Master OTW's Hacker Training Camp

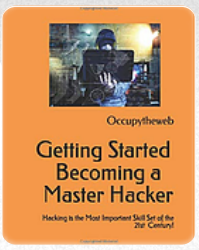
Log In

Fools talk;
The Wise listen.

Search Site

It's Finally Here!

Cyber Ninja @CyberSecTalk · 14h
Replying to @three_cube
Could not stop reading! The author is not only an elite cybersec expert, he is a natural born teacher too! I have never seen better approach in a hacking book. If you are serious about learning hacking skills, it would be very wise of you to get this book. Gem and future classic.



Click Here to Get Yours!

Return to Blog

OTW · May 28 · 3 min read

OSINT, Part 3: Extracting Employee Names from Companies (Tesla and Breitbart) on LinkedIn

Welcome back, my aspiring cyber warriors!

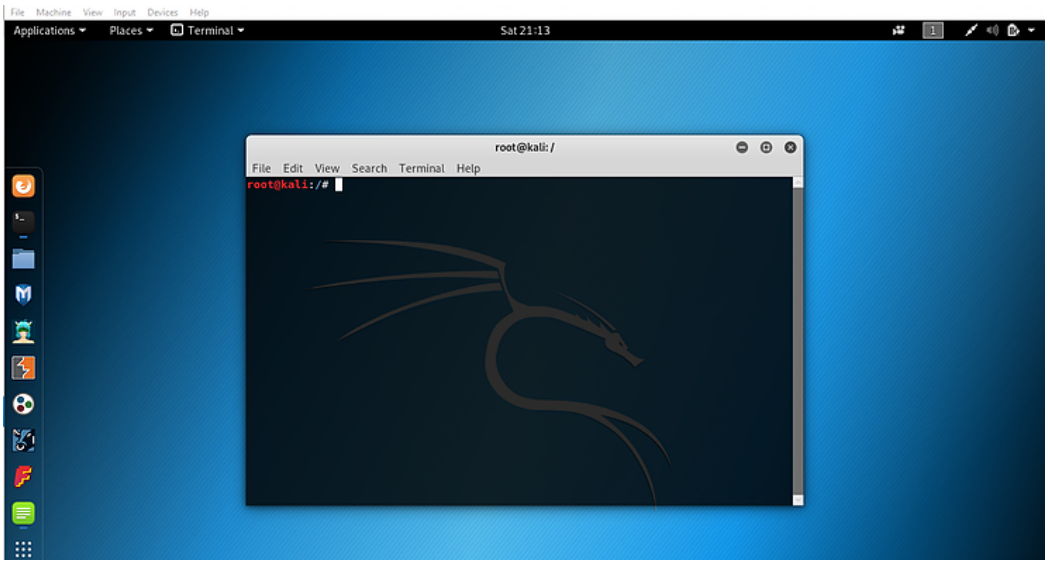
The Internet is the deepest and widest data repository in the history of the world! Those who can extract and cultivate intelligence from it, will be empowered like none other!

This data can be used for offensive security and forensic investigations, among many other applications.

Crosslinked is one more tool for automating the gathering of this data from the huge repository. Crosslinked is Python script for extracting company employee names from LinkedIn. Of course, we could do this manually, but this tool will save us many tens of hours of tedious work.

Step #1: Fire Up Kali

The first step, of course, is to fire up our trusty Kali and open a terminal.



Step #2: Download and install [crosslinked.py]

Crosslinked is not built into Kali, nor is it in our Kali repository but we can find it on github.com. Simply clone it from **m8r0wn**'s repository.

kali > git clone https://github.com/m8r0wn/crosslinked

```
root@kali:~# git clone https://github.com/m8r0wn/crosslinked
Cloning into 'crosslinked'...
remote: Enumerating objects: 34, done.
remote: Counting objects: 100% (34/34), done.
remote: Compressing objects: 100% (30/30), done.
remote: Total 34 (delta 15), reused 11 (delta 3), pack-reused 0
Unpacking objects: 100% (34/34), done.
root@kali:~# cd crosslinked
```

Next, we need to download and install crosslinked's requirements. There should be a file named **requirements.txt** in our new crosslinked directory.

kali > cd crosslinked

kali > pip3 install -r requirements.txt

```
root@kali:~/crosslinked# pip3 install -r requirements.txt
Collecting bs4 (from -r requirements.txt (line 1))
  Downloading https://files.pythonhosted.org/packages/10/ed/7e8b97591f6f456174139ec089c769f89a94a1a4025fe967691de971f314/bs4-0.0.1.tar.gz
Requirement already satisfied: beautifulsoup4 in /usr/lib/python3/dist-packages (from bs4->-r requirements.txt (line 1)) (4.6.3)
Building wheels for collected packages: bs4
  Running setup.py bdist_wheel for bs4 ... done
  Stored in directory: /root/.cache/pip/wheels/a0/b0/b2/4f80b9456b87abedbc0bf2d52235414c3467d8889be38dd472
Successfully built bs4
Installing collected packages: bs4
Successfully installed bs4-0.0.1
root@kali:~/crosslinked#
```

Step #3: Crosslinked Help

Before we begin working with crosslinked, let's look at its cursory help file.

kali > crosslinked -h

```
root@kali:~/crosslinked# ./crosslinked.py -h
positional arguments:
  company_name  Target company name

optional arguments:
  -h, --help    show this help message and exit
  -t TIMEOUT    Timeout [seconds] for search threads (Default: 25)
  -j JITTER     Jitter for scraping evasion (Default: 0)
  -o OUTFILE    Change name of output file (default: names.txt
  -f NFORMAT    Format names, ex: 'domain\{f}{last}', '{first}.{last}@domain.com'
  -s, --safe    Only parse names with company in title (Reduces false positives)
  -v            Show names and titles recovered after enumeration
```

In it's simplest form, the crosslinked syntax looks like this;

crosslinked.py <name format> <company>

It's also important to note that you must give yourself permission to run the script.

kali > chmod 755 crosslinked.py

Step #4: Extracting Tesla Employees from LinkedIn

Now that we have everything setup with crosslinked script, let's see whether we can find employees of Tesla, Elon Musk's electric car company. To do so, we need to specify the name format and the company name.

kali > ./crosslinked.py -f '{first}.{last}'@tesla.com' tesla

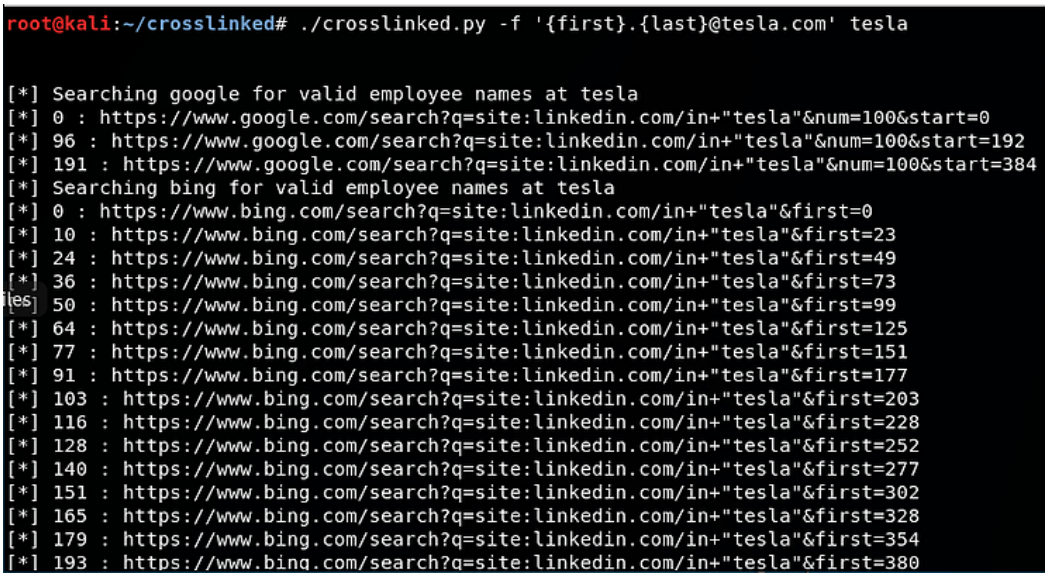
Where:

crosslinked.py is the command

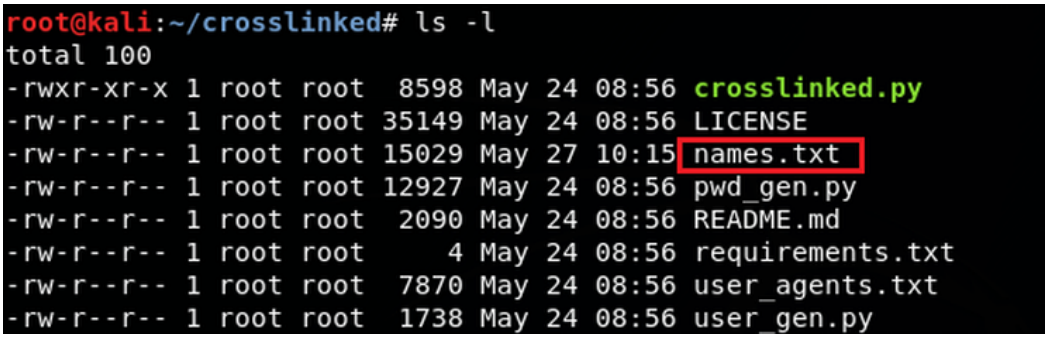
-f format option of the names

'{first}.{last}@company.com' the name format to use

tesla the company we are searching



When the script has completed its run, crosslinked should place a file in the default directory named **names.txt.** We can find it by simply doing a long listing.



To see the contents of this file, simply use the command **more** before the file name. As we can see above, crosslinked was able to extract the names of hundreds of people who work at Elon Musk's Tesla.

kali > more names.txt

```
root@kali:~/crosslinked# more names.txt
jerome.guillen@tesla.com
peter.hochholdinger@tesla.com
kevin.kassekert@tesla.com
kim.wong@tesla.com
brian.dow@tesla.com
drew.baglino@tesla.com
sascha.zahnd@tesla.com
brendan.meyer@tesla.com
david.zhang@tesla.com
morrie.eisenberg@tesla.com
sri.srinivasa@tesla.com
casey.schendel@tesla.com
zach.kirkhorn@tesla.com
daniel.king@tesla.com
kristin.finkelstein@tesla.com
nagesh.saldi@tesla.com
drew.bennett@tesla.com
alex.liebl@tesla.com
nicholas.brown@tesla.com
jorge.milburn@tesla.com
anders.bell@tesla.com
peter.blades@tesla.com
nicholas.lambert@tesla.com
dr.marcus@tesla.com
robin.ren@tesla.com
keely.sulprizio@tesla.com
jonathan.chang@tesla.com
jatinder.dhillon@tesla.com
charlie.haley@tesla.com
andrej.karpathy@tesla.com
jb.straubel@tesla.com
suraj.nagaraj@tesla.com
jenna.halperin@tesla.com
charlotte.corley@tesla.com
shae.otsuka@tesla.com
dr.michael@tesla.com
alex.wolff@tesla.com
felicia.mayo@tesla.com
--More--(6%)
```

Step #6: Extract the People Working at Breitbart News

Let's see if we can do the same task against another company. Let's find the employees of Breitbart News, the hate-mongering, conspiracy promoting, racist and mysogynist online magazine.

We already have the Tesla employees in the names.txt file, so unless we want to append the Breitbart employees to that file, we will need to direct crosslinked to create a new file. We can do that using the **-o** switch (see the help screen above).

kali > ./crosslinked.py -f '{first}.{last}@breitbart' breitbart -o breitbart.txt

```
root@kali:~/crosslinked# ./crosslinked.py -f '{first}.{last}@breitbart' breitbart -o breitbart.txt
[*] Searching google for valid employee names at breitbart
[*] 0 : https://www.google.com/search?q=site:linkedin.com/in+"breitbart"&num=100&start=0
[*] 92 : https://www.google.com/search?q=site:linkedin.com/in+"breitbart"&num=100&start=161
[*] 160 : https://www.google.com/search?q=site:linkedin.com/in+"breitbart"&num=100&start=304
[*] 161 : https://www.google.com/search?q=site:linkedin.com/in+"breitbart"&num=100&start=328
[*] Searching bing for valid employee names at breitbart
```

```
[*] 0 : https://www.bing.com/search?q=site:linkedin.com/in+"breitbart"&first=0
[*] 10 : https://www.bing.com/search?q=site:linkedin.com/in+"breitbart"&first=23
[*] 24 : https://www.bing.com/search?q=site:linkedin.com/in+"breitbart"&first=49
[*] 37 : https://www.bing.com/search?q=site:linkedin.com/in+"breitbart"&first=75
[*] 47 : https://www.bing.com/search?q=site:linkedin.com/in+"breitbart"&first=99
[*] 57 : https://www.bing.com/search?q=site:linkedin.com/in+"breitbart"&first=125
[*] 70 : https://www.bing.com/search?q=site:linkedin.com/in+"breitbart"&first=151
[*] 79 : https://www.bing.com/search?q=site:linkedin.com/in+"breitbart"&first=177
[*] 91 : https://www.bing.com/search?q=site:linkedin.com/in+"breitbart"&first=203
[*] 100 : https://www.bing.com/search?q=site:linkedin.com/in+"breitbart"&first=229
[*] 109 : https://www.bing.com/search?q=site:linkedin.com/in+"breitbart"&first=255
[*] 118 : https://www.bing.com/search?q=site:linkedin.com/in+"breitbart"&first=281
[*] 128 : https://www.bing.com/search?q=site:linkedin.com/in+"breitbart"&first=305
```

Now, crosslinked goes out and extracts the Breitbart employee names from LinkedIn. When we do a long listing on our default directory, we find the file **breitbart.txt** that we directed crosslinked to create in the command above.

kali > ls -l

```
root@kali:~/crosslinked# ls -l
total 116
-rw-r--r-- 1 root root 12172 May 27 10:58 breitbart.txt
-rwxr-xr-x 1 root root  8598 May 24 08:56 crosslinked.py
-rw-r--r-- 1 root root 35149 May 24 08:56 LICENSE
-rw-r--r-- 1 root root 19286 May 27 10:56 names.txt
-rw-r--r-- 1 root root 12927 May 24 08:56 pwd_gen.py
-rw-r--r-- 1 root root  2090 May 24 08:56 README.md
-rw-r--r-- 1 root root     4 May 24 08:56 requirements.txt
-rw-r--r-- 1 root root  7870 May 24 08:56 user_agents.txt
-rw-r--r-- 1 root root  1738 May 24 08:56 user_gen.py
```

We can see the contents of that file by prefacing the file name with "more".

kali > more breitbart.txt

```
root@kali:~/crosslinked# more breitbart.txt
alex.marlow@breitbart
ben.kew@breitbart
michelle.moons@breitbart
rebecca.mansour@breitbart
pamela.key@breitbart
larry.solov@breitbart
amanda.house@breitbart
alana.mastrangelo@breitbart
jeff.poor@breitbart
john.carney@breitbart
joel.pollak@breitbart
neil.munro@breitbart
alexandre.breitbart@breitbart
nate.church@breitbart
samuel.chi@breitbart
matthew.boyle@breitbart
donna.breitbart@breitbart
tziona.breitbart@breitbart
edwin.mora@breitbart
charlie.spiering@breitbart
allum.bokhari@breitbart
kyle.morris@breitbart
andrew.breitbart@breitbart
samuel.breitbart@breitbart
nele.breitbart@breitbart
alex.breitbart@breitbart
charles.breitbart@breitbart
```

As you can see, crosslinked was capable ofextracting hundreds of employees names from

LinkedIn that work at Breitbart News. These are the people you can thank for defiling the public discourse with hate-filled, racist, xenophobic, and mysogynist misinformation.

Summary

The Internet harbors a vast wealth of information just waiting to untethered. Crosslinked helps us automate the process of extracting employee names for particular companies from LinkedIn, which may be crucial in a digital forensic investigation or penetration testing environment.

1,267 views

Recent Posts

See All

Join the Cyber Warrior Team at Hackers-Arise! Become a MEMBER! J...

1,235 Write a comment 5

Database Hacking, Part 4: Extracting Data with sqlmap

16,467 Write a comment 2

SCADA Hacking: Anatomy of a SCADA Malware, BlackEnergy 3

1,562 Write a comment 2

Log in to leave a comment.