using two words joined, try separating the two parts using a dash or reversing the words. For this search, we could use dead-cow, cow-dead, or any combination thereof.

As you go through the list, you can rule out plenty of sites since the sites are no longer active such as Google+. Many sites may result from someone else using the same username or setting up a profile using the name to either take credit for the individual's work or other reasons.

The text file has clickable links, so as you go through the investigation, you can visit the different sites looking for images and other data that might be of use.

Summary

Sometimes the most mundane piece of information can be the clue you are looking for. One of the most prolific child molesters in the history of the U.K., Richard Huckle, was found by searching the Internet for social media accounts using his preferred username, K. Once they located the accounts, they were able to pinpoint the exact individual using his preferred greeting for every post, Hiya! Huckle had a social media account registered with a social media site for vehicle enthusiasts where users could post their vehicles' images. He posted a frontal image of his car showing his license plate. Game over!

His arrest came after Australian investigators tracking the activities of another pedophile who ran a child abuse site on the dark web noticed Huckle (K) was a prolific user of the website. He was arrested at Gatwick airport by National Crime Agency officials when he returned home for Christmas in 2014.