34

From the email on CORP-ADT01 we know that the corporate emails are moving to a new email format, we also know that we have a propriety breached credentials service that we utilize to find credentials on the network. Using LinkedIn along leetLinked we can pull names and emails from LinkedIn to format with namely and insert into Breach || GTFO.

OSINT with LinkedIn Overview
LinkedIn, since its creation, has represented an incredibly valuable vector for information gathering. After all, who would have imagined that people would just post their company structure, job overview, and work history publicly online if we just asked them kindly? When I personally perform recon (Dark speaking here), I typically visit LinkedIn throughout my second stage of OSINT - typically following an initial automated phrase wherein I gather any emails that are publicly available. Once that is done and we know the format our target uses for emails, we can journey onto LinkedIn and scrape names to expand our email list. We'll dive into this further in the next section with leetlinked.

Utilizing leetLinked for LinkedIn Recon
LeetLinked is a tool developed by Sq00ky and Horshark to automate discovery of company employees LinkedIn accounts. By discovering all the employees an organization has, this can be incredibly useful for password spraying, phishing, or other targeted attacks. Built in, it has a feature to generate emails based off a given format, which you can use to password spray, phish, or lookup their email in a databreach. The optional -p flag utilizes HaveIBeenPwned's API to query each email and enumerate which databreaches the victim has been apart of. The output is stored within a spreadsheet that can be converted into a table and easily sorted and organized.

Instaling leetLinked
Installation of leetLinked is as easy as one, two, three... four!
1.) cd /opt
2.) git clone https://github.com/Sq00ky/LeetLinked
3.) cd /opt/leetlinked
4.) pip3 install -r requirements.txt

```
┌─[root@pandorasbox]─[~]
└──╼ #git clone https://github.com/Sq00ky/LeetLinked.git /opt/leetlinked
Cloning into '/opt/leetlinked'...
remote: Enumerating objects: 79, done.
remote: Counting objects: 100% (79/79), done.
remote: Compressing objects: 100% (70/70), done.
remote: Total 79 (delta 34), reused 6 (delta 3), pack-reused 0
Unpacking objects: 100% (79/79), 26.03 KiB | 346.00 KiB/s, done.
┌─[root@pandorasbox]─[~]
└──╼ #cd /opt/leetlinked
┌─[root@pandorasbox]─[/opt/leetlinked]
└──╼ #pip3 install -r requirements.txt
Requirement already satisfied: bs4 in /usr/local/lib/python3.8/dist-packages (from -r requirements.txt (line 1)) (0.0.1)
Requirement already satisfied: unidecode in /usr/local/lib/python3.8/dist-packages (from -r requirements.txt (line 2)) (1.1.1)
Requirement already satisfied: requests in /usr/local/lib/python3.8/dist-packages (from -r requirements.txt (line 3)) (2.23.0)
Requirement already satisfied: argparse in /usr/local/lib/python3.8/dist-packages (from -r requirements.txt (line 4)) (1.4.0)
Requirement already satisfied: xlwt in /usr/local/lib/python3.8/dist-packages (from -r requirements.txt (line 5)) (1.3.0)
Requirement already satisfied: xlrd in /usr/local/lib/python3.8/dist-packages (from -r requirements.txt (line 6)) (1.2.0)
Requirement already satisfied: beautifulsoup4 in /usr/local/lib/python3.8/dist-packages (from bs4->-r requirements.txt (line 1)) (4.9.1)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/lib/python3.8/dist-packages (from requests->-r requirements.txt (line 3)) (1.25.9)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.8/dist-packages (from requests->-r requirements.txt (line 3)) (2020.4.5.2)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.8/dist-packages (from requests->-r requirements.txt (line 3)) (2.9)
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.8/dist-packages (from requests->-r requirements.txt (line 3)) (3.0.4)
Requirement already satisfied: soupsieve>1.2 in /usr/local/lib/python3.8/dist-packages (from beautifulsoup4->bs4->-r requirements.txt (line 1)) (2.0.1)
┌─[root@pandorasbox]─[/opt/leetlinked]
└──╼ #python3 leetlinked.py
leetlinked.py: error: the following arguments are required: -e/--email-domain, -f/--email-format, company_name
┌─[✗]─[root@pandorasbox]─[/opt/leetlinked]
└──╼ #
```

leetLinked's installation

Pulling info from LinkedIn with leetLinked

LeetLinked is a LinkedIn Recon tool used to gather employees at a company by utilizing search engines like Google and Bing. The results are returned in an Excel Spreadsheet for ease of use. If you have a Have I Been Pwned API key it can also tell you what data breaches the user has been involved in, with a separate section for specific password breaches. In combination with Dehashed, or other password breach lookup services, it can be an effective red-team ops tool.

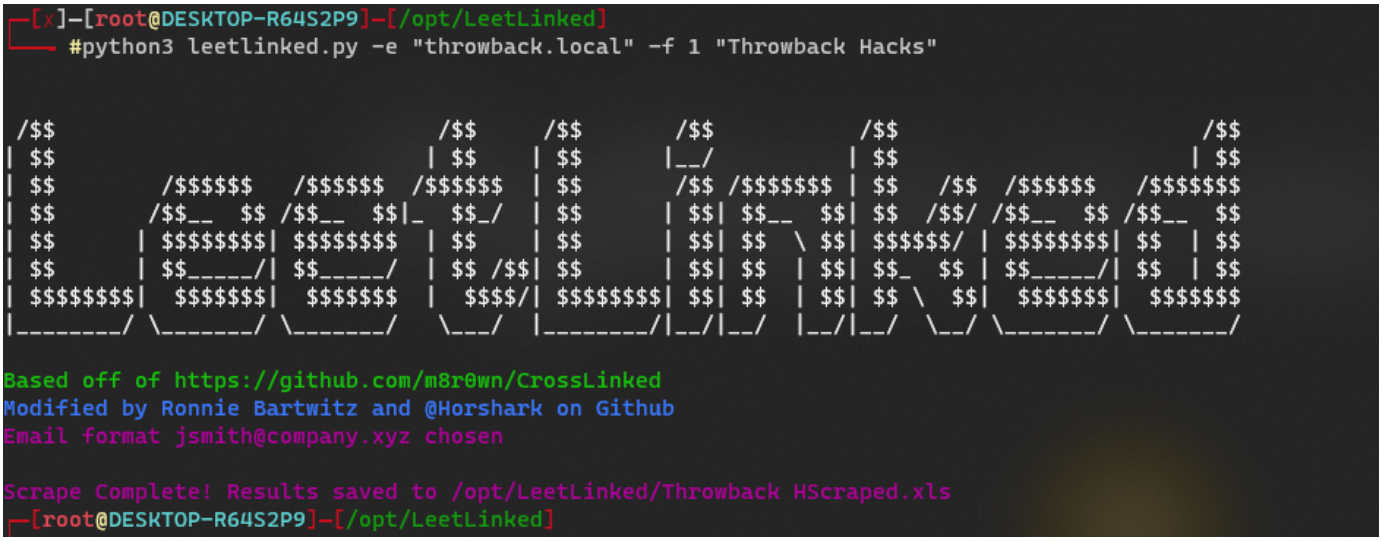
Using the tool is easy, there are several required arguments, -e for the email domain. For example, gmail.com, -f, which serves as the email format. There is a list of built-in email formats that are included in the tool that you can choose from, and adding your own is easy enough if you understand basic python. Lastly, a positional argument of the Company Name you're trying to search for is required.

```
┌─[root@DESKTOP-R64S2P9]─[/opt/LeetLinked]
└──╼ #python3 leetlinked.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)
  -s, --safe            Only parse names with company in title (Reduces false positives)
  -e EMAIL_DOMAIN, --email-domain EMAIL_DOMAIN
                        Include the email domain for email-generation (Example: microsoft.com)
  -p HIBP, --hibp HIBP  Runs all of the emails through HaveIBeenPwned's API and will list pwned accounts, API key is a required argument.

  -f EMAIL_FORMAT, --email-format EMAIL_FORMAT
                        Generates emails based on various formats, 1=jsmith 2=johnsmith 3=johns 4=smithj 5=john.smith 6=smith.john 7=smit
h, 8=john 9=john_smith 10=smith_john 11=js
┌─[root@DESKTOP-R64S2P9]─[/opt/LeetLinked]
└──╼ #
```

leetLinked's help menu

Combining all the arguments together and you will end up with a spreadsheet of all the First/Last names it finds from LinkedIn, plus emails (with the supplied email format you give it). This is incredibly useful for automating Human Recon for Phishing campaigns.

1.) python3 leetlinked.py -e "throwback.local" -f 1 "Throwback Hacks"



leetLinked pulling names from LinkedIn

Now that we a list of employees' first and last names we can put them into namely along with the email format from the admin desktop to create a wordlist and insert into Breach || GTFO.

Answer the questions below

Read the above and use leetLinked to gather info from LinkedIn.

Completed