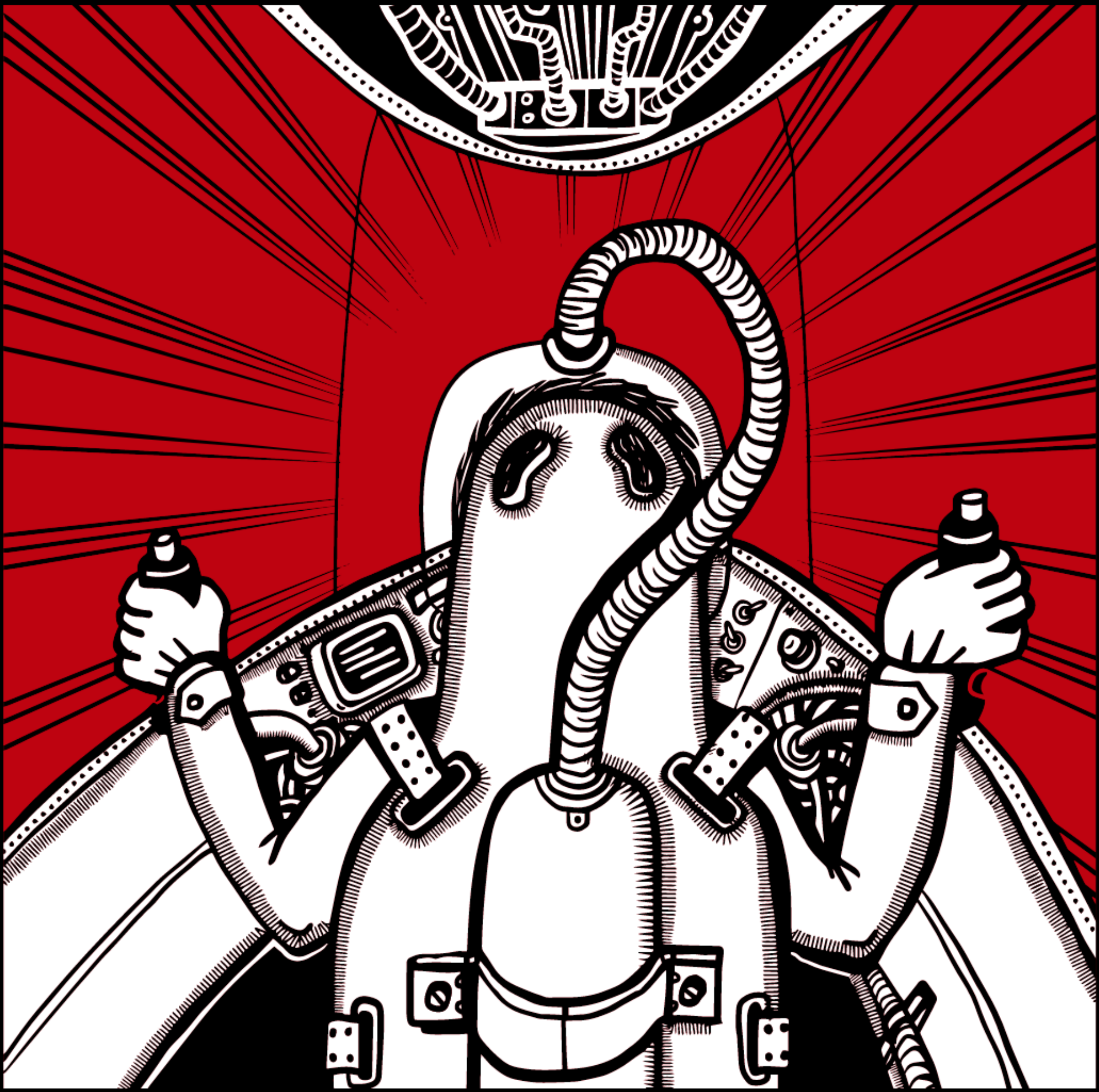
Open Source Intelligence How-To
for Talent Sourcers, Recruiters, Security Agencies and Cyberstalkers

BUILDING AN OSINT SUPER MACHINE

FOR PEOPLE AND ORGANIZATION SURVEILLANCE



JOSÉ KADLEC
FROM PRAGUE

VOL 1.0

Building an OSINT Super Machine
for People and Organization Surveillance

José Kadlec

AUTHOR

José Kadlec is a former ethical hacker, digital forensic examiner and hardcore Linux engineer who went head over into the talent sourcing industry utilizing his cross-field experience.

Based on the OSINT techniques co-founded a holding of companies *Datacruit*, *GoodCall* and *Recruitment Academy*. They made made it to the FT1000 as the 415th fastest growing company in Europe by Financial Times.

Follow José on: LinkedIn | Twitter | Facebook | Instagram

www.JoseKadlec.com

🔎 Introduction to OSINT (Open Source Intelligence)

OSINT (Open-Source Intelligence) methods has been already commonly adopted by the talent sourcing community. And it's logical. OSINT is a methodology for collecting and analyzing data from publicly available sources. So, it is safe to say that every LinkedIn search query is OSINT as well and therefore every sourcer or a recruiter uses that on everyday basis.

The twist is that with OSINT you can go way deeper and challenge what we usually understand as the publicly available data.

There are several reasons why you can utilize the OSINT methods in the process of recruitment.

To name a few we should definitely include:

- Searching for candidates (longlisting)
- Data enrichment on candidates (screening, engagement)
- Searching for contact information (approaching)
- Mapping the market (talent mapping)
- Discovering information about specific companies (market mapping)

We can break down the publicly available data into several layers:

- Native social media data (that's what talent sourcers use the most)
- Other digital data placed off the social media networks (images, maps, resumes, etc.)
- Deep data (e.g. data from the privates social media profiles or from disallowed directories in robots.txt)
- Leaked data (e.g. Apollo leak - leaked or scraped databases of various services such as LinkedIn, Clubhouse, etc.)
- Dark web data (data searchable on .onion domains only)
- Cyber data beyond Internet (e.g. GSM networks)

Sometimes it is not about if you can capture the data only but how efficiently and if you can postprocess those data easily so we also go vertically from:

- the usage without any 3rd party tools (e.g. LinkedIn and Twitter search)
- common search engines (Google, Yandex, etc.)
- specialized search engines (e.g. Shodan, Censys, Carrot2)
- web applications and plugins (username search engines, e-mail verification services)
- to specialized OSINT bundles (Maltego)
- up to a dedicated Linux-based OSINT distributions (OSINTTUX, Kali Linux, Buscador).

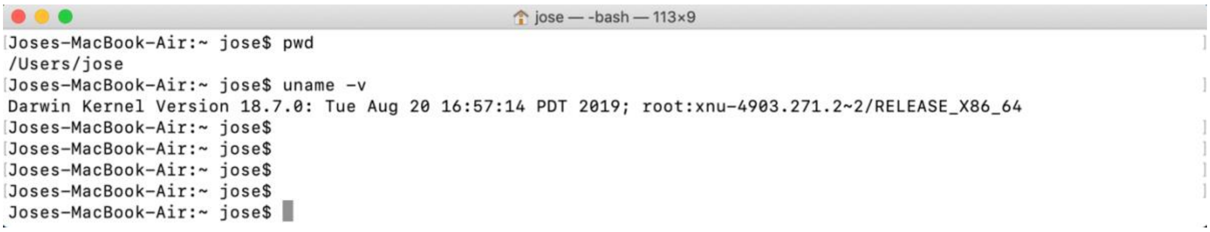
In this ebook, I'm going to focus on the last one and that's how to practice OSINT using Linux-based systems – more specifically over so called CLI (Command Line Interface) applications.

The advantage of using a command-line applications is usually better performance and ability to post-process results better than with the web applications or browser extensions. If the web app doesn’t have an API and webhooks (e.g. for Zapier or Integromat), you cannot do much with the output. That’s why the OSINT web apps we are talking about are the best for one off usage. The command-line apps might also offer more options as they are easier to implement for the developer who doesn’t need to develop a GUI.

⚠ DISCLAIMER: In no event shall the author of this ebook be liable for any special, consequential, incidental or indirect damages whatsoever (including, without limitation, damages for loss of business profits, business interruption, loss of business information, or any other pecuniary loss or damage) arising out of the use of this product.

Where can I use some Linux?

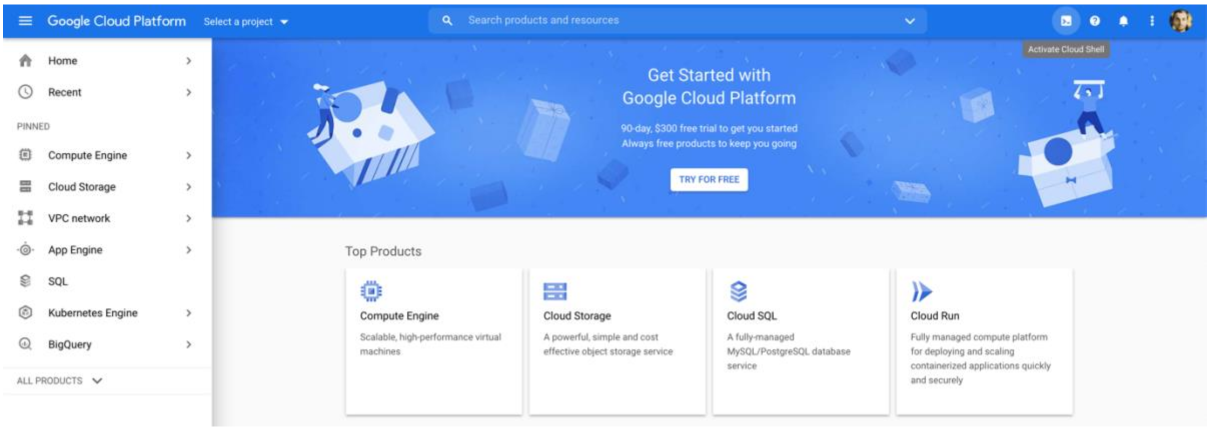
Linux is a Unix-based system and there are many variations and forks developed over the past. Actually, if you are using MacOS, you are also using Unix-based system (not Linux thought) and you can access the terminal (open *Terminal* app) and use Unix commands.



```
Joses-MacBook-Air:~ jose$ pwd
/Users/jose
Joses-MacBook-Air:~ jose$ uname -v
Darwin Kernel Version 18.7.0: Tue Aug 20 16:57:14 PDT 2019; root:xnu-4903.271.2~2/RELEASE_X86_64
Joses-MacBook-Air:~ jose$
Joses-MacBook-Air:~ jose$
Joses-MacBook-Air:~ jose$
Joses-MacBook-Air:~ jose$
Joses-MacBook-Air:~ jose$ █
```

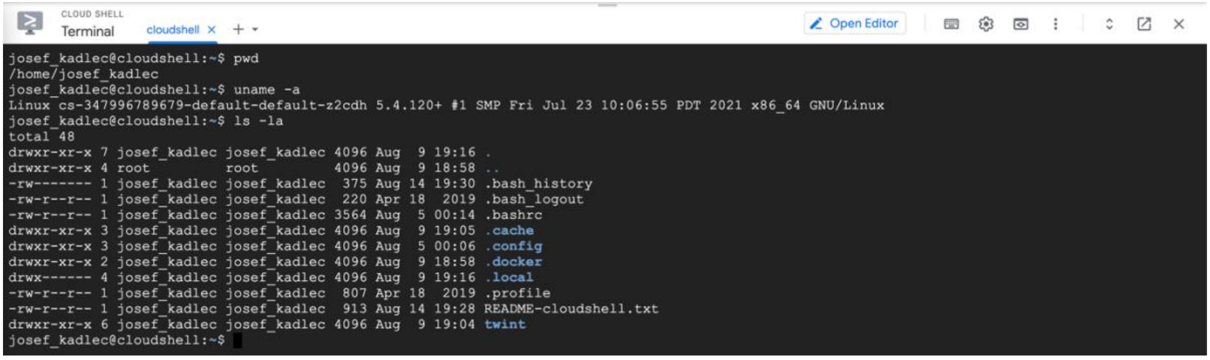
Google Cloud Console

Another way how to use a pure Linux system is *Google Cloud Console*. Open [*Google Cloud Console*]() and click on *Activate Cloud Shell*.



Google Cloud Platform
Select a project
Search products and resources
Activate Cloud Shell
Home
Recent
PINNED
Compute Engine
Cloud Storage
VPC network
App Engine
SQL
Kubernetes Engine
BigQuery
ALL PRODUCTS
Get Started with
Google Cloud Platform
90-day, $300 free trial to get you started
Always free products to keep you going
TRY FOR FREE
Top Products
Compute Engine
Scalable, high-performance virtual machines
Cloud Storage
A powerful, simple and cost effective object storage service
Cloud SQL
A fully-managed MySQL/PostgreSQL database service
Cloud Run
Fully managed compute platform for deploying and scaling containerized applications quickly and securely

This is going to open a terminal window with your private Linux-based system provided by Google in their cloud.



```
josef_kadlec@cloudshell:~$ pwd
/home/josef_kadlec
josef_kadlec@cloudshell:~$ uname -a
Linux cs-347996789679-default-default-z2cdh 5.4.120+ #1 SMP Fri Jul 23 10:06:55 PDT 2021 x86_64 GNU/Linux
josef_kadlec@cloudshell:~$ ls -la
total 48
drwxr-xr-x 7 josef_kadlec josef_kadlec 4096 Aug  9 19:16 .
drwxr-xr-x 4 root         root         4096 Aug  9 18:58 ..
-rw------- 1 josef_kadlec josef_kadlec  375 Aug 14 19:30 .bash_history
-rw-r--r-- 1 josef_kadlec josef_kadlec  220 Apr 18  2019 .bash_logout
-rw-r--r-- 1 josef_kadlec josef_kadlec 3564 Aug  5 00:14 .bashrc
drwxr-xr-x 3 josef_kadlec josef_kadlec 4096 Aug  9 19:05 .cache
drwxr-xr-x 3 josef_kadlec josef_kadlec 4096 Aug  5 00:06 .config
drwxr-xr-x 2 josef_kadlec josef_kadlec 4096 Aug  9 18:58 .docker
drwx------ 4 josef_kadlec josef_kadlec 4096 Aug  9 19:16 .local
-rw-r--r-- 1 josef_kadlec josef_kadlec  807 Apr 18  2019 .profile
-rw-r--r-- 1 josef_kadlec josef_kadlec  913 Aug 14 19:28 README-cloudshell.txt
drwxr-xr-x 6 josef_kadlec josef_kadlec 4096 Aug  9 19:04 twint
josef_kadlec@cloudshell:~$ █
```

This is your text-based Linux where you can install and configure applications as you need. Pretty handy!

Note: You can access command line prompt on MS Windows as well but that's very limited functionality and you cannot compare it with the possibilities of the Unix-based terminal.

If you don't want to install your own Linux system and simply use Google Cloud Console, you can skip directly to the part *Let's Do Some OSINT!*.

Kali Linux in the Virtual Machine

Another way to use Linux system is to install your own. There are many Linux distributions you can choose from such as *Ubuntu, Debian, SUSE Linux, Fedora, Mandriva* or *Red Hat Enterprise Linux (RHEL)* which is more for the server-side usage. There are also so-called Live Linux distributions you can simply boot from a CD/DVD or an USB stick and they just run in your RAM. No installation needed. Kali supports that as well.

When I started with Linux about 20 years ago, I was a big fan of Slackware Linux and tweaking that for various purposes – for example creating a fully encrypted system which encrypt/decrypt on the fly once you boot or shutdown the laptop.

There is a plethora of distros for various usage and one of the then we are going to showcase for OSINT is *Kali Linux* focused on IT security.

Note: You might heard of other OSINT distributions such as OSINTTUX, Buscador (not supported anymore), Huron, Dora OSINT VM, CSI Linux, Tsuguri Linux or Trace Labs OSINT VM. For our needs you can use any of those even if Kali Linux is probably the most popular.

You have several options where to install Kali Linux:

- A dedicated empty computer (including ARMs) or server which you can reinstall
- *Raspberry PI* computer if you have a reason for that
- You can also install it and run it on MS Windows using WSL
- You can run it as a live boot USB drive without installation
- You can run it in a container such as Docker or LXD
- Real or virtualized server provided by a server house hosting
- Your own virtualized machine – that's what we are going to do.

Computer in a computer

I’m going to show the usage over the virtualized machine on our currently used computer as this is really convenient way for the purpose of OSINT.

There are two types of virtualizations – Type 1 and Type 2. Type 1 is a low-level virtualization where the hypervisor running the virtualized system(s) is sitting directly on the computer hardware – there is no OS in between. The bare-metal hypervisor is actually a small OS itself.

Physical Server -> Hypervisor type 1 -> Virtual Machine(s)

Linux system itself can be also used as hypervisor type 1 using KVM to make it even more complicated.

Type 2 hypervisor which we are going to use is running on the operations system we use on our computer (MacOS, MS Windows, etc.) and the virtualized system is created on top of that.

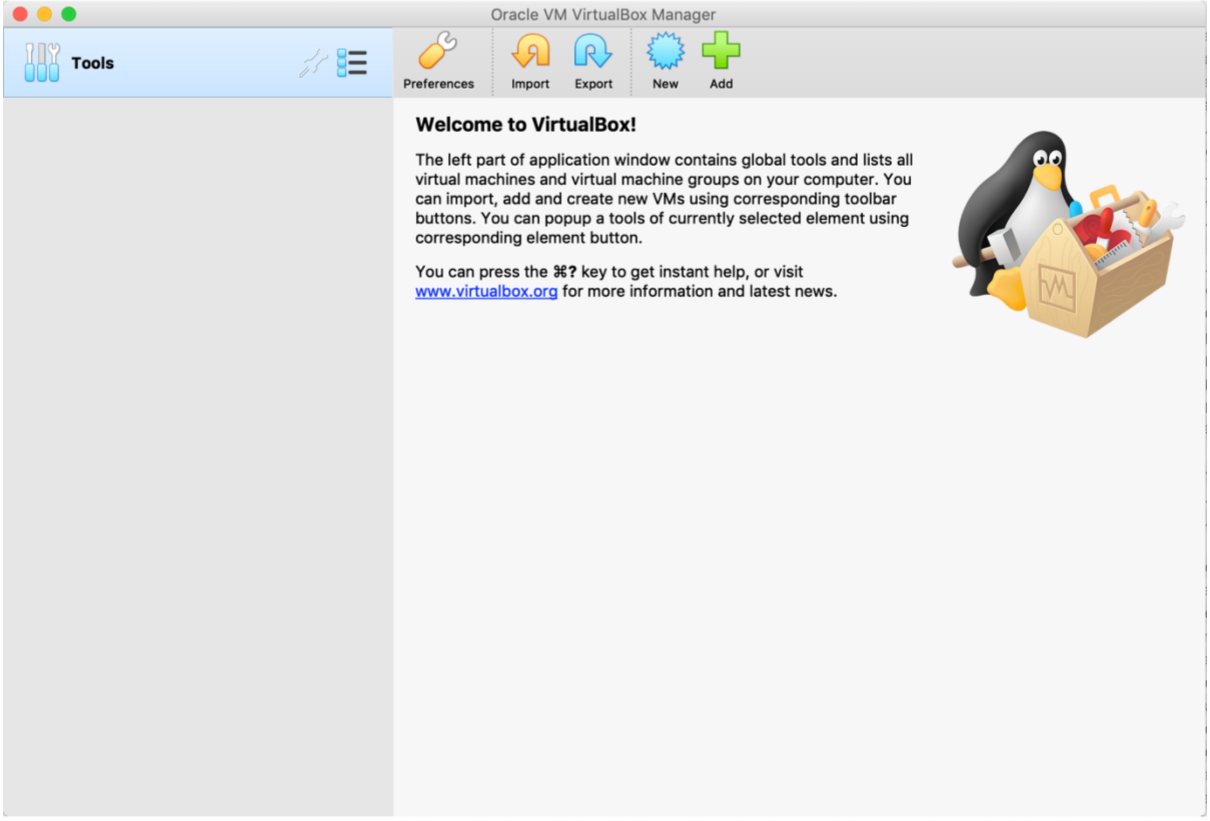
Physical Server -> Operations System -> Hypervisor type 2 -> Virtual Machine(s)

The hypervisors are being made by well-known companies such as VMware, Microsoft, Oracle and Parallels. I used to use VMware but currently got used to *Oracle VM VirtualBox*.

So, my setup will be:

Macbook Air -> MacOS -> Oracle VM VirtualBox -> Kali Linux

Regardless of if you use MacOS, Windows or Linux on your desktop, you can download and install VirtualBox on your computer.



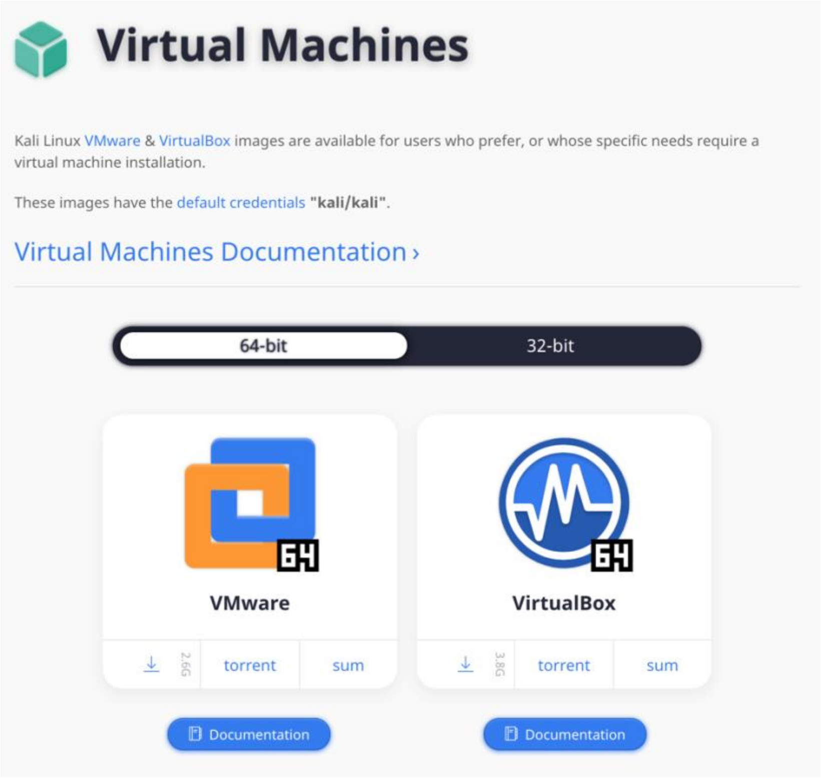
Welcome to VirtualBox!

The left part of application window contains global tools and lists all virtual machines and virtual machine groups on your computer. You can import, add and create new VMs using corresponding toolbar buttons. You can popup a tools of currently selected element using corresponding element button.

You can press the ⌘? key to get instant help, or visit www.virtualbox.org for more information and latest news.

You can see that the interface is pretty simple.

Now let's download Kali Linux which we are going to install in the virtual machine we haven't created yet. There are already ISO images prepared for a 64-bit VirtualBox so let's download that.



Virtual Machines

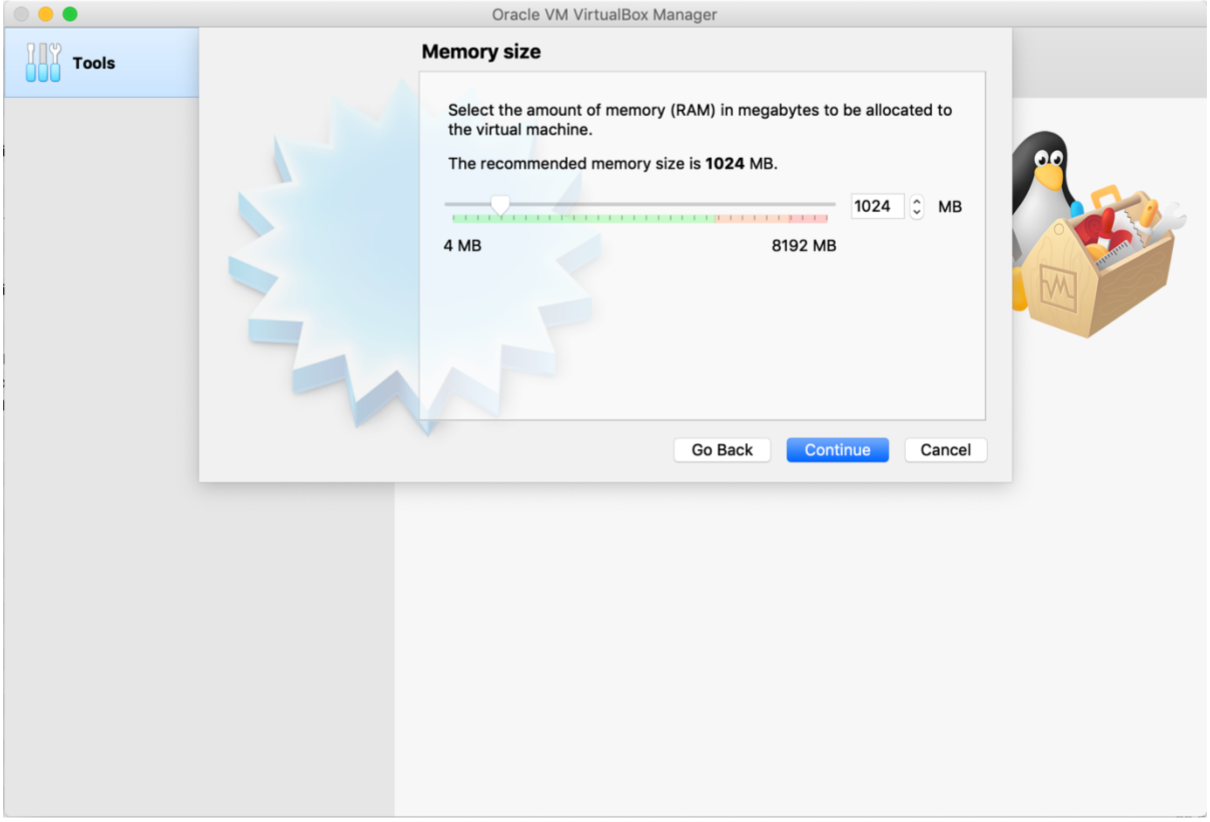
Kali Linux VMware & VirtualBox images are available for users who prefer, or whose specific needs require a virtual machine installation.

These images have the default credentials "kali/kali".

Virtual Machines Documentation ›

Now let's create our first virtual computer on this computer. Hit *New* icon in VirtualBox. Create a name of your virtual machine, insert which directory to put it at (I use an external hard drive) and choose that is going to be Debian Linux 64 bit as Kali Linux is based on traditional Debian Linux.

You can choose how much of your virtual memory (RAM) and physical memory (your hard disk) should be allocated for this virtual machine. Go with default or adjust based on free space and RAM capabilities.



Oracle VM VirtualBox Manager

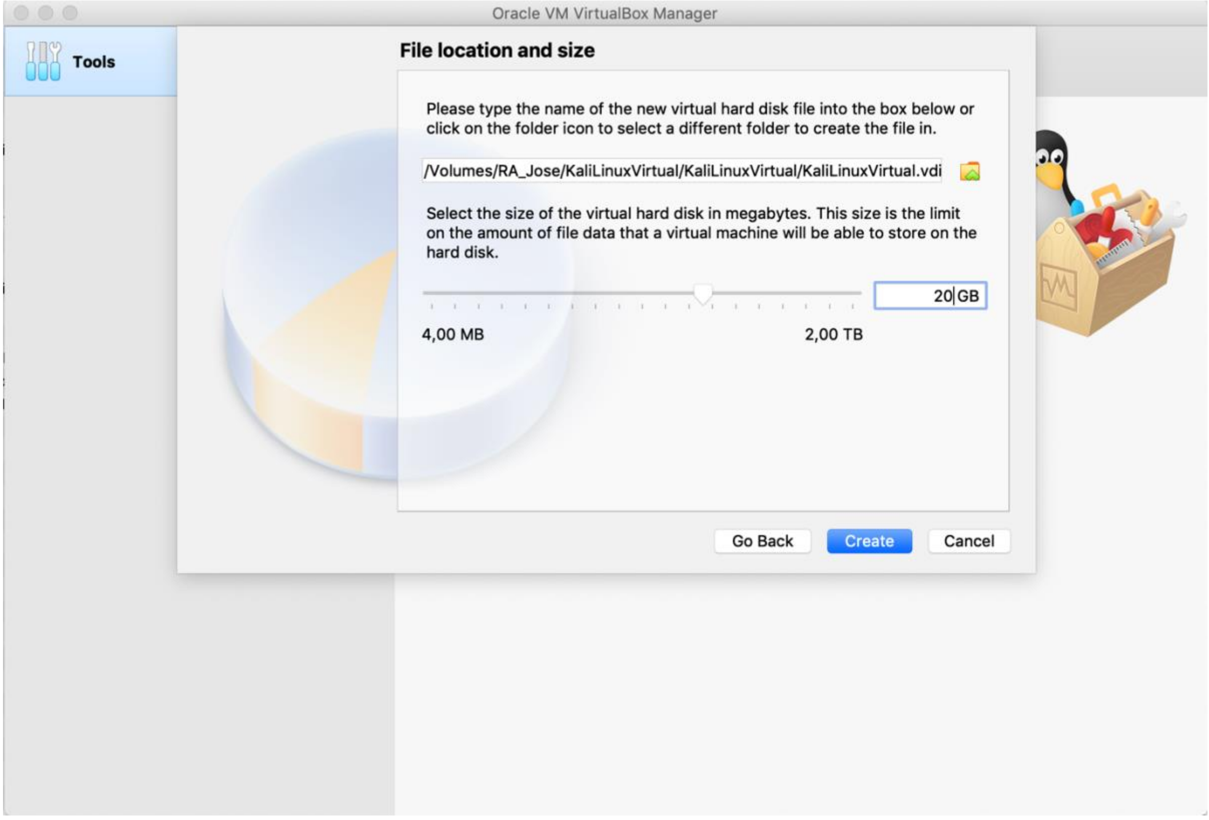
Tools

Memory size

Select the amount of memory (RAM) in megabytes to be allocated to the virtual machine.

The recommended memory size is **1024** MB.

1024 MB



Oracle VM VirtualBox Manager

Tools

File location and size

Please type the name of the new virtual hard disk file into the box below or click on the folder icon to select a different folder to create the file in.

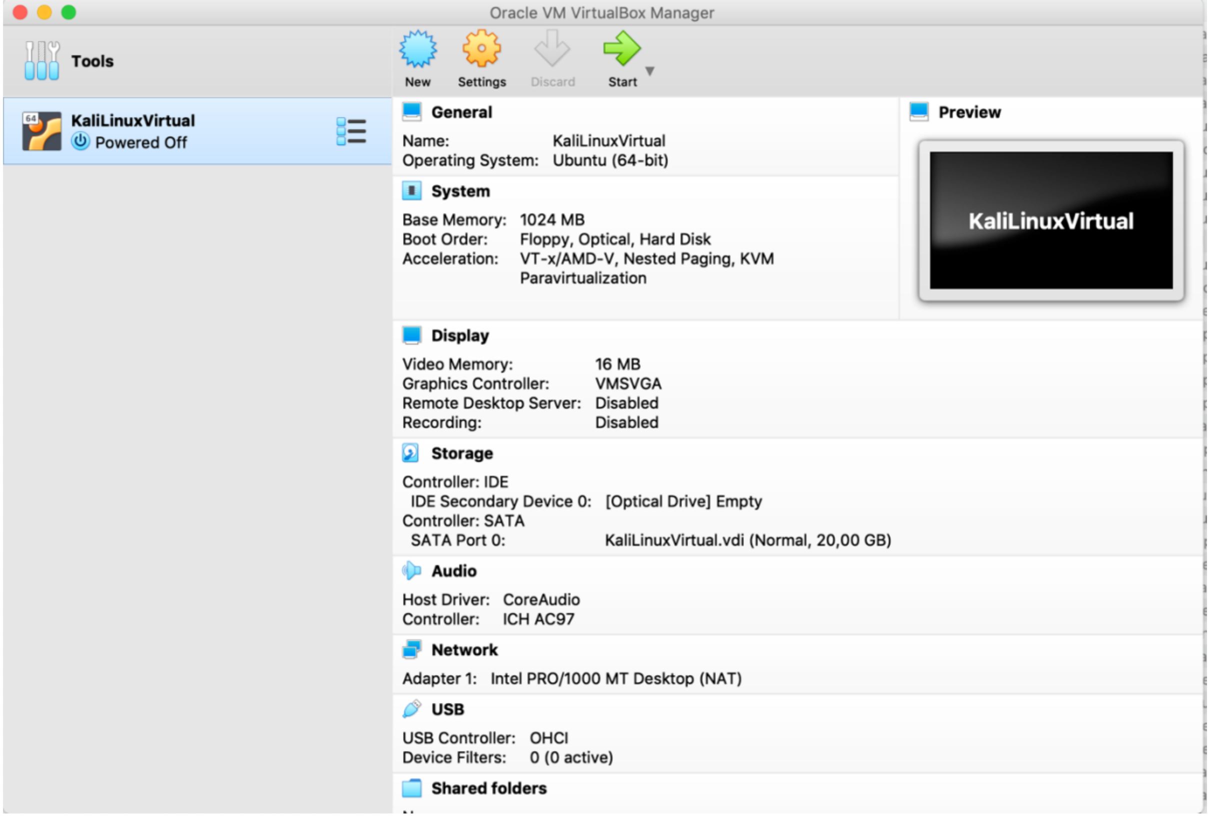
/Volumes/RA_Jose/KaliLinuxVirtual/KaliLinuxVirtual/KaliLinuxVirtual.vdi

Select the size of the virtual hard disk in megabytes. This size is the limit on the amount of file data that a virtual machine will be able to store on the hard disk.

20 GB

I put 1 GB of RAM and 20 GB of disk space.

And you are done. Now you created a computer inside your computer.



General

Name: KaliLinuxVirtual
Operating System: Ubuntu (64-bit)

System

Base Memory: 1024 MB
Boot Order: Floppy, Optical, Hard Disk
Acceleration: VT-x/AMD-V, Nested Paging, KVM Paravirtualization

Display

Video Memory: 16 MB
Graphics Controller: VMSVGA
Remote Desktop Server: Disabled
Recording: Disabled

Storage

Controller: IDE
IDE Secondary Device 0: [Optical Drive] Empty
Controller: SATA
SATA Port 0: KaliLinuxVirtual.vdi (Normal, 20,00 GB)

Audio

Host Driver: CoreAudio
Controller: ICH AC97

Network

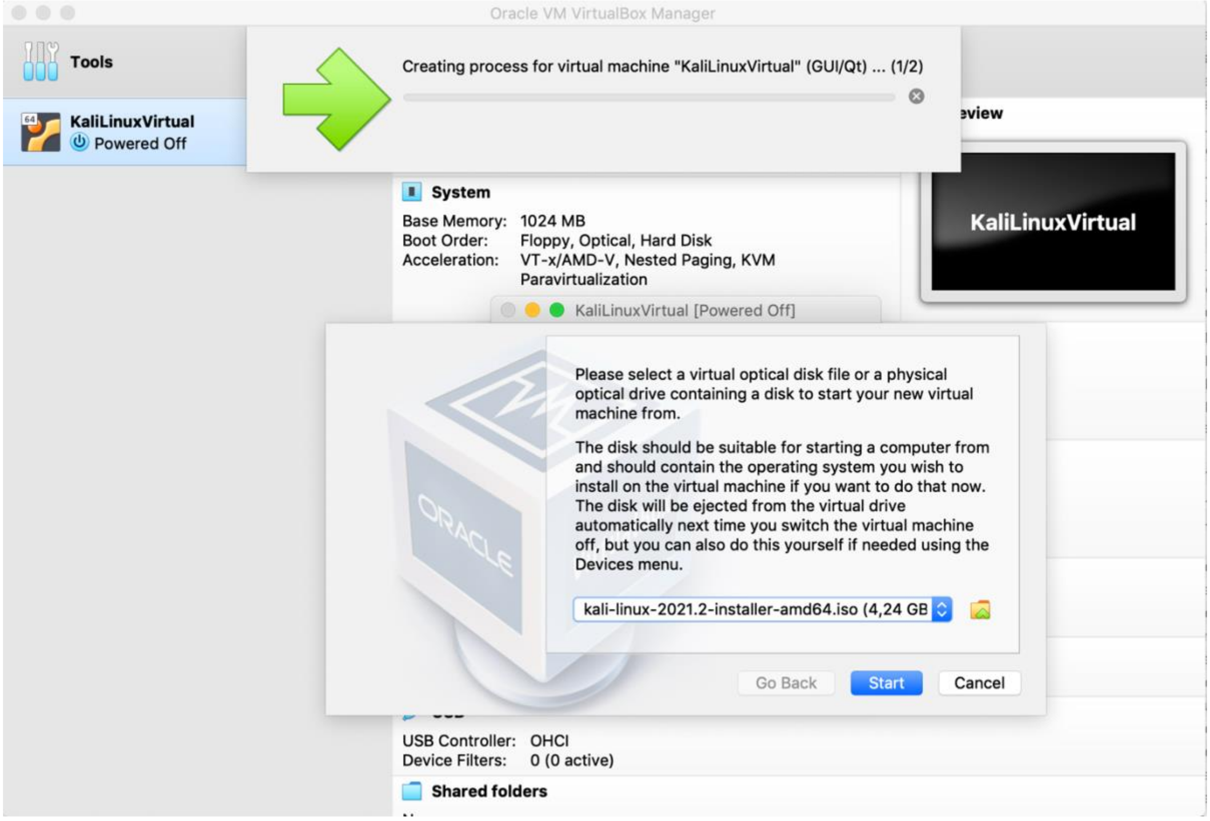
Adapter 1: Intel PRO/1000 MT Desktop (NAT)

USB

USB Controller: OHCI
Device Filters: 0 (0 active)

Shared folders

It provides with information about system, storage, ports. etc. as it would be a standalone computer. The problem is that it is empty – we need to install Kali Linux on it. So, let's boot this virtual machine (hit *Start* icon) and when prompted, provide the path to the Kali Linux ISO file we downloaded.



Creating process for virtual machine "KaliLinuxVirtual" (GUI/Qt) ... (1/2)

System

Base Memory: 1024 MB
Boot Order: Floppy, Optical, Hard Disk
Acceleration: VT-x/AMD-V, Nested Paging, KVM Paravirtualization

Please select a virtual optical disk file or a physical optical drive containing a disk to start your new virtual machine from.

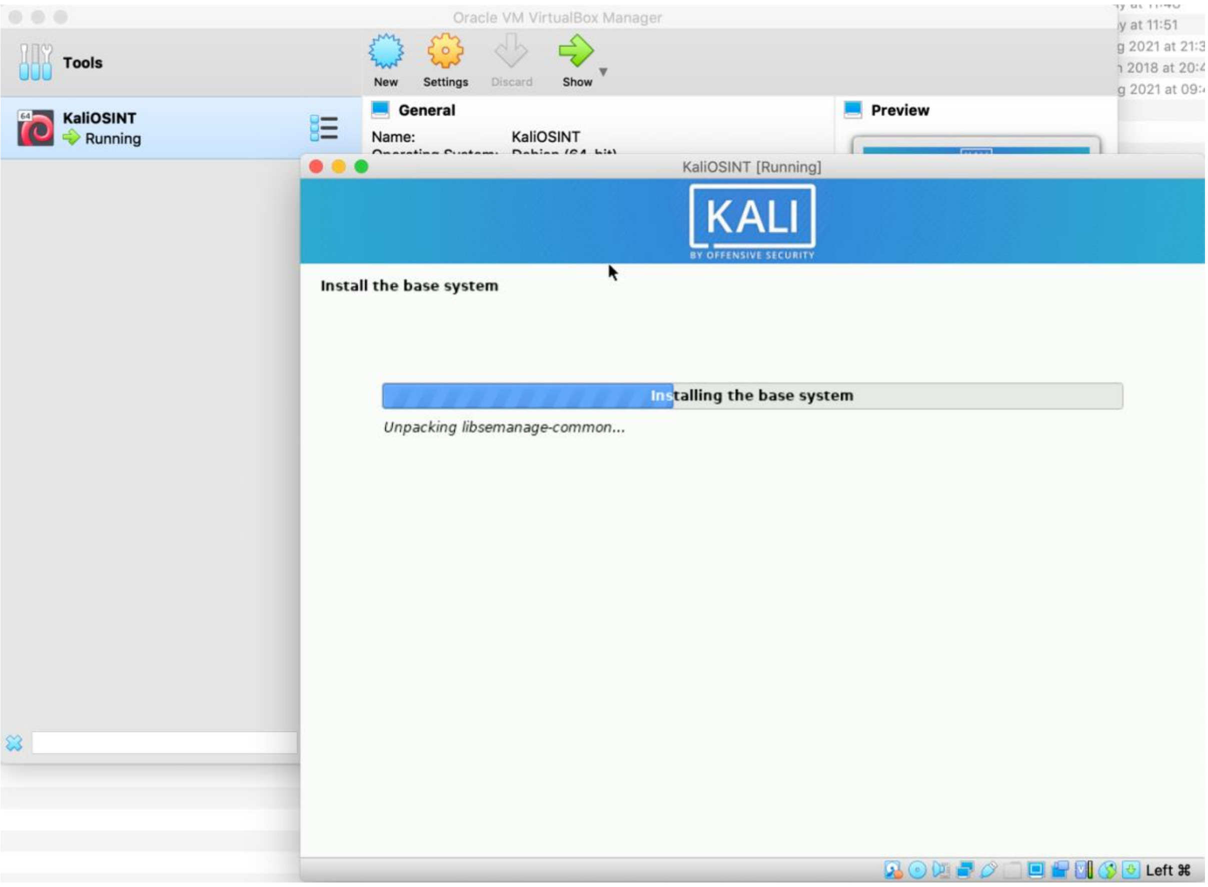
The disk should be suitable for starting a computer from and should contain the operating system you wish to install on the virtual machine if you want to do that now. The disk will be ejected from the virtual drive automatically next time you switch the virtual machine off, but you can also do this yourself if needed using the Devices menu.

kali-linux-2021.2-installer-amd64.iso (4,24 GB

USB Controller: OHCI
Device Filters: 0 (0 active)

Shared folders

Proceed with the installation. Mostly go with default settings. Don’t be confused when it asks you that it will format the whole hard drive. It means the hard drive in the virtual machine only – it does not have the access to your hosting computer OS. Fill hostname and your password to be able to log in into the Kali Linux.



Name: KaliOSINT

Install the base system

Installing the base system

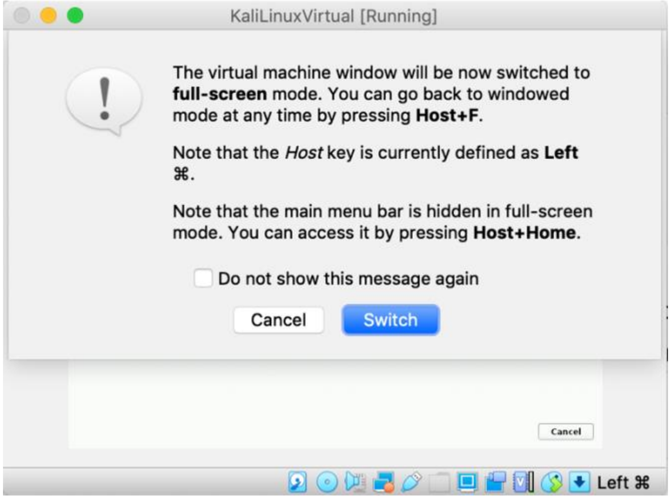
Unpacking libsemanage-common...

VirtualBox Basic Features

You can obviously start, shutdown or reset the machine like with your own computer. What's a unique feature is that you can ***close the virtual machine and save the*** state (*Close -> Save State*). You can have some apps running or have a configuration file opened and not saved. Once you boot your virtual machine once again, you will see all of those apps in the same state while we shut it down.

You can also easily ***clone your virtual machine*** (*Clone* feature). Maybe you are about to do something risky which can damage the operating system. You can easily clone the machine with everything on it as a backup. You may also need more virtual machines than one.

You can run the virtual machine in full screen so then it really looks like you are using Kali Linux in our case. Sometimes the problem might be to get out of the virtual machine back your hosting computer. That's why there is ***a host key*** defined.



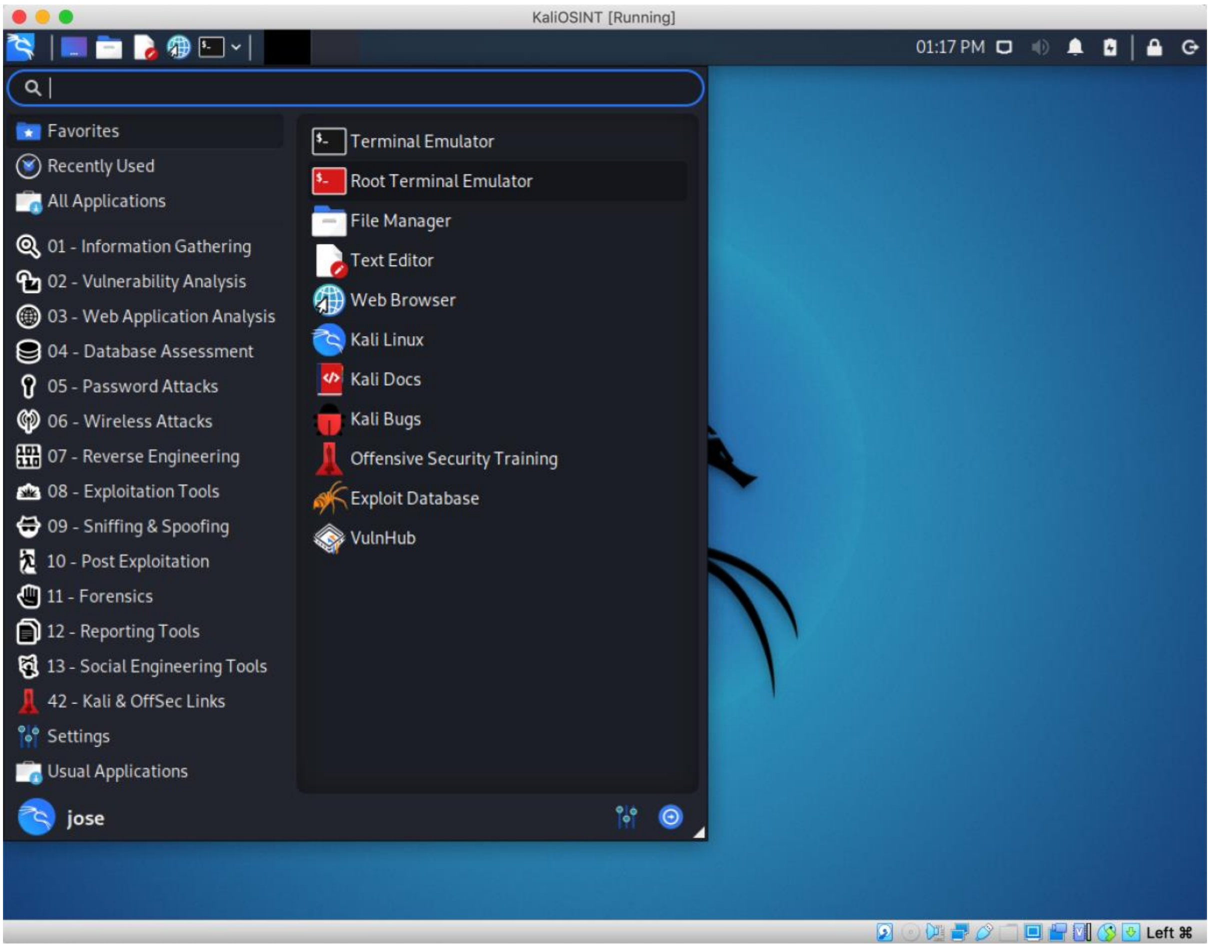
The virtual machine window will be now switched to **full-screen** mode. You can go back to windowed mode at any time by pressing **Host+F**.

Note that the *Host* key is currently defined as **Left ⌘**.

Note that the main menu bar is hidden in full-screen mode. You can access it by pressing **Host+Home**.

□ Do not show this message again

Now you should be able to boot Kali Linux in your virtual machine and log in. As you can notice, Linux doesn't need to be command line only system. If you proceeded the installation with default settings and therefore installed *Xfce* GUI, you should see the following.



You can notice that Kali Linux is already fully loaded with various apps for OSINT and cyber security sorted into the logical categories.

And that is not all. To avoid that Kali Linux will be super heavy distribution, there are other so call Kali Linux Metapackages you can download and install from their website. There is e.g. 3 GB package *kali-linux-forensic*, 1,5 GB package *kali-linux-rfid*, 1,8 GB package *kali-linux-voip* or 6,6 GB package *kali-linux-wireless*.

We will be installing some other tools from other sources as well.

Before we start using any tools, I recommend to get the ropes of Linux shell. The standard GNU shell called *Bash* to be specific but there is *zsh* by default on Kali Linux. Type *echo $SHELL* if you want to know which shell you are on right now. You don't need to know any scripting but some basic syntax would be handy. Throughout my examples you will learn a lot including things such as piping, redirecting program output to files, using program switches and more. If you want to learn from scratch, open this manual at freeCodeCamp for instance.

Let’s Do Some OSINT!

I’m going to demonstrate some of the OSINT tools you can use and why it is beneficial to execute them as a Linux-based apps.

Note: If you don’t want to install your own Linux for some reason, you can just open *Google Cloud Console* and install these programs there.

For the most of the presented scripts, you need Python3 so let’s do the following prerequisites just to be sure we are up-to-date.

You can find out what version of Python you have installed by the following commands:

```
python --version
python2 --version
python3 --version
```

By default, you are not the root (means the administrator on Linux) so as a common user you have to use the command *sudo* to execute commands with the root privileges and you will be asked for the root password:

```
sudo apt-get update
sudo apt-get install python3
sudo apt-get install python3-pip
```

Osintgram

Osintgram is an OSINT tool for Instagram. Obviously. It is not in the standard repertoire of security apps of Kali Linux so we have to download it and install it from Osintgram GitHub project.

Let’s download Osintgram directly from the git by the following command:

```
git clone https://github.com/Datalux/Osintgram.git
```

Skip into the Osintgram directory:

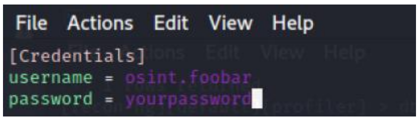
```
cd Osintgram
```

And install Osintgram requirements:

```
pip install -r requirements.txt
```

Before using this tool, I recommend to set up a secondary Instagram account which you are going to use for OSINT purpose only. You have to insert Instagram login and password into this app which you don't want to do with your primary Instagram account. On top of that, we will be testing things in non-standard way of how you normally use Instagram, so possible ban of the used account is also an option. For that reason I created a secondary account *osint.foobar*.

Complete the credentials of your Instagram account in the file config/credentials.ini. You can use one of the text editors such as *vim* (that's what I prefer) and *nano*. Or install an editor with graphical user interface such as *xedit* or *gedit* (this one is not in Kali Linux by default so you can install it with this command: *sudo apt-get install gedit*).



```
[Credentials]
username = osint.foobar
password = yourpassword
```

You also need to edit the file *settings.json* to look as follows:

{}

The easiest way how to do that if you are in the *Osintgram/config* directory is with the following command:

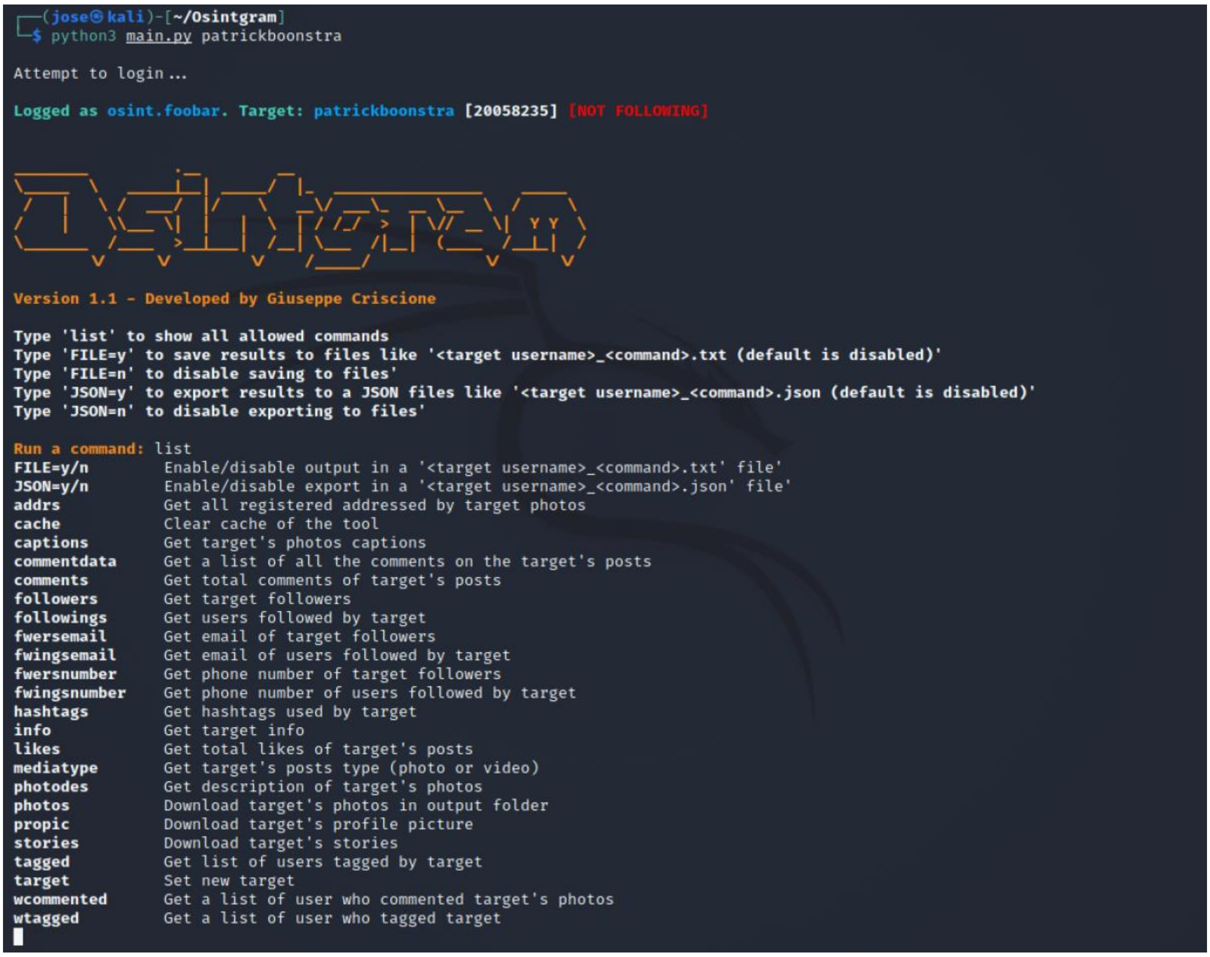
echo "{}" > settings.json

Osintgram is scripted in Python and you launch it in the following way:

python3 main.py <target_Instagram_nickname>

As usual I use an account of my friend, a voluntary guinea pig, Patrick Boonstra from the Netherlands. Thanks Pat!

python3 main.py patrickboonstra



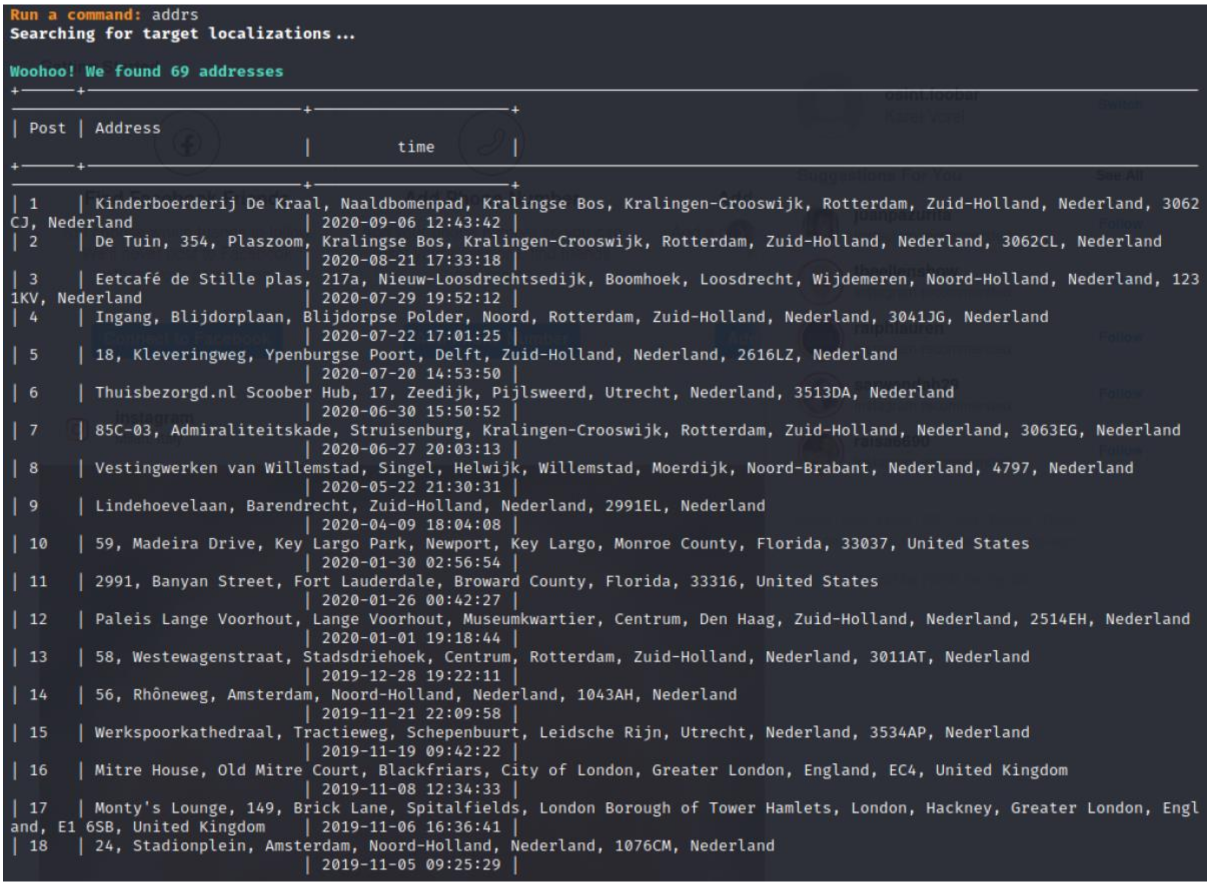
```
┌──(jose㉿kali)-[~/Osintgram]
└─$ python3 main.py patrickboonstra

Attempt to login...

Logged as osint.foobar. Target: patrickboonstra [20058235] [NOT FOLLOWING]


Version 1.1 - Developed by Giuseppe Criscione

Type 'list' to show all allowed commands
Type 'FILE=y' to save results to files like '<target username>_<command>.txt (default is disabled)'
Type 'FILE=n' to disable saving to files'
Type 'JSON=y' to export results to a JSON files like '<target username>_<command>.json (default is disabled)'
Type 'JSON=n' to disable exporting to files'

Run a command: list
FILE=y/n        Enable/disable output in a '<target username>_<command>.txt' file'
JSON=y/n        Enable/disable export in a '<target username>_<command>.json' file'
addrs           Get all registered addressed by target photos
cache           Clear cache of the tool
captions        Get target's photos captions
commentdata     Get a list of all the comments on the target's posts
comments        Get total comments of target's posts
followers       Get target followers
followings      Get users followed by target
fwersemail      Get email of target followers
fwingsemail     Get email of users followed by target
fwersnumber     Get phone number of target followers
fwingsnumber    Get phone number of users followed by target
hashtags        Get hashtags used by target
info            Get target info
likes           Get total likes of target's posts
mediatype       Get target's posts type (photo or video)
photodes        Get description of target's photos
photos          Download target's photos in output folder
propic          Download target's profile picture
stories         Download target's stories
tagged          Get list of users tagged by target
target          Set new target
wcommented      Get a list of user who commented target's photos
wtagged         Get a list of user who tagged target
```

We can notice Patrick's Instagram ID and also the fact that we don't follow his account with our OSINT Instagram account.

I run the command *list* to see all the options I have with this tool. Let's go through the most interesting ones.

addrs

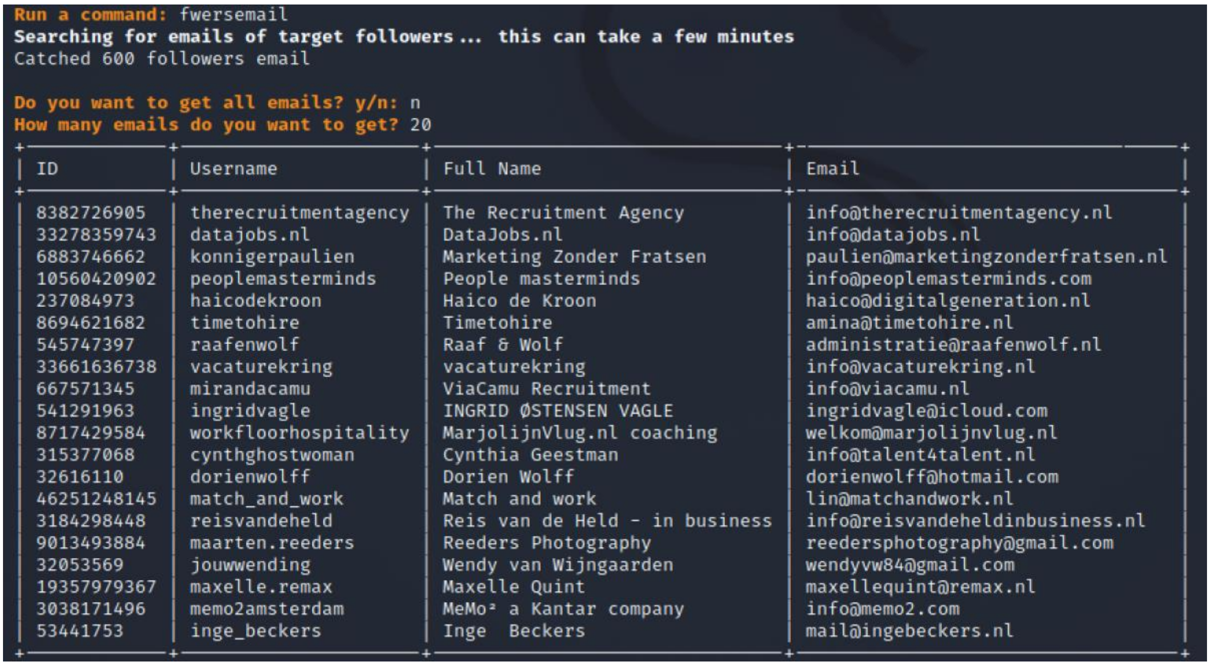
This command is going through all target's posts and gather the registered address whenever possible.



```
Run a command: addrs
Searching for target localizations ...

Woohoo! We found 69 addresses
+------+--------------------------------------------------------------------------------------------------
-----------------------------+--------------------------+
| Post | Address
                             |          time            |
+------+--------------------------------------------------------------------------------------------------
-----------------------------+--------------------------+
| 1    | Kinderboerderij De Kraal, Naaldbomenpad, Kralingse Bos, Kralingen-Crooswijk, Rotterdam, Zuid-Holland, Nederland, 3062
CJ, Nederland                | 2020-09-06 12:43:42 |
| 2    | De Tuin, 354, Plaszoom, Kralingse Bos, Kralingen-Crooswijk, Rotterdam, Zuid-Holland, Nederland, 3062CL, Nederland
                             | 2020-08-21 17:33:18 |
| 3    | Eetcafé de Stille plas, 217a, Nieuw-Loosdrechtsedijk, Boomhoek, Loosdrecht, Wijdemeren, Noord-Holland, Nederland, 123
1KV, Nederland               | 2020-07-29 19:52:12 |
| 4    | Ingang, Blijdorplaan, Blijdorpse Polder, Noord, Rotterdam, Zuid-Holland, Nederland, 3041JG, Nederland
                             | 2020-07-22 17:01:25 |
| 5    | 18, Kleveringweg, Ypenburgse Poort, Delft, Zuid-Holland, Nederland, 2616LZ, Nederland
                             | 2020-07-20 14:53:50 |
| 6    | Thuisbezorgd.nl Scoober Hub, 17, Zeedijk, Pijlsweerd, Utrecht, Nederland, 3513DA, Nederland
                             | 2020-06-30 15:50:52 |
| 7    | 85C-03, Admiraliteitskade, Struisenburg, Kralingen-Crooswijk, Rotterdam, Zuid-Holland, Nederland, 3063EG, Nederland
                             | 2020-06-27 20:03:13 |
| 8    | Vestingwerken van Willemstad, Singel, Helwijk, Willemstad, Moerdijk, Noord-Brabant, Nederland, 4797, Nederland
                             | 2020-05-22 21:30:31 |
| 9    | Lindehoevelaan, Barendrecht, Zuid-Holland, Nederland, 2991EL, Nederland
                             | 2020-04-09 18:04:08 |
| 10   | 59, Madeira Drive, Key Largo Park, Newport, Key Largo, Monroe County, Florida, 33037, United States
                             | 2020-01-30 02:56:54 |
| 11   | 2991, Banyan Street, Fort Lauderdale, Broward County, Florida, 33316, United States
                             | 2020-01-26 00:42:27 |
| 12   | Paleis Lange Voorhout, Lange Voorhout, Museumkwartier, Centrum, Den Haag, Zuid-Holland, Nederland, 2514EH, Nederland
                             | 2020-01-01 19:18:44 |
| 13   | 58, Westewagenstraat, Stadsdriehoek, Centrum, Rotterdam, Zuid-Holland, Nederland, 3011AT, Nederland
                             | 2019-12-28 19:22:11 |
| 14   | 56, Rhôneweg, Amsterdam, Noord-Holland, Nederland, 1043AH, Nederland
                             | 2019-11-21 22:09:58 |
| 15   | Werkspoorkathedraal, Tractieweg, Schepenbuurt, Leidsche Rijn, Utrecht, Nederland, 3534AP, Nederland
                             | 2019-11-19 09:42:22 |
| 16   | Mitre House, Old Mitre Court, Blackfriars, City of London, Greater London, England, EC4, United Kingdom
                             | 2019-11-08 12:34:33 |
| 17   | Monty's Lounge, 149, Brick Lane, Spitalfields, London Borough of Tower Hamlets, London, Hackney, Greater London, Engl
and, E1 6SB, United Kingdom  | 2019-11-06 16:36:41 |
| 18   | 24, Stadionplein, Amsterdam, Noord-Holland, Nederland, 1076CM, Nederland
                             | 2019-11-05 09:25:29 |
```

fwersemail

Get email addresses of target’s followers is not problem at all. I limit the output to 20 email addresses.

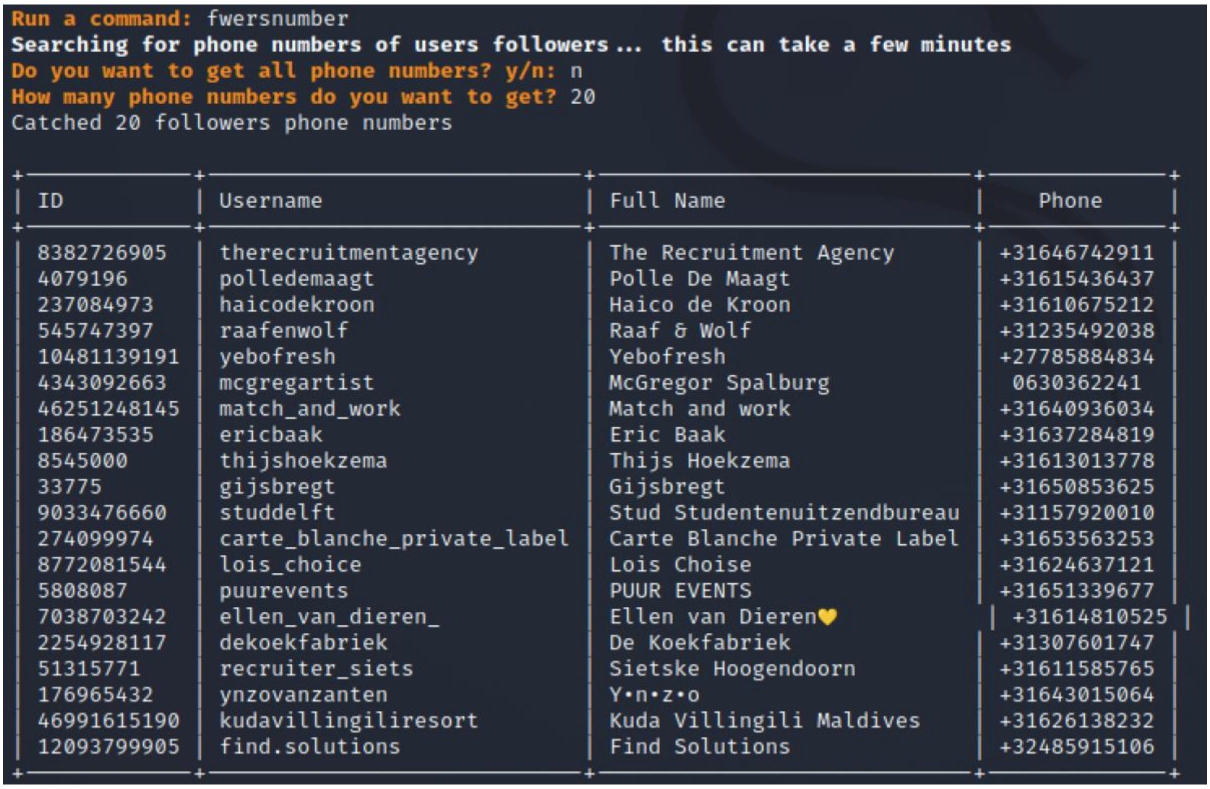


```
Run a command: fwersemail
Searching for emails of target followers ... this can take a few minutes
Catched 600 followers email

Do you want to get all emails? y/n: n
How many emails do you want to get? 20
+-------------+--------------------+------------------------------+------------------------------------+
| ID          | Username           | Full Name                    | Email                              |
+-------------+--------------------+------------------------------+------------------------------------+
| 8382726905  | therecruitmentagency | The Recruitment Agency     | info@therecruitmentagency.nl       |
| 33278359743 | datajobs.nl        | DataJobs.nl                  | info@datajobs.nl                   |
| 6883746662  | konnigerpaulien    | Marketing Zonder Fratsen     | paulien@marketingzonderfratsen.nl  |
| 10560420902 | peoplemasterminds  | People masterminds           | info@peoplemasterminds.com         |
| 237084973   | haicodekroon       | Haico de Kroon               | haico@digitalgeneration.nl         |
| 8694621682  | timetohire         | Timetohire                   | amina@timetohire.nl                |
| 545747397   | raafenwolf         | Raaf & Wolf                  | administratie@raafenwolf.nl        |
| 33661636738 | vacaturekring      | vacaturekring                | info@vacaturekring.nl              |
| 667571345   | mirandacamu        | ViaCamu Recruitment          | info@viacamu.nl                    |
| 541291963   | ingridvagle        | INGRID ØSTENSEN VAGLE        | ingridvagle@icloud.com             |
| 8717429584  | workfloorhospitality | MarjolijnVlug.nl coaching  | welkom@marjolijnvlug.nl            |
| 315377068   | cynthghostwoman    | Cynthia Geestman             | info@talent4talent.nl              |
| 32616110    | dorienwolff        | Dorien Wolff                 | dorienwolff@hotmail.com            |
| 46251248145 | match_and_work     | Match and work               | lin@matchandwork.nl                |
| 3184298448  | reisvandeheld      | Reis van de Held - in business | info@reisvandeheldinbusiness.nl  |
| 9013493884  | maarten.reeders    | Reeders Photography          | reedersphotography@gmail.com       |
| 32053569    | jouwwending        | Wendy van Wijngaarden        | wendyvw84@gmail.com                |
| 19357979367 | maxelle.remax      | Maxelle Quint                | maxellequint@remax.nl              |
| 3038171496  | memo2amsterdam     | MeMo² a Kantar company       | info@memo2.com                     |
| 53441753    | inge_beckers       | Inge  Beckers                | mail@ingebeckers.nl                |
+-------------+--------------------+------------------------------+------------------------------------+
```

If you want to get the email addresses of people followed by the target, use **fwingsemail**.

fwersnumber

The similar method we did with email addresses can be done with telephone numbers. These are 20 telephone numbers of Patrick's followers.

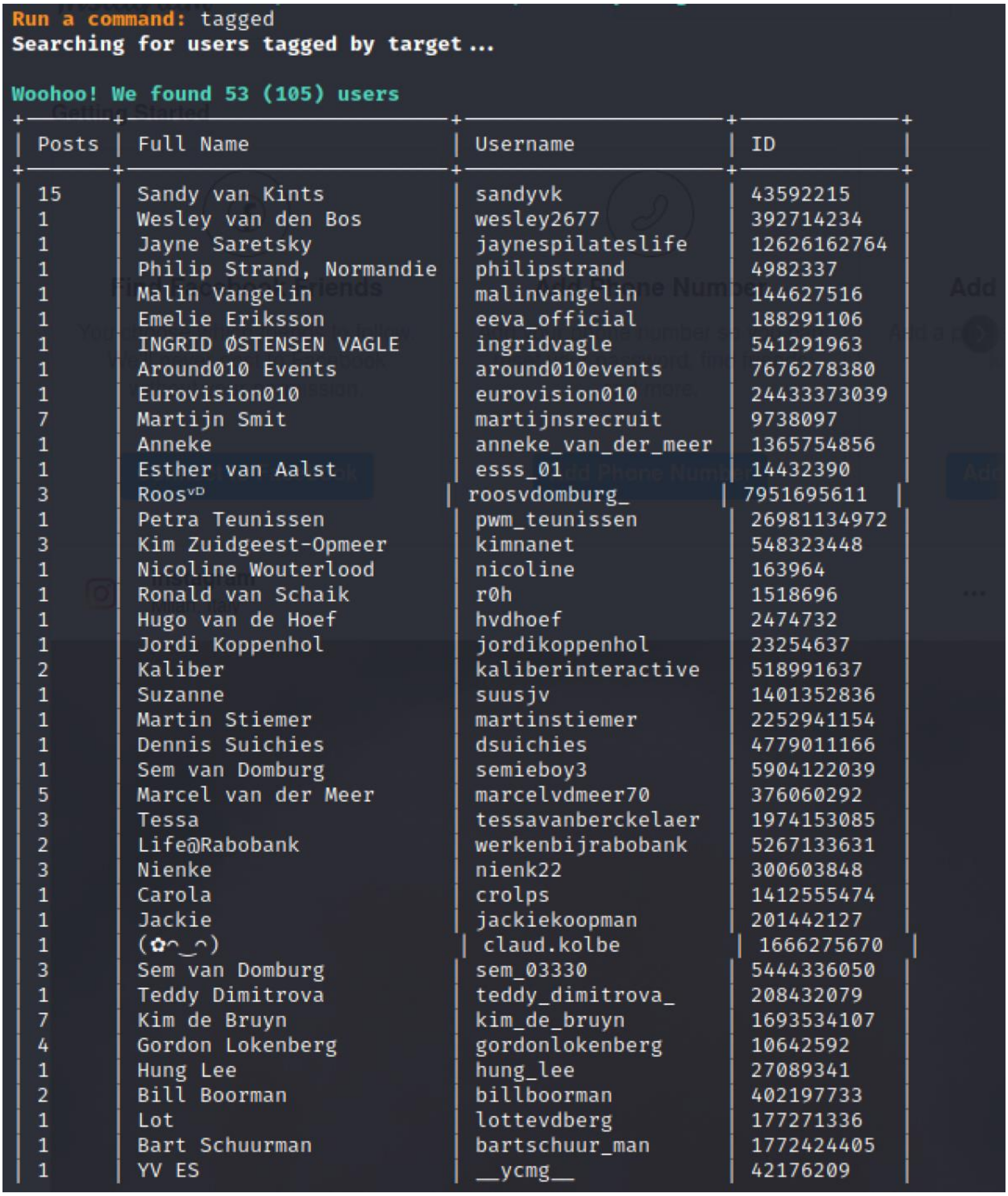


```
Run a command: fwersnumber
Searching for phone numbers of users followers ... this can take a few minutes
Do you want to get all phone numbers? y/n: n
How many phone numbers do you want to get? 20
Catched 20 followers phone numbers

+-------------+----------------------------+----------------------------+-------------+
| ID          | Username                   | Full Name                  |    Phone    |
+-------------+----------------------------+----------------------------+-------------+
| 8382726905  | therecruitmentagency       | The Recruitment Agency     | +31646742911 |
| 4079196     | polledemaagt               | Polle De Maagt             | +31615436437 |
| 237084973   | haicodekroon               | Haico de Kroon             | +31610675212 |
| 545747397   | raafenwolf                 | Raaf & Wolf                | +31235492038 |
| 10481139191 | yebofresh                  | Yebofresh                  | +27785884834 |
| 4343092663  | mcgregartist               | McGregor Spalburg          |  0630362241  |
| 46251248145 | match_and_work             | Match and work             | +31640936034 |
| 186473535   | ericbaak                   | Eric Baak                  | +31637284819 |
| 8545000     | thijshoekzema              | Thijs Hoekzema             | +31613013778 |
| 33775       | gijsbregt                  | Gijsbregt                  | +31650853625 |
| 9033476660  | studdelft                  | Stud Studentenuitzendbureau | +31157920010 |
| 274099974   | carte_blanche_private_label | Carte Blanche Private Label | +31653563253 |
| 8772081544  | lois_choice                | Lois Choise                | +31624637121 |
| 5808087     | puurevents                 | PUUR EVENTS                | +31651339677 |
| 7038703242  | ellen_van_dieren_          | Ellen van Dieren💛          |  +31614810525 |
| 2254928117  | dekoekfabriek              | De Koekfabriek             | +31307601747 |
| 51315771    | recruiter_siets            | Sietske Hoogendoorn        | +31611585765 |
| 176965432   | ynzovanzanten              | Y•n•z•o                    | +31643015064 |
| 46991615190 | kudavillingiliresort       | Kuda Villingili Maldives   | +31626138232 |
| 12093799905 | find.solutions             | Find Solutions             | +32485915106 |
+-------------+----------------------------+----------------------------+-------------+
```

The telephone numbers of people who Patrick's a.k.a. target follow would be gathered with the command **fwingsnumber**.

tagged

If you want to quickly find out which users the target tags the most in his/her comments, use this command.

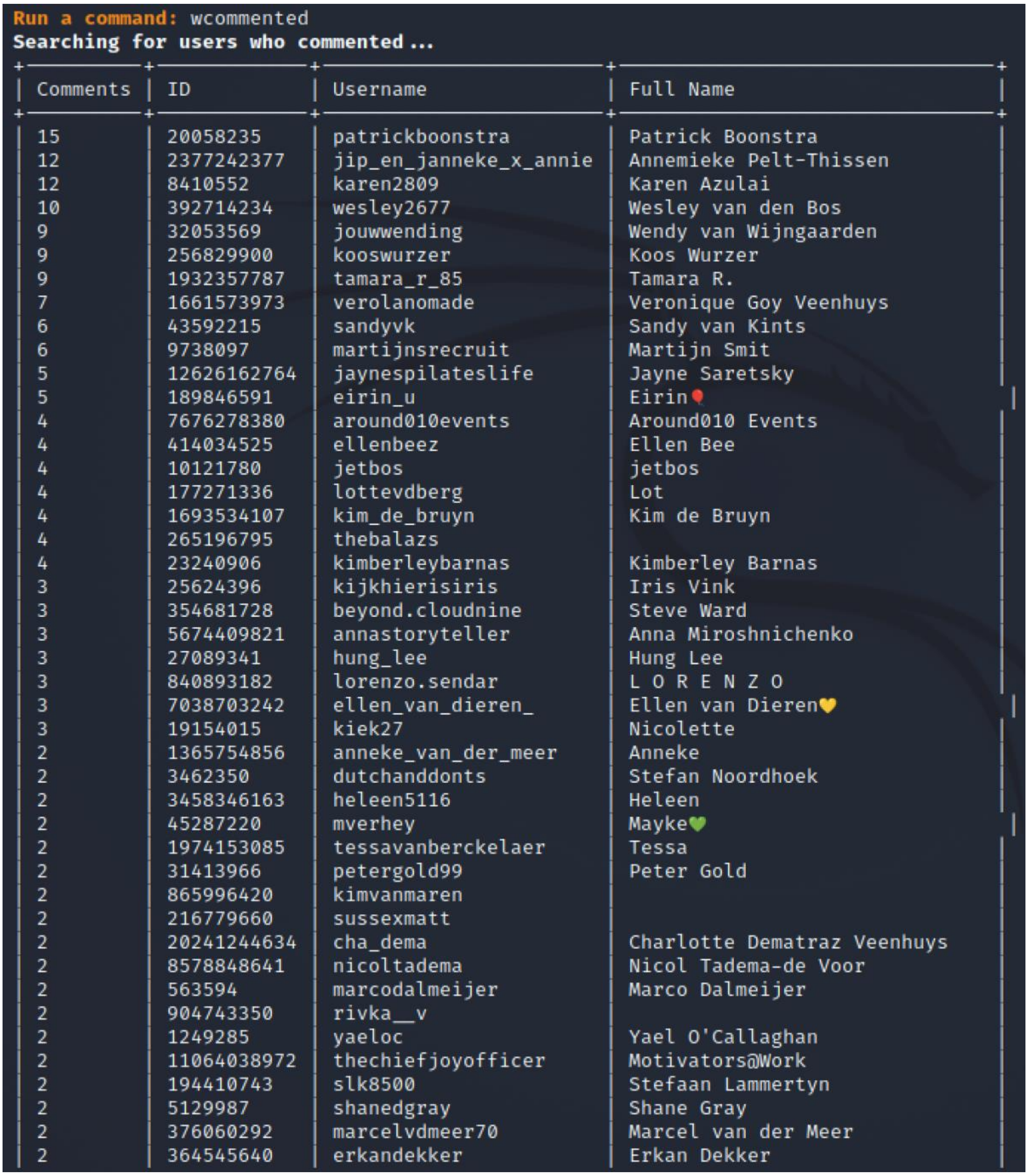


```
Run a command: tagged
Searching for users tagged by target ...

Woohoo! We found 53 (105) users
+-------+---------------------------+----------------------+-------------+
| Posts | Full Name                 | Username             | ID          |
+-------+---------------------------+----------------------+-------------+
| 15    | Sandy van Kints           | sandyvk              | 43592215    |
| 1     | Wesley van den Bos        | wesley2677           | 392714234   |
| 1     | Jayne Saretsky            | jaynespilateslife    | 12626162764 |
| 1     | Philip Strand, Normandie  | philipstrand         | 4982337     |
| 1     | Malin Vangelin            | malinvangelin        | 144627516   |
| 1     | Emelie Eriksson           | eeva_official        | 188291106   |
| 1     | INGRID ØSTENSEN VAGLE     | ingridvagle          | 541291963   |
| 1     | Around010 Events          | around010events      | 7676278380  |
| 1     | Eurovision010             | eurovision010        | 24433373039 |
| 7     | Martijn Smit              | martijnsrecruit      | 9738097     |
| 1     | Anneke                    | anneke_van_der_meer  | 1365754856  |
| 1     | Esther van Aalst          | esss_01              | 14432390    |
| 3     | Roosᵛᴰ                    | roosvdomburg_        | 7951695611  |
| 1     | Petra Teunissen           | pwm_teunissen        | 26981134972 |
| 3     | Kim Zuidgeest-Opmeer      | kimnanet             | 548323448   |
| 1     | Nicoline Wouterlood       | nicoline             | 163964      |
| 1     | Ronald van Schaik         | r0h                  | 1518696     |
| 1     | Hugo van de Hoef          | hvdhoef              | 2474732     |
| 1     | Jordi Koppenhol           | jordikoppenhol       | 23254637    |
| 2     | Kaliber                   | kaliberinteractive   | 518991637   |
| 1     | Suzanne                   | suusjv               | 1401352836  |
| 1     | Martin Stiemer            | martinstiemer        | 2252941154  |
| 1     | Dennis Suichies           | dsuichies            | 4779011166  |
| 1     | Sem van Domburg           | semieboy3            | 5904122039  |
| 5     | Marcel van der Meer       | marcelvdmeer70       | 376060292   |
| 3     | Tessa                     | tessavanberckelaer   | 1974153085  |
| 2     | Life@Rabobank             | werkenbijrabobank    | 5267133631  |
| 3     | Nienke                    | nienk22              | 300603848   |
| 1     | Carola                    | crolps               | 1412555474  |
| 1     | Jackie                    | jackiekoopman        | 201442127   |
| 1     | (✿◠‿◠)                    | claud.kolbe          | 1666275670  |
| 3     | Sem van Domburg           | sem_03330            | 5444336050  |
| 1     | Teddy Dimitrova           | teddy_dimitrova_     | 208432079   |
| 7     | Kim de Bruyn              | kim_de_bruyn         | 1693534107  |
| 4     | Gordon Lokenberg          | gordonlokenberg      | 10642592    |
| 1     | Hung Lee                  | hung_lee             | 27089341    |
| 2     | Bill Boorman              | billboorman          | 402197733   |
| 1     | Lot                       | lottevdberg          | 177271336   |
| 1     | Bart Schuurman            | bartschuur_man       | 1772424405  |
| 1     | YV ES                     | __ycmg__             | 42176209    |
```

If you want to quickly get the list of users who tagged the target, there is a command for that too - **wtagged**.

wcommented

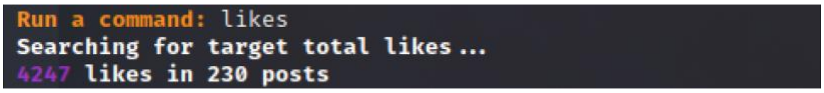
One interesting list might be people who commented on target’s posts.



```
Run a command: wcommented
Searching for users who commented ...
+----------+-------------+------------------------+---------------------------------+
| Comments | ID          | Username               | Full Name                       |
+----------+-------------+------------------------+---------------------------------+
| 15       | 20058235    | patrickboonstra        | Patrick Boonstra                |
| 12       | 2377242377  | jip_en_janneke_x_annie | Annemieke Pelt-Thissen          |
| 12       | 8410552     | karen2809              | Karen Azulai                    |
| 10       | 392714234   | wesley2677             | Wesley van den Bos              |
| 9        | 32053569    | jouwwending            | Wendy van Wijngaarden           |
| 9        | 256829900   | kooswurzer             | Koos Wurzer                     |
| 9        | 1932357787  | tamara_r_85            | Tamara R.                       |
| 7        | 1661573973  | verolanomade           | Veronique Goy Veenhuys          |
| 6        | 43592215    | sandyvk                | Sandy van Kints                 |
| 6        | 9738097     | martijnsrecruit        | Martijn Smit                    |
| 5        | 12626162764 | jaynespilateslife      | Jayne Saretsky                  |
| 5        | 189846591   | eirin_u                | Eirin🎈                          |
| 4        | 7676278380  | around010events        | Around010 Events                |
| 4        | 414034525   | ellenbeez              | Ellen Bee                       |
| 4        | 10121780    | jetbos                 | jetbos                          |
| 4        | 177271336   | lottevdberg            | Lot                             |
| 4        | 1693534107  | kim_de_bruyn           | Kim de Bruyn                    |
| 4        | 265196795   | thebalazs              |                                 |
| 4        | 23240906    | kimberleybarnas        | Kimberley Barnas                |
| 3        | 25624396    | kijkhierisiris         | Iris Vink                       |
| 3        | 354681728   | beyond.cloudnine       | Steve Ward                      |
| 3        | 5674409821  | annastoryteller        | Anna Miroshnichenko             |
| 3        | 27089341    | hung_lee               | Hung Lee                        |
| 3        | 840893182   | lorenzo.sendar         | L O R E N Z O                   |
| 3        | 7038703242  | ellen_van_dieren_      | Ellen van Dieren💛               |
| 3        | 19154015    | kiek27                 | Nicolette                       |
| 2        | 1365754856  | anneke_van_der_meer    | Anneke                          |
| 2        | 3462350     | dutchanddonts          | Stefan Noordhoek                |
| 2        | 3458346163  | heleen5116             | Heleen                          |
| 2        | 45287220    | mverhey                | Mayke💚                          |
| 2        | 1974153085  | tessavanberckelaer     | Tessa                           |
| 2        | 31413966    | petergold99            | Peter Gold                      |
| 2        | 865996420   | kimvanmaren            |                                 |
| 2        | 216779660   | sussexmatt             |                                 |
| 2        | 20241244634 | cha_dema               | Charlotte Dematraz Veenhuys     |
| 2        | 8578848641  | nicoltadema            | Nicol Tadema-de Voor            |
| 2        | 563594      | marcodalmeijer         | Marco Dalmeijer                 |
| 2        | 904743350   | rivka__v               |                                 |
| 2        | 1249285     | yaeloc                 | Yael O'Callaghan                |
| 2        | 11064038972 | thechiefjoyofficer     | Motivators@Work                 |
| 2        | 194410743   | slk8500                | Stefaan Lammertyn               |
| 2        | 5129987     | shanedgray             | Shane Gray                      |
| 2        | 376060292   | marcelvdmeer70         | Marcel van der Meer             |
| 2        | 364545640   | erkandekker            | Erkan Dekker                    |
```

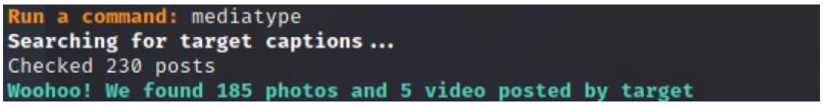
likes

You can get the lump sum of all likes for all the target’s posts.



```
Run a command: likes
Searching for target total likes ...
4247 likes in 230 posts
```

mediatype

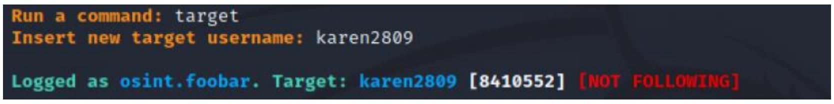


```
Run a command: mediatype
Searching for target captions ...
Checked 230 posts
Woohoo! We found 185 photos and 5 video posted by target
```

There are many commands which are self-explanatory and they mostly download content such as stories, comments, captions, followers, followings and others. They include commands: **captions, commentdata, comments, followers, followings, hashtags, photodec, propic** and **stories**.

target

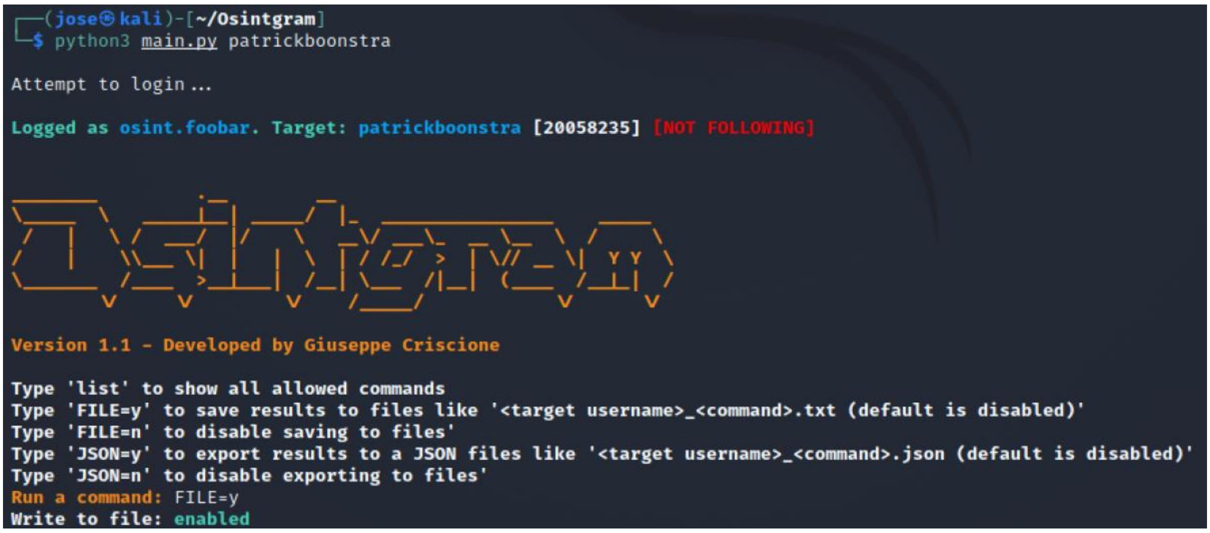
With this command you can quickly change the target to someone else. In my case I noticed that Karen Azulai commented quite often to Patrick's posts, so let's switch the target to her.



```
Run a command: target
Insert new target username: karen2809

Logged as osint.foobar. Target: karen2809 [8410552] [NOT FOLLOWING]
```

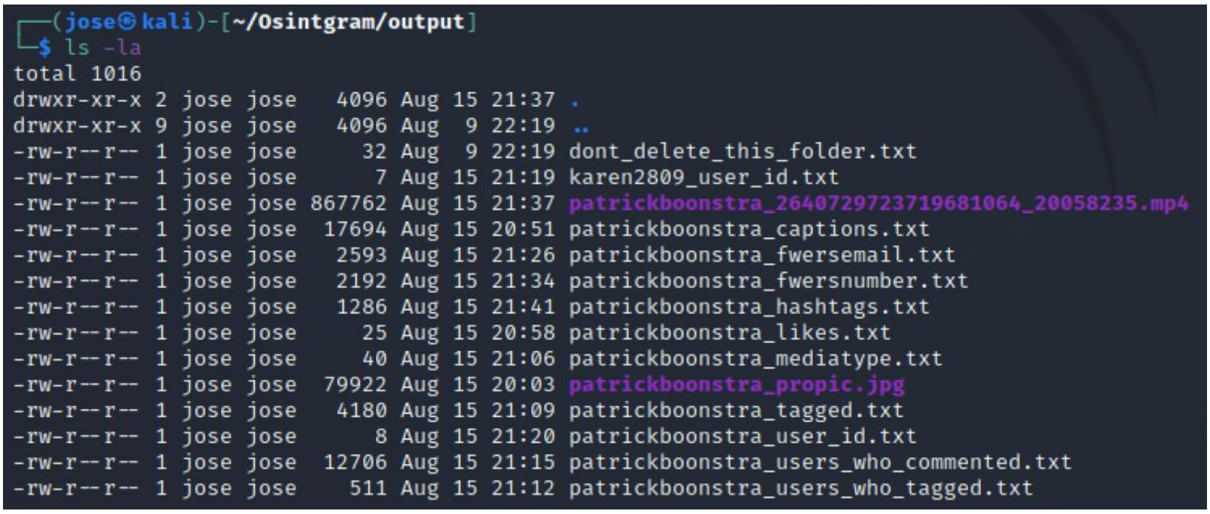
Processing the output

The beauty of the Linux shell is that it is very responsive. *Osintgram* is a sort of interactive command line application so you cannot utilize pipes or output redirecting like with the non-interactive commands (I will show you later). The thing when you start Osintgram, you can enable logging of all outputs you get from the used commands by a command **FILE=y**.



```
┌──(jose㉿kali)-[~/Osintgram]
└─$ python3 main.py patrickboonstra

Attempt to login ...

Logged as osint.foobar. Target: patrickboonstra [20058235] [NOT FOLLOWING]


________         .__        __                               
\_____  \   _____|__| _____/  |_  ________________    _____  
 /   |   \ /  ___/  |/    \   __\/ ___\_  __ \__  \  /     \ 
/    |    \\___ \|  |   |  \  | / /_/  >  | \// __ \|  Y Y  \
\_______  /____  >__|___|  /__| \___  /|__|  (____  /__|_|  /
        \/     \/        \/    /_____/            \/      \/ 

Version 1.1 - Developed by Giuseppe Criscione

Type 'list' to show all allowed commands
Type 'FILE=y' to save results to files like '<target username>_<command>.txt (default is disabled)'
Type 'FILE=n' to disable saving to files'
Type 'JSON=y' to export results to a JSON files like '<target username>_<command>.json (default is disabled)'
Type 'JSON=n' to disable exporting to files'
Run a command: FILE=y
Write to file: enabled
```

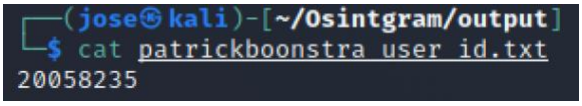
All output data is then stored in a separate directory *output* in the *Osintgram* directory.



```
┌──(jose㉿kali)-[~/Osintgram/output]
└─$ ls -la
total 1016
drwxr-xr-x 2 jose jose   4096 Aug 15 21:37 .
drwxr-xr-x 9 jose jose   4096 Aug  9 22:19 ..
-rw-r--r-- 1 jose jose     32 Aug  9 22:19 dont_delete_this_folder.txt
-rw-r--r-- 1 jose jose      7 Aug 15 21:19 karen2809_user_id.txt
-rw-r--r-- 1 jose jose 867762 Aug 15 21:37 patrickboonstra_2640729723719681064_20058235.mp4
-rw-r--r-- 1 jose jose  17694 Aug 15 20:51 patrickboonstra_captions.txt
-rw-r--r-- 1 jose jose   2593 Aug 15 21:26 patrickboonstra_fwersemail.txt
-rw-r--r-- 1 jose jose   2192 Aug 15 21:34 patrickboonstra_fwersnumber.txt
-rw-r--r-- 1 jose jose   1286 Aug 15 21:41 patrickboonstra_hashtags.txt
-rw-r--r-- 1 jose jose     25 Aug 15 20:58 patrickboonstra_likes.txt
-rw-r--r-- 1 jose jose     40 Aug 15 21:06 patrickboonstra_mediatype.txt
-rw-r--r-- 1 jose jose  79922 Aug 15 20:03 patrickboonstra_propic.jpg
-rw-r--r-- 1 jose jose   4180 Aug 15 21:09 patrickboonstra_tagged.txt
-rw-r--r-- 1 jose jose      8 Aug 15 21:20 patrickboonstra_user_id.txt
-rw-r--r-- 1 jose jose  12706 Aug 15 21:15 patrickboonstra_users_who_commented.txt
-rw-r--r-- 1 jose jose    511 Aug 15 21:12 patrickboonstra_users_who_tagged.txt
```

Here comes the efficient work in Linux prompt. If you want to open a text file, you have several options.

If you want just to glimpse into the file, you can use the Linux command **cat** followed by the path to the file. If we are in the same directory where the file is presented, we can use its name only without a path.

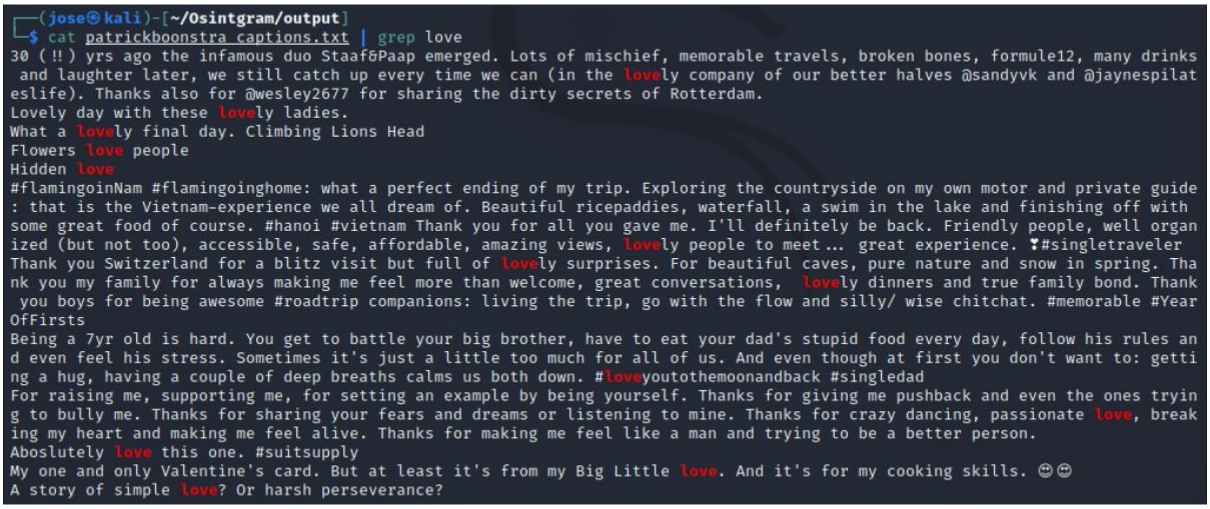


```
┌──(jose㉿kali)-[~/Osintgram/output]
└─$ cat patrickboonstra_user_id.txt
20058235
```

This might be inconvenient for files with a lot of content but you can already use some post processing using pipes.

Let's say we want to filter all Instagram photo captions containing the keyword *love*. We can do that by sending the output from cat command to grep command which can be done over a pipe sing |.

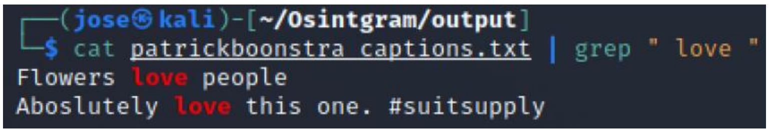
cat patrickboonstra_captions.txt | grep love



```
┌──(jose㉿kali)-[~/Osintgram/output]
└─$ cat patrickboonstra_captions.txt | grep love
30 (!!) yrs ago the infamous duo Staaf&Paap emerged. Lots of mischief, memorable travels, broken bones, formule12, many drinks
 and laughter later, we still catch up every time we can (in the lovely company of our better halves @sandyvk and @jaynespilat
eslife). Thanks also for @wesley2677 for sharing the dirty secrets of Rotterdam.
Lovely day with these lovely ladies.
What a lovely final day. Climbing Lions Head
Flowers love people
Hidden love
#flamingoinNam #flamingoinghome: what a perfect ending of my trip. Exploring the countryside on my own motor and private guide
: that is the Vietnam-experience we all dream of. Beautiful ricepaddies, waterfall, a swim in the lake and finishing off with
some great food of course. #hanoi #vietnam Thank you for all you gave me. I'll definitely be back. Friendly people, well organ
ized (but not too), accessible, safe, affordable, amazing views, lovely people to meet… great experience. ❣#singletraveler
Thank you Switzerland for a blitz visit but full of lovely surprises. For beautiful caves, pure nature and snow in spring. Tha
nk you my family for always making me feel more than welcome, great conversations,  lovely dinners and true family bond. Thank
 you boys for being awesome #roadtrip companions: living the trip, go with the flow and silly/ wise chitchat. #memorable #Year
OfFirsts
Being a 7yr old is hard. You get to battle your big brother, have to eat your dad's stupid food every day, follow his rules an
d even feel his stress. Sometimes it's just a little too much for all of us. And even though at first you don't want to: getti
ng a hug, having a couple of deep breaths calms us both down. #loveyoutothemoonandback #singledad
For raising me, supporting me, for setting an example by being yourself. Thanks for giving me pushback and even the ones tryin
g to bully me. Thanks for sharing your fears and dreams or listening to mine. Thanks for crazy dancing, passionate love, break
ing my heart and making me feel alive. Thanks for making me feel like a man and trying to be a better person.
Aboslutely love this one. #suitsupply
My one and only Valentine's card. But at least it's from my Big Little love. And it's for my cooking skills. 😊😊
A story of simple love? Or harsh perseverance?
```

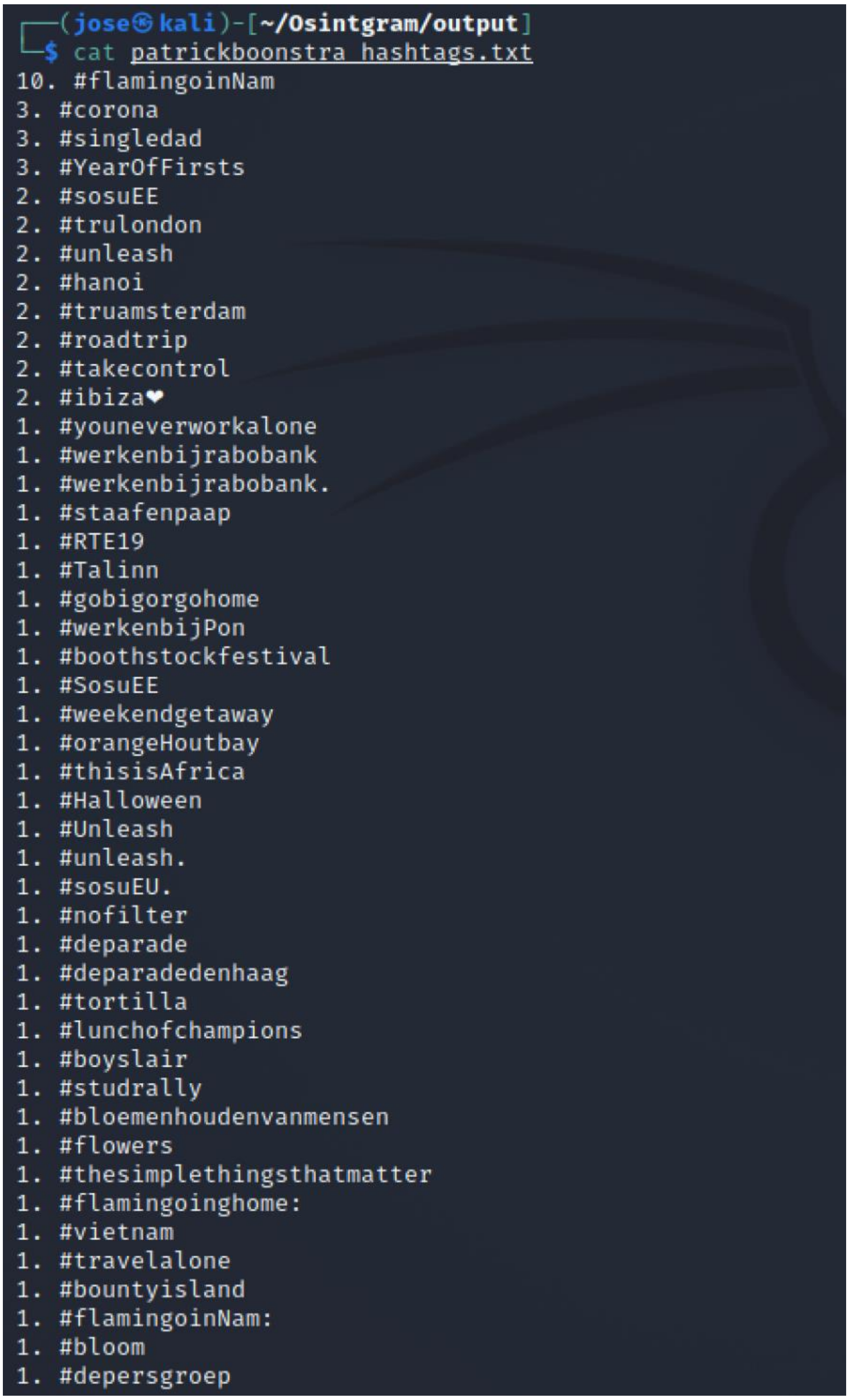
We can see that the output is not ideal as it included other words containing the word *love* such as *lovely*. Linux shell supports regular expressions but in this it is easy fix to get a word *love* which is not a part of any other word or a hashtag.

cat patrickboonstra_captions.txt | grep " love "



```
┌──(jose㉿kali)-[~/Osintgram/output]
└─$ cat patrickboonstra_captions.txt | grep " love "
Flowers love people
Aboslutely love this one. #suitsupply
```

Another example. Let's see the content of the output file with target's used hashtags.



```
┌──(jose㉿kali)-[~/Osintgram/output]
└─$ cat patrickboonstra_hashtags.txt
10. #flamingoinNam
3. #corona
3. #singledad
3. #YearOfFirsts
2. #sosuEE
2. #trulondon
2. #unleash
2. #hanoi
2. #truamsterdam
2. #roadtrip
2. #takecontrol
2. #ibiza❤
1. #youneverworkalone
1. #werkenbijrabobank
1. #werkenbijrabobank.
1. #staafenpaap
1. #RTE19
1. #Talinn
1. #gobigorgohome
1. #werkenbijPon
1. #boothstockfestival
1. #SosuEE
1. #weekendgetaway
1. #orangeHoutbay
1. #thisisAfrica
1. #Halloween
1. #Unleash
1. #unleash.
1. #sosuEU.
1. #nofilter
1. #deparade
1. #deparadedenhaag
1. #tortilla
1. #lunchofchampions
1. #boyslair
1. #studrally
1. #bloemenhoudenvanmensen
1. #flowers
1. #thesimplethingsthatmatter
1. #flamingoinghome:
1. #vietnam
1. #travelalone
1. #bountyisland
1. #flamingoinNam:
1. #bloom
1. #depersgroep
```

The screenshot doesn’t show the full view of the file which includes also lines with an empty hashtag. We want to exclude the lines with empty hashtags, sort it in the opposite order from 1 up to 10 and save to a new file.

This would look as follows:

cat patrickboonstra_hashtags.txt | grep [^#]$ | sort -n > patrick_hashtags_sorted.txt

Of course, I don't remember all of the switches to every Linux command. That's why there are manual pages to every command. Just type for instance *man sort*. You will find out that *sort* command supports saving the output file by the switch *-o* or *--output* so you can use it instead of redirect symbol >.

```
cat patrickboonstra_hashtags.txt | grep [^#]$ | grep [^1] | sort -n -o patrick_hashtags_sorted.txt
```

If you just want to go through larger files and you don't plan to edit them, you can use commands *more* and *less* followed by the file again. You can just press Enter to navigate through the file.

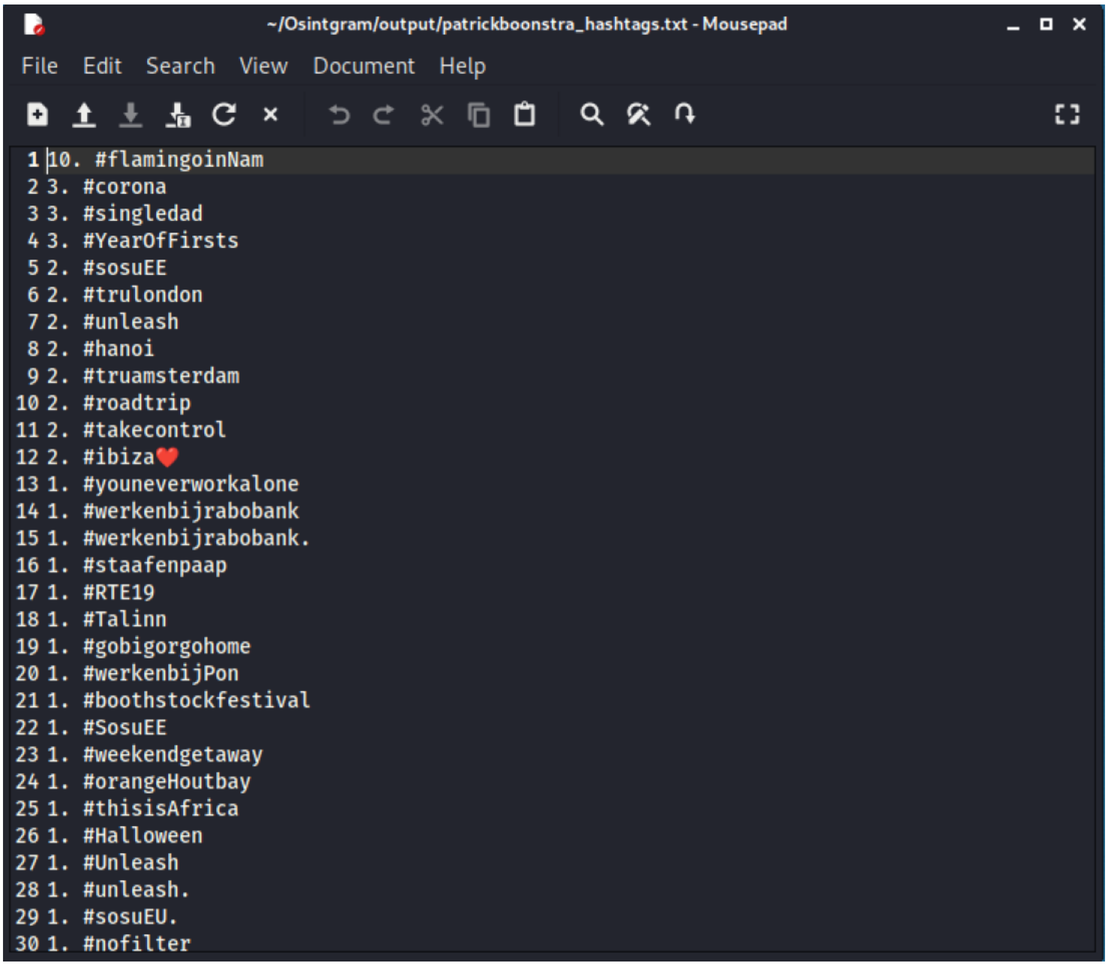
If you expect to edit the file, some text editor would be handy. There are several CLI (command-line interface) text editors such as ***nano*** or ***vim*** (or the older version **vi**) which if you got used to them, they are super-fast and efficient. They behave as standard Linux commands so just type:

```
vim filename
```

I prefer *vim* but you have to learn some commands which this app uses such as how to get from insert mode to command mode and vice versa. Once you get it into your muscle memory, it's unbeatable.

And finally, you can use some text editors with a graphic interface such as ***xedit*** or ***mousepad*** which is a standard application in Kali Linux. You can launch them in a same way as any other commands from the Linux shell. In this case I use:

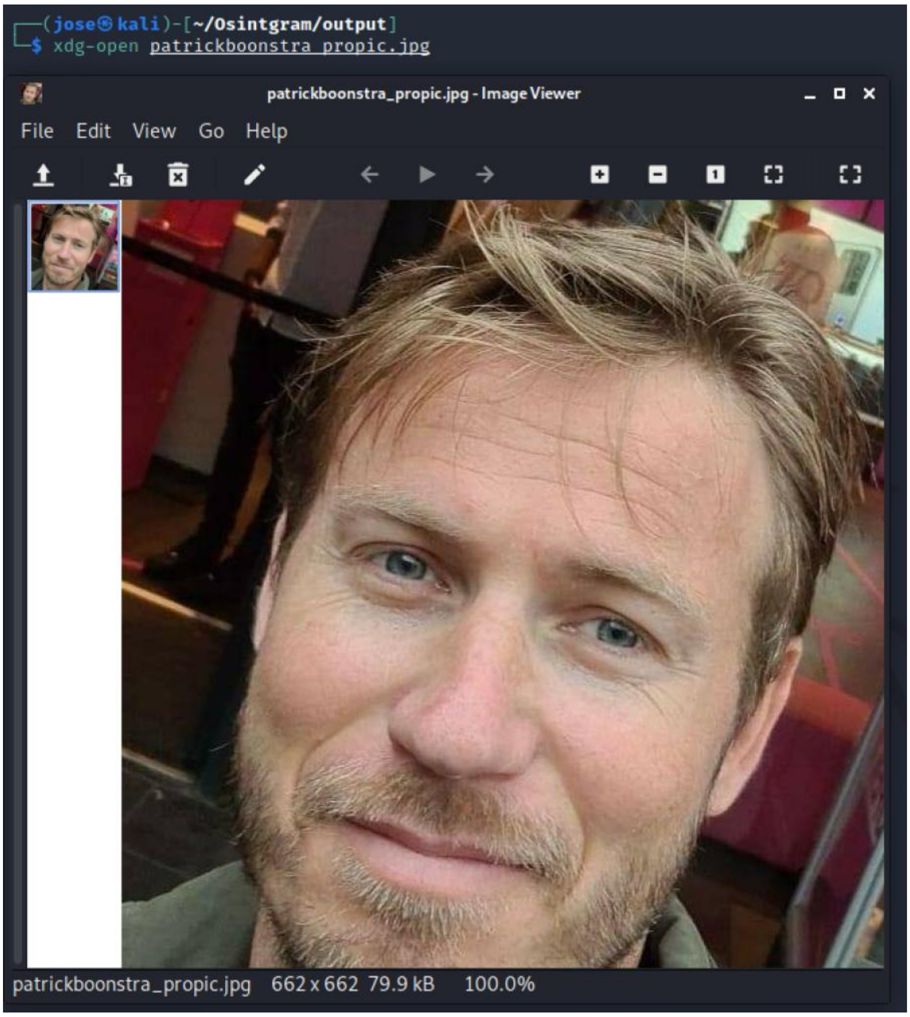
```
mousepad patrickboonstra_hashtags.txt
```



```
10. #flamingoinNam
3. #corona
3. #singledad
3. #YearOfFirsts
2. #sosuEE
2. #trulondon
2. #unleash
2. #hanoi
2. #truamsterdam
2. #roadtrip
2. #takecontrol
2. #ibiza❤
1. #youneverworkalone
1. #werkenbijrabobank
1. #werkenbijrabobank.
1. #staafenpaap
1. #RTE19
1. #Talinn
1. #gobigorgohome
1. #werkenbijPon
1. #boothstockfestival
1. #SosuEE
1. #weekendgetaway
1. #orangeHoutbay
1. #thisisAfrica
1. #Halloween
1. #Unleash
1. #unleash.
1. #sosuEU.
1. #nofilter
```

In the Osintgram output directory can be found not only text files but also pictures and video files (Instragram stories).

If you want to open a picture, just type the command ***xdg-open*** followed by the file name:



```
┌──(jose㉿kali)-[~/Osintgram/output]
└─$ xdg-open patrickboonstra_propic.jpg
```

patrickboonstra_propic.jpg - Image Viewer
File Edit View Go Help
patrickboonstra_propic.jpg 662 x 662 79.9 kB 100.0%

For the video files you can use mplayer, vlc or mpg123 which you have to install into your Kali Linux. If you want to do so, execute the following command:

sudo apt install mplayer

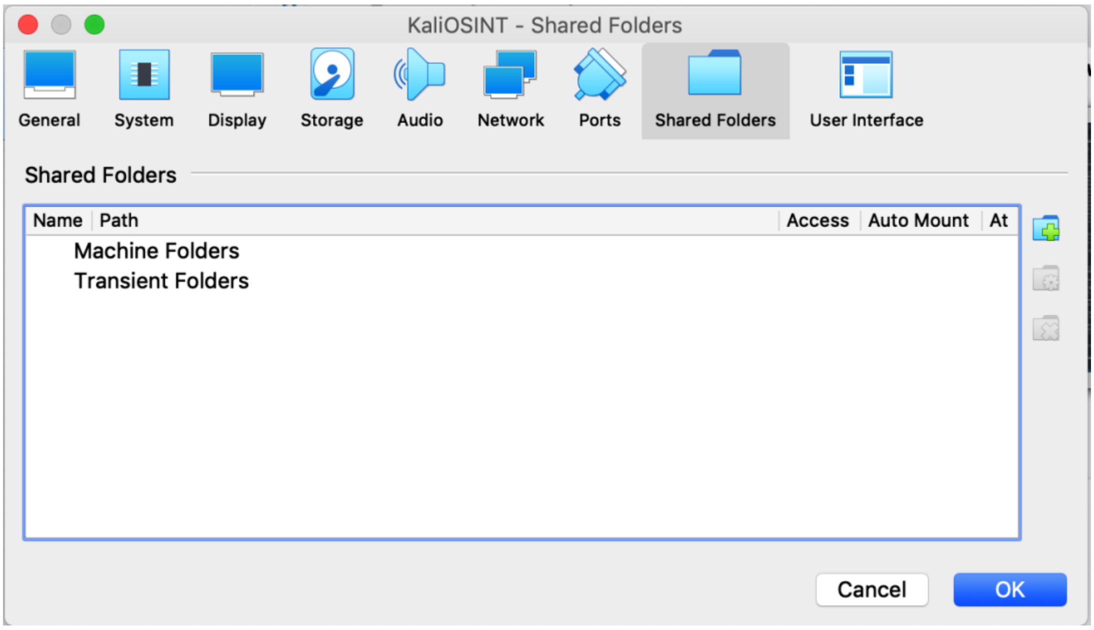
Enter your Kali Linux password and confirm *Y* to install. Then start the video with a command:

mplayer videofile.mp4

With this command I can play Patrick's Instagram stories.

Of course, there is another way to work with outputs. If you are more confident on your own hosting operating system (MacOS or MS Windows), you can set a shared directory by VirtualBox which will be visible by the hosting system and virtual machine as well.

You can do that by opening settings of the specific virtual machine on VirtualBox.



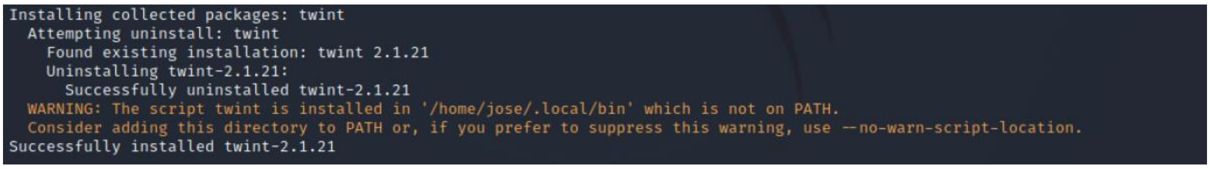
Name	Path	Access	Auto Mount	At
Machine Folders				
Transient Folders				

Twint

Firstly, let's install the app by the following commands:

git clone --depth==1 http://github.com/twintproject/twint.git
cd twint
pip3 install . -r requirements.txt

It might happen that you get this warning message during the installation.



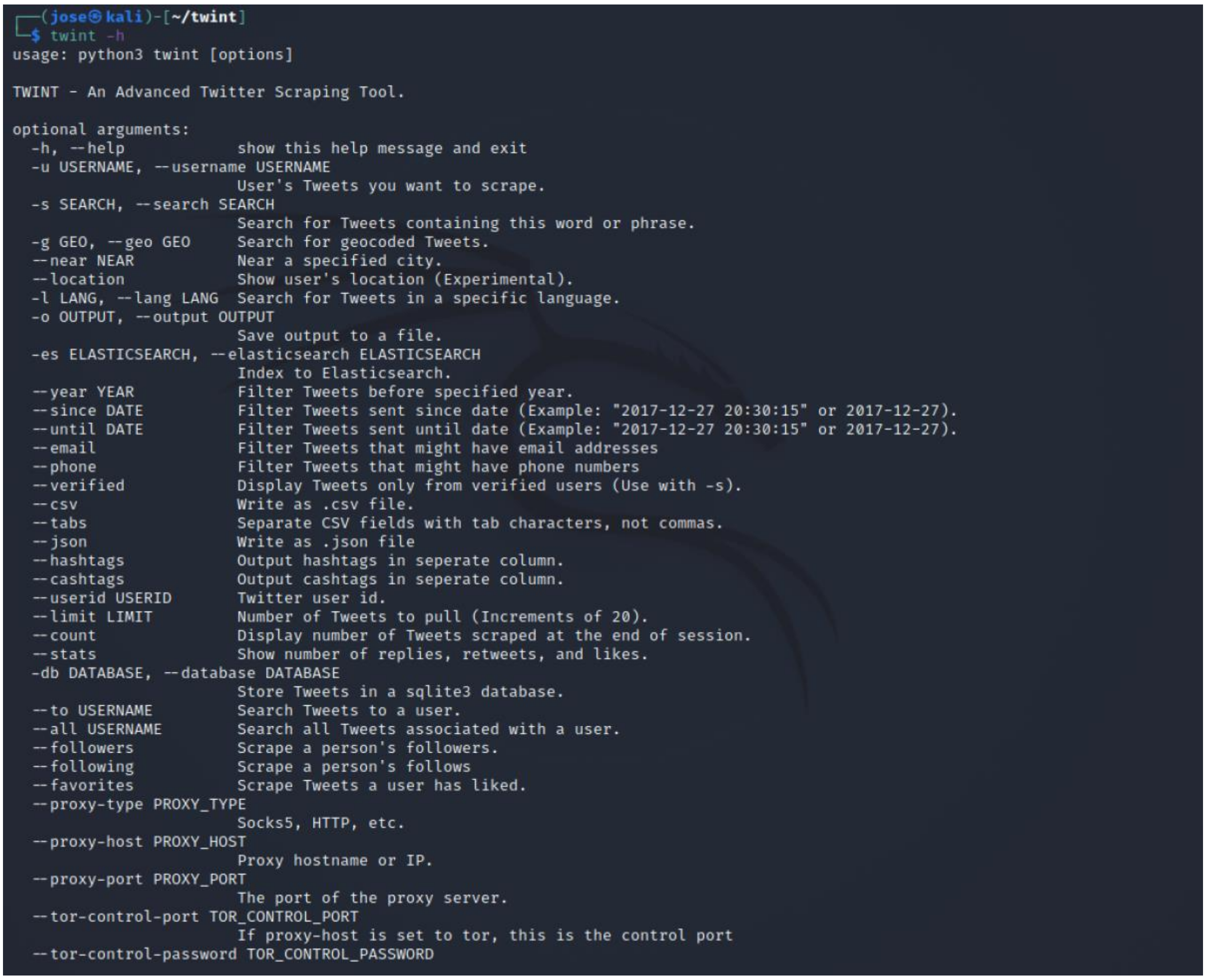
```
Installing collected packages: twint
  Attempting uninstall: twint
    Found existing installation: twint 2.1.21
    Uninstalling twint-2.1.21:
      Successfully uninstalled twint-2.1.21
  WARNING: The script twint is installed in '/home/jose/.local/bin' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed twint-2.1.21
```

It means that you cannot execute the command **twint** just by typing twint but you have to state the whole path to the twint file - */home/jose/.local/bin/twint* in my case. If you update the general Linux variable $PATH by adding the directory (as you can see in the following command), you can simply type *twint* only whenever you need to call this command.

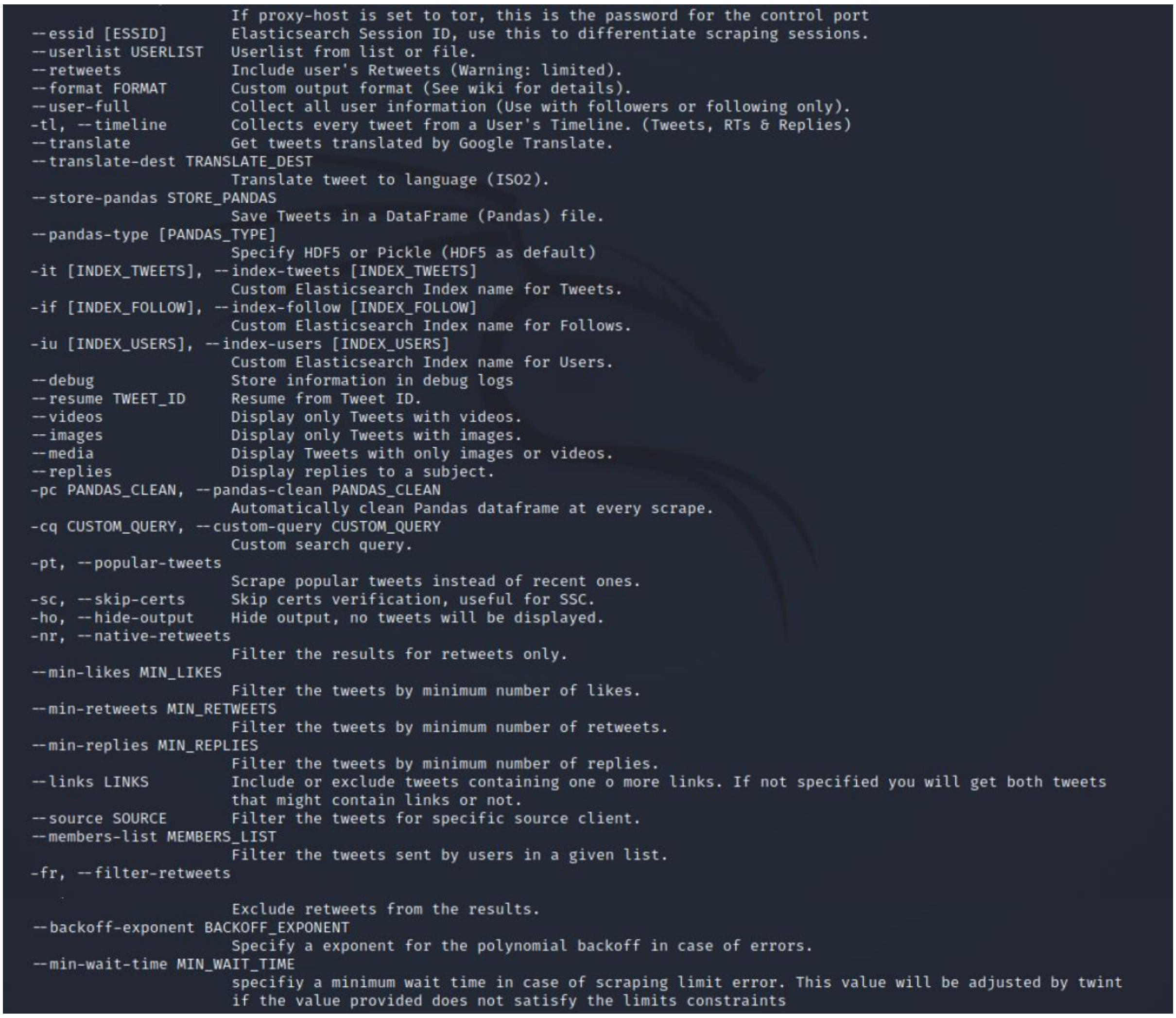
export PATH=$PATH:/home/jose/.local/bin/

Twint is the ultimate tool when researching data on Twitter. We can be analyzing tweets of a specific person, a group of people or start the analysis from some specific tweets related to a location for instance.

When we open the help page by a switch *-h* or *--help*, you get an idea about extensivity of this program.



```
┌──(jose㉿kali)-[~/twint]
└─$ twint -h
usage: python3 twint [options]

TWINT - An Advanced Twitter Scraping Tool.

optional arguments:
  -h, --help            show this help message and exit
  -u USERNAME, --username USERNAME
                        User's Tweets you want to scrape.
  -s SEARCH, --search SEARCH
                        Search for Tweets containing this word or phrase.
  -g GEO, --geo GEO     Search for geocoded Tweets.
  --near NEAR           Near a specified city.
  --location            Show user's location (Experimental).
  -l LANG, --lang LANG  Search for Tweets in a specific language.
  -o OUTPUT, --output OUTPUT
                        Save output to a file.
  -es ELASTICSEARCH, --elasticsearch ELASTICSEARCH
                        Index to Elasticsearch.
  --year YEAR           Filter Tweets before specified year.
  --since DATE          Filter Tweets sent since date (Example: "2017-12-27 20:30:15" or 2017-12-27).
  --until DATE          Filter Tweets sent until date (Example: "2017-12-27 20:30:15" or 2017-12-27).
  --email               Filter Tweets that might have email addresses
  --phone               Filter Tweets that might have phone numbers
  --verified            Display Tweets only from verified users (Use with -s).
  --csv                 Write as .csv file.
  --tabs                Separate CSV fields with tab characters, not commas.
  --json                Write as .json file
  --hashtags            Output hashtags in seperate column.
  --cashtags            Output cashtags in seperate column.
  --userid USERID       Twitter user id.
  --limit LIMIT         Number of Tweets to pull (Increments of 20).
  --count               Display number of Tweets scraped at the end of session.
  --stats               Show number of replies, retweets, and likes.
  -db DATABASE, --database DATABASE
                        Store Tweets in a sqlite3 database.
  --to USERNAME         Search Tweets to a user.
  --all USERNAME        Search all Tweets associated with a user.
  --followers           Scrape a person's followers.
  --following           Scrape a person's follows
  --favorites           Scrape Tweets a user has liked.
  --proxy-type PROXY_TYPE
                        Socks5, HTTP, etc.
  --proxy-host PROXY_HOST
                        Proxy hostname or IP.
  --proxy-port PROXY_PORT
                        The port of the proxy server.
  --tor-control-port TOR_CONTROL_PORT
                        If proxy-host is set to tor, this is the control port
  --tor-control-password TOR_CONTROL_PASSWORD
```



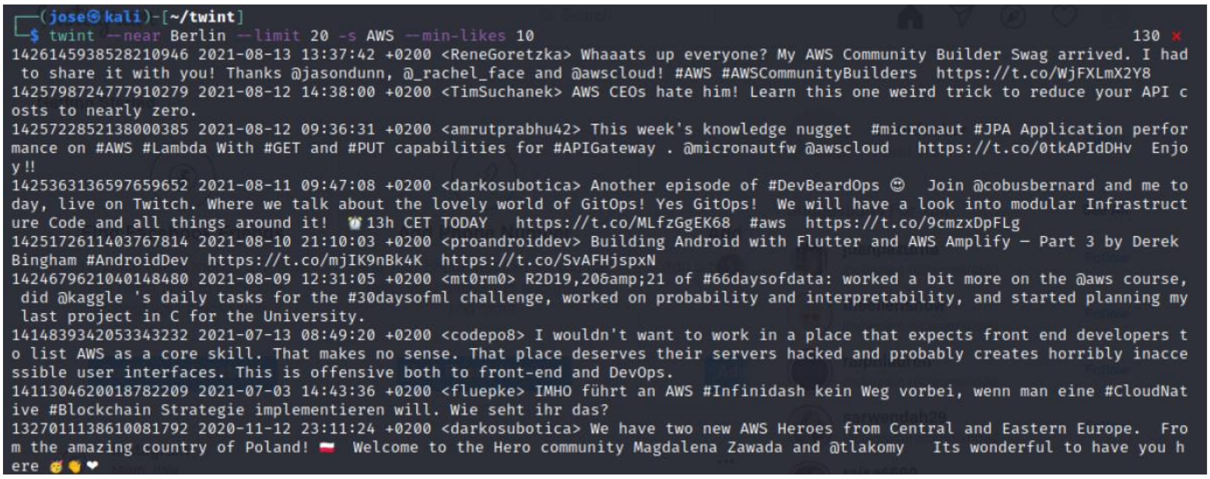
```
                        If proxy-host is set to tor, this is the password for the control port
  --essid [ESSID]       Elasticsearch Session ID, use this to differentiate scraping sessions.
  --userlist USERLIST   Userlist from list or file.
  --retweets            Include user's Retweets (Warning: limited).
  --format FORMAT       Custom output format (See wiki for details).
  --user-full           Collect all user information (Use with followers or following only).
  -tl, --timeline       Collects every tweet from a User's Timeline. (Tweets, RTs & Replies)
  --translate           Get tweets translated by Google Translate.
  --translate-dest TRANSLATE_DEST
                        Translate tweet to language (ISO2).
  --store-pandas STORE_PANDAS
                        Save Tweets in a DataFrame (Pandas) file.
  --pandas-type [PANDAS_TYPE]
                        Specify HDF5 or Pickle (HDF5 as default)
  -it [INDEX_TWEETS], --index-tweets [INDEX_TWEETS]
                        Custom Elasticsearch Index name for Tweets.
  -if [INDEX_FOLLOW], --index-follow [INDEX_FOLLOW]
                        Custom Elasticsearch Index name for Follows.
  -iu [INDEX_USERS], --index-users [INDEX_USERS]
                        Custom Elasticsearch Index name for Users.
  --debug               Store information in debug logs
  --resume TWEET_ID     Resume from Tweet ID.
  --videos              Display only Tweets with videos.
  --images              Display only Tweets with images.
  --media               Display Tweets with only images or videos.
  --replies             Display replies to a subject.
  -pc PANDAS_CLEAN, --pandas-clean PANDAS_CLEAN
                        Automatically clean Pandas dataframe at every scrape.
  -cq CUSTOM_QUERY, --custom-query CUSTOM_QUERY
                        Custom search query.
  -pt, --popular-tweets
                        Scrape popular tweets instead of recent ones.
  -sc, --skip-certs     Skip certs verification, useful for SSC.
  -ho, --hide-output    Hide output, no tweets will be displayed.
  -nr, --native-retweets
                        Filter the results for retweets only.
  --min-likes MIN_LIKES
                        Filter the tweets by minimum number of likes.
  --min-retweets MIN_RETWEETS
                        Filter the tweets by minimum number of retweets.
  --min-replies MIN_REPLIES
                        Filter the tweets by minimum number of replies.
  --links LINKS         Include or exclude tweets containing one o more links. If not specified you will get both tweets
                        that might contain links or not.
  --source SOURCE       Filter the tweets for specific source client.
  --members-list MEMBERS_LIST
                        Filter the tweets sent by users in a given list.
  -fr, --filter-retweets

                        Exclude retweets from the results.
  --backoff-exponent BACKOFF_EXPONENT
                        Specify a exponent for the polynomial backoff in case of errors.
  --min-wait-time MIN_WAIT_TIME
                        specifiy a minimum wait time in case of scraping limit error. This value will be adjusted by twint
                        if the value provided does not satisfy the limits constraints
```

The advantage of using *twint* from *Osintgram* is that you don’t need any burner account as Twitter data are pretty much open for anybody.

So, we have no obstacle to start right away.

Let’s say we want to see tweets from Berlin containing the keyword AWS (the cloud service Amazon AWS) and having 10 likes at least. We can do that this way:

twint --near Berlin --limit 20 -s AWS --min-likes 10



```
┌──(jose㉿kali)-[~/twint]
└─$ twint --near Berlin --limit 20 -s AWS --min-likes 10                                                          130 ✗
1426145938528210946 2021-08-13 13:37:42 +0200 <ReneGoretzka> Whaaats up everyone? My AWS Community Builder Swag arrived. I had
 to share it with you! Thanks @jasondunn, @_rachel_face and @awscloud! #AWS #AWSCommunityBuilders  https://t.co/WjFXLmX2Y8
1425798724777910279 2021-08-12 14:38:00 +0200 <TimSuchanek> AWS CEOs hate him! Learn this one weird trick to reduce your API c
osts to nearly zero.
1425722852138000385 2021-08-12 09:36:31 +0200 <amrutprabhu42> This week's knowledge nugget  #micronaut #JPA Application perfor
mance on #AWS #Lambda With #GET and #PUT capabilities for #APIGateway . @micronautfw @awscloud   https://t.co/0tkAPIdDHv  Enjo
y !!
1425363136597659652 2021-08-11 09:47:08 +0200 <darkosubotica> Another episode of #DevBeardOps 😍  Join @cobusbernard and me to
day, live on Twitch. Where we talk about the lovely world of GitOps! Yes GitOps!  We will have a look into modular Infrastruct
ure Code and all things around it!  ⏰13h CET TODAY   https://t.co/MLfzGgEK68  #aws  https://t.co/9cmzxDpFLg
1425172611403767814 2021-08-10 21:10:03 +0200 <proandroiddev> Building Android with Flutter and AWS Amplify — Part 3 by Derek
Bingham #AndroidDev  https://t.co/mjIK9nBk4K  https://t.co/SvAFHjspxN
1424679621040148480 2021-08-09 12:31:05 +0200 <mt0rm0> R2D19,20&amp;21 of #66daysofdata: worked a bit more on the @aws course,
 did @kaggle 's daily tasks for the #30daysofml challenge, worked on probability and interpretability, and started planning my
 last project in C for the University.
1414839342053343232 2021-07-13 08:49:20 +0200 <codepo8> I wouldn't want to work in a place that expects front end developers t
o list AWS as a core skill. That makes no sense. That place deserves their servers hacked and probably creates horribly inacce
ssible user interfaces. This is offensive both to front-end and DevOps.
1411304620018782209 2021-07-03 14:43:36 +0200 <fluepke> IMHO führt an AWS #Infinidash kein Weg vorbei, wenn man eine #CloudNat
ive #Blockchain Strategie implementieren will. Wie seht ihr das?
1327011138610081792 2020-11-12 23:11:24 +0200 <darkosubotica> We have two new AWS Heroes from Central and Eastern Europe.  Fro
m the amazing country of Poland! 🇵🇱  Welcome to the Hero community Magdalena Zawada and @tlakomy   Its wonderful to have you h
ere 🥳👏❤
```

I limit the output to 20 tweests otherwise it would be going constantly unless you stop it. I will show you how to save this efficiently into a file in various formats later on.

Another way to specify location of searched tweets is by geographic (geo or GPS) coordinates. Let’s Google for Berlin coordinates and set 30 km radius.

twint -g "52.531677,13.381777,30km" --limit 20 -s AWS --min-likes 10

Now let’s focus the research on a specific person or group of people. Let’s display tweets of my precious guinea pig Patrick Boostra but only those from 2014.

twint -u PatrickBoonstra --year 2014

We can also be more specific and filter all Patrick’s tweets from the mid of 2013 to the mid of 2014.

twint -u PatrickBoonstra --since 2013-07-01 --until 2014-06-30

You can adjust many other parameters including:

--videos
--images
--media

to display tweets with videos, images or either of them.

You can filter tweets to a specific user:

--to USERNAME

Or tweets associated with a specific user:

--all USERNAME

You can exclude retweets with a command:

```
-fr
```

Or include retweets:

```
--retweets
```

If you want to display tweets by the verified user's only, it is also possible:

```
--verified
```

You can do the research for more than one target at the same time. Just create a text file with a single Twitter usernames per line and use the parameter:

```
--userlist target_list.txt
```

Usually, you will need to store the outputs rather than reading them from the terminal shell. There are several ways how to do that.

You can simply let twint to write output into a text file:

```
-o output_file.txt
```

For this task we wouldn't even need a specific parameter as you can use Linux output redirecting so the following command would do the same:

```
twint -u PatrickBoonstra --since 2013-07-01 --until 2014-06-30 >> patrickboonstra_output_tweets.txt
```

You can also create these files in the CSV format:

```
-o output_file.csv --csv
```

Or JSON:

```
-o output_file.csv --json
```

There is also a possibility to save the output to sqlite3 database, send it to Elastic search or to Pandas which you can use for e.g. analyzing tweets with NLP.

If you need to cover your digital identity from you do the OSINT research from, you can use a proxy or send your request through TOR network as well.

The beauty of the most of the CLI apps (Python3 scripts in this case) is that you can easily incorporate them into your shell scripts or your Python3 scripts for example by importing the function:

```
import twint

c = twint.Config()
c.Username = "noneprivacy"
c.Limit = 100
c.Store_csv = True
c.Output = "none.csv"
c.Lang = "en"
c.Translate = True
c.TranslateDest = "it"
twint.run.Search(c)
```

Sherlock

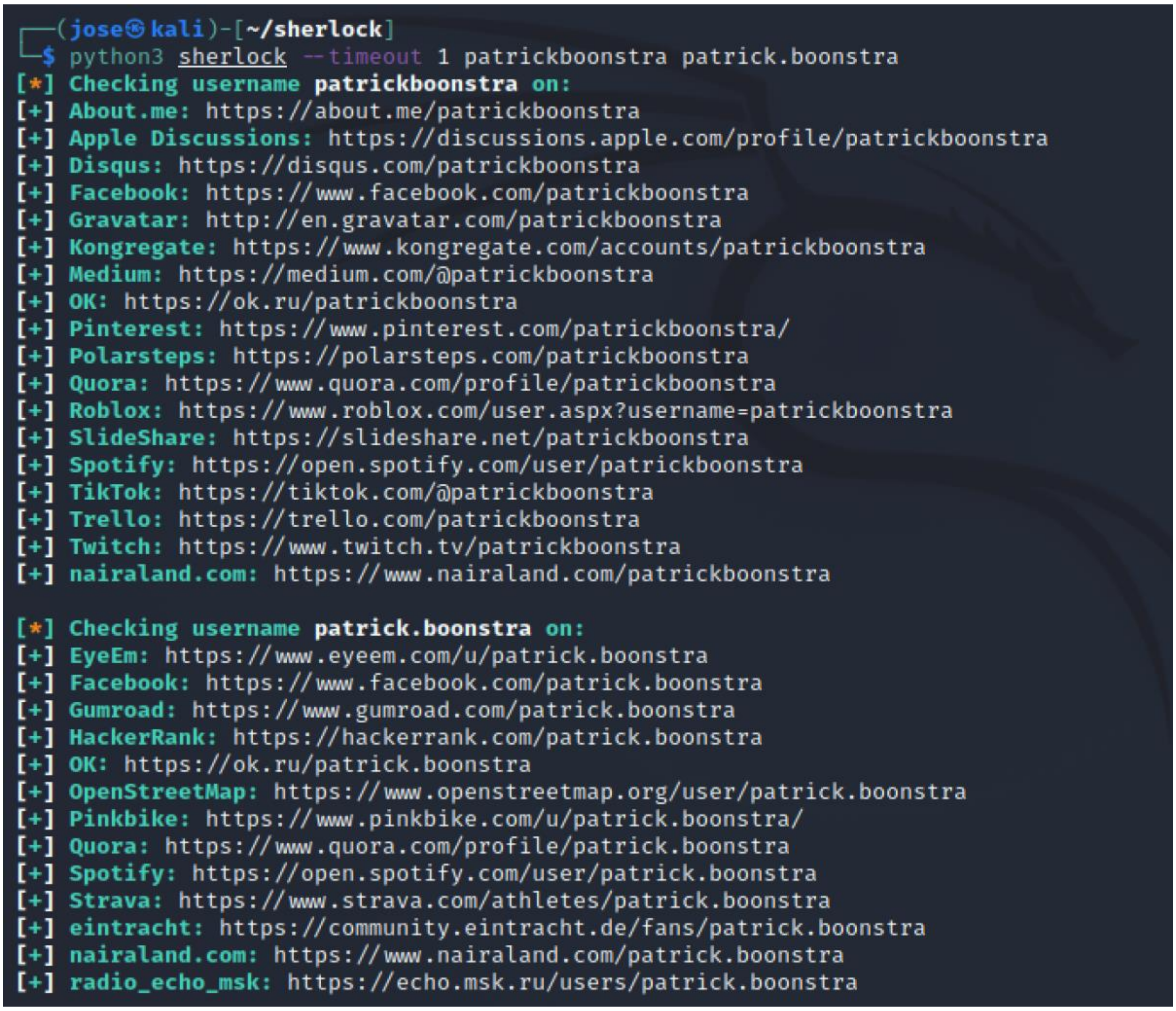
Another neat OSINT tool for information gathering is Sherlock. You might know cross-referencing web tools such as namechk.com or namecheckup.com. Sherlock is something like that on the command line.

There is a standard installation:

```
git clone https://github.com/sherlock-project/sherlock.git
cd sherlock
python3 -m pip install -r requirements.txt
```

Let’s find out the profiles of our guinea pig Patrick Boonstra. We know Patrick’s username *patrickboonstra* from Instagram. So let’s start from there and also expand it for the version *patrick.boonstra*.

```
python3 sherlock --timeout 1 patrickboonstra patrick.boonstra
```

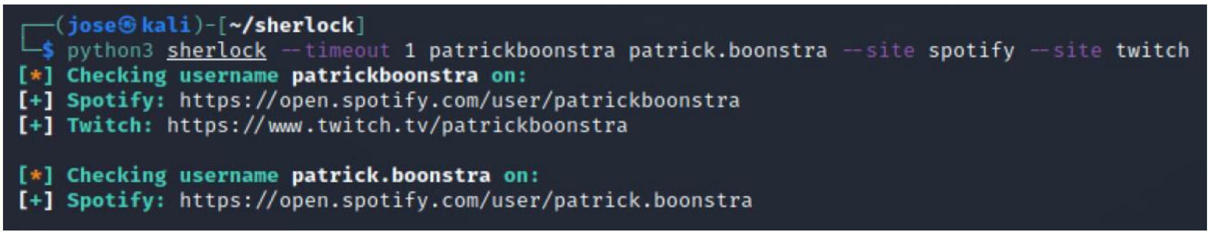


```
┌──(jose㉿kali)-[~/sherlock]
└─$ python3 sherlock --timeout 1 patrickboonstra patrick.boonstra
[*] Checking username patrickboonstra on:
[+] About.me: https://about.me/patrickboonstra
[+] Apple Discussions: https://discussions.apple.com/profile/patrickboonstra
[+] Disqus: https://disqus.com/patrickboonstra
[+] Facebook: https://www.facebook.com/patrickboonstra
[+] Gravatar: http://en.gravatar.com/patrickboonstra
[+] Kongregate: https://www.kongregate.com/accounts/patrickboonstra
[+] Medium: https://medium.com/@patrickboonstra
[+] OK: https://ok.ru/patrickboonstra
[+] Pinterest: https://www.pinterest.com/patrickboonstra/
[+] Polarsteps: https://polarsteps.com/patrickboonstra
[+] Quora: https://www.quora.com/profile/patrickboonstra
[+] Roblox: https://www.roblox.com/user.aspx?username=patrickboonstra
[+] SlideShare: https://slideshare.net/patrickboonstra
[+] Spotify: https://open.spotify.com/user/patrickboonstra
[+] TikTok: https://tiktok.com/@patrickboonstra
[+] Trello: https://trello.com/patrickboonstra
[+] Twitch: https://www.twitch.tv/patrickboonstra
[+] nairaland.com: https://www.nairaland.com/patrickboonstra

[*] Checking username patrick.boonstra on:
[+] EyeEm: https://www.eyeem.com/u/patrick.boonstra
[+] Facebook: https://www.facebook.com/patrick.boonstra
[+] Gumroad: https://www.gumroad.com/patrick.boonstra
[+] HackerRank: https://hackerrank.com/patrick.boonstra
[+] OK: https://ok.ru/patrick.boonstra
[+] OpenStreetMap: https://www.openstreetmap.org/user/patrick.boonstra
[+] Pinkbike: https://www.pinkbike.com/u/patrick.boonstra/
[+] Quora: https://www.quora.com/profile/patrick.boonstra
[+] Spotify: https://open.spotify.com/user/patrick.boonstra
[+] Strava: https://www.strava.com/athletes/patrick.boonstra
[+] eintracht: https://community.eintracht.de/fans/patrick.boonstra
[+] nairaland.com: https://www.nairaland.com/patrick.boonstra
[+] radio_echo_msk: https://echo.msk.ru/users/patrick.boonstra
```

This is a starting point as not all of these profiles are actually our Patrick's profiles. Some of them are also bogus not existing profiles.

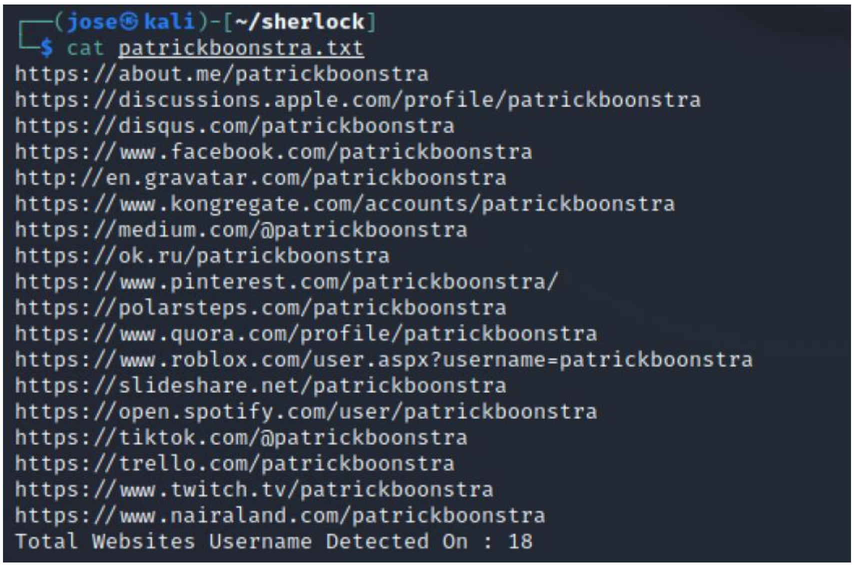
If you want to check the existence of target profiles on specific sites, use the operator *--site*. Let's check if Patrick the guinea pig has profiles on Spotify and Twitch.

python3 sherlock --timeout 1 patrickboonstra patrick.boonstra --site spotify --site twitch



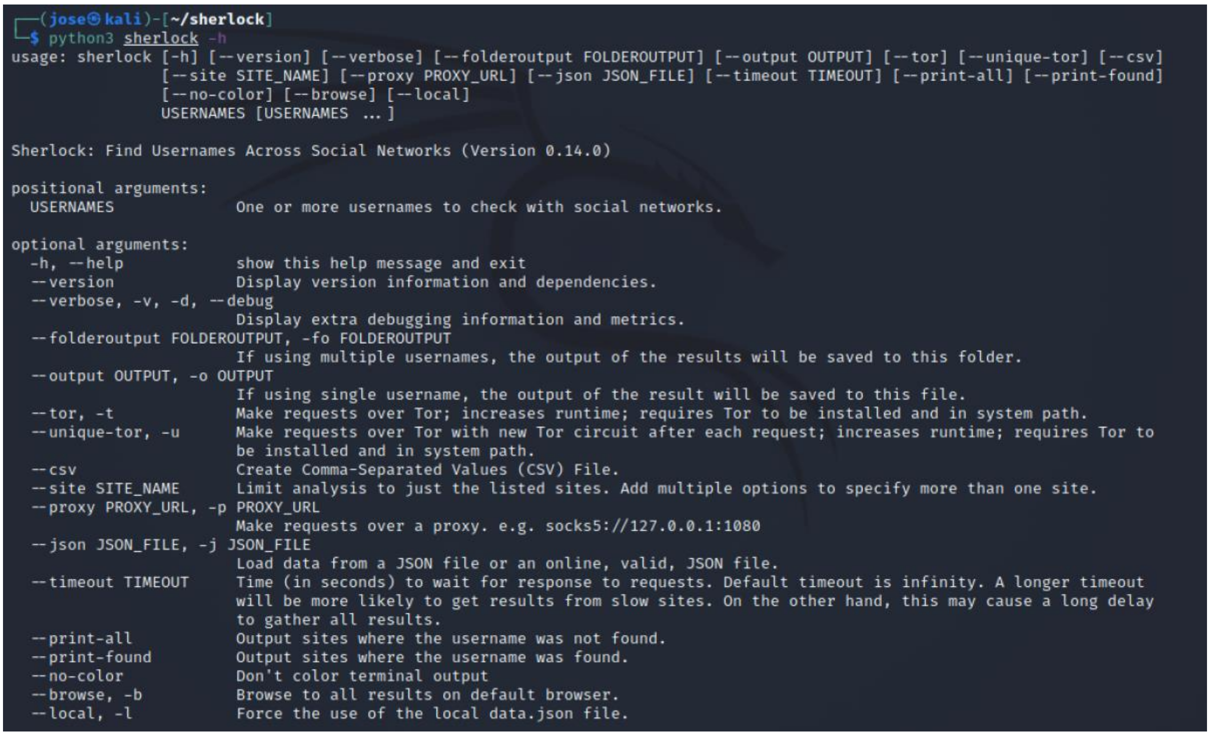
```
┌──(jose㉿kali)-[~/sherlock]
└─$ python3 sherlock --timeout 1 patrickboonstra patrick.boonstra --site spotify --site twitch
[*] Checking username patrickboonstra on:
[+] Spotify: https://open.spotify.com/user/patrickboonstra
[+] Twitch: https://www.twitch.tv/patrickboonstra

[*] Checking username patrick.boonstra on:
[+] Spotify: https://open.spotify.com/user/patrick.boonstra
```

The output is automatically saved to the text files named after the searched usernames.



```
┌──(jose㉿kali)-[~/sherlock]
└─$ cat patrickboonstra.txt
https://about.me/patrickboonstra
https://discussions.apple.com/profile/patrickboonstra
https://disqus.com/patrickboonstra
https://www.facebook.com/patrickboonstra
http://en.gravatar.com/patrickboonstra
https://www.kongregate.com/accounts/patrickboonstra
https://medium.com/@patrickboonstra
https://ok.ru/patrickboonstra
https://www.pinterest.com/patrickboonstra/
https://polarsteps.com/patrickboonstra
https://www.quora.com/profile/patrickboonstra
https://www.roblox.com/user.aspx?username=patrickboonstra
https://slideshare.net/patrickboonstra
https://open.spotify.com/user/patrickboonstra
https://tiktok.com/@patrickboonstra
https://trello.com/patrickboonstra
https://www.twitch.tv/patrickboonstra
https://www.nairaland.com/patrickboonstra
Total Websites Username Detected On : 18
```

You can change the output file by the option *-o* or save it to a specific folder by the option *-fo*. There is also a possibility to store the output as CSV or JSON. Using proxy or TOR network for better anonymity is also an option. Use the switch *-h* to see all the options.



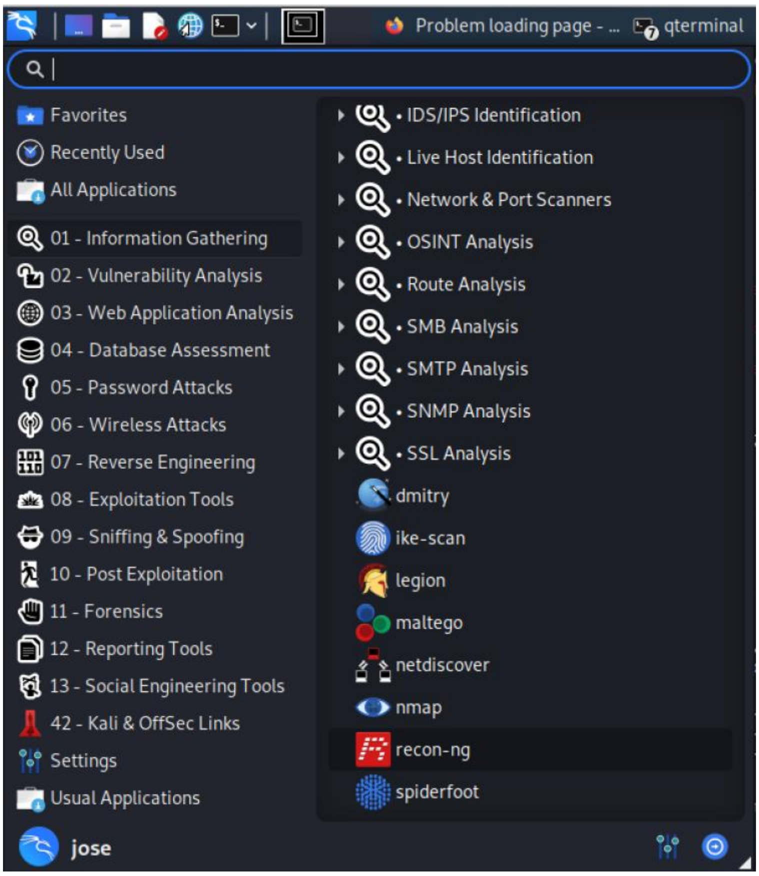
```
┌──(jose㉿kali)-[~/sherlock]
└─$ python3 sherlock -h
usage: sherlock [-h] [--version] [--verbose] [--folderoutput FOLDEROUTPUT] [--output OUTPUT] [--tor] [--unique-tor] [--csv]
                [--site SITE_NAME] [--proxy PROXY_URL] [--json JSON_FILE] [--timeout TIMEOUT] [--print-all] [--print-found]
                [--no-color] [--browse] [--local]
                USERNAMES [USERNAMES ...]

Sherlock: Find Usernames Across Social Networks (Version 0.14.0)

positional arguments:
  USERNAMES             One or more usernames to check with social networks.

optional arguments:
  -h, --help            show this help message and exit
  --version             Display version information and dependencies.
  --verbose, -v, -d, --debug
                        Display extra debugging information and metrics.
  --folderoutput FOLDEROUTPUT, -fo FOLDEROUTPUT
                        If using multiple usernames, the output of the results will be saved to this folder.
  --output OUTPUT, -o OUTPUT
                        If using single username, the output of the result will be saved to this file.
  --tor, -t             Make requests over Tor; increases runtime; requires Tor to be installed and in system path.
  --unique-tor, -u      Make requests over Tor with new Tor circuit after each request; increases runtime; requires Tor to
                        be installed and in system path.
  --csv                 Create Comma-Separated Values (CSV) File.
  --site SITE_NAME      Limit analysis to just the listed sites. Add multiple options to specify more than one site.
  --proxy PROXY_URL, -p PROXY_URL
                        Make requests over a proxy. e.g. socks5://127.0.0.1:1080
  --json JSON_FILE, -j JSON_FILE
                        Load data from a JSON file or an online, valid, JSON file.
  --timeout TIMEOUT     Time (in seconds) to wait for response to requests. Default timeout is infinity. A longer timeout
                        will be more likely to get results from slow sites. On the other hand, this may cause a long delay
                        to gather all results.
  --print-all           Output sites where the username was not found.
  --print-found         Output sites where the username was found.
  --no-color            Don't color terminal output
  --browse, -b          Browse to all results on default browser.
  --local, -l           Force the use of the local data.json file.
```

There is another lightweight tool of this sort called Skiptracer which can also find information about the US car plates. If you want something more robust, see *recon-ng* as the next showcased example.

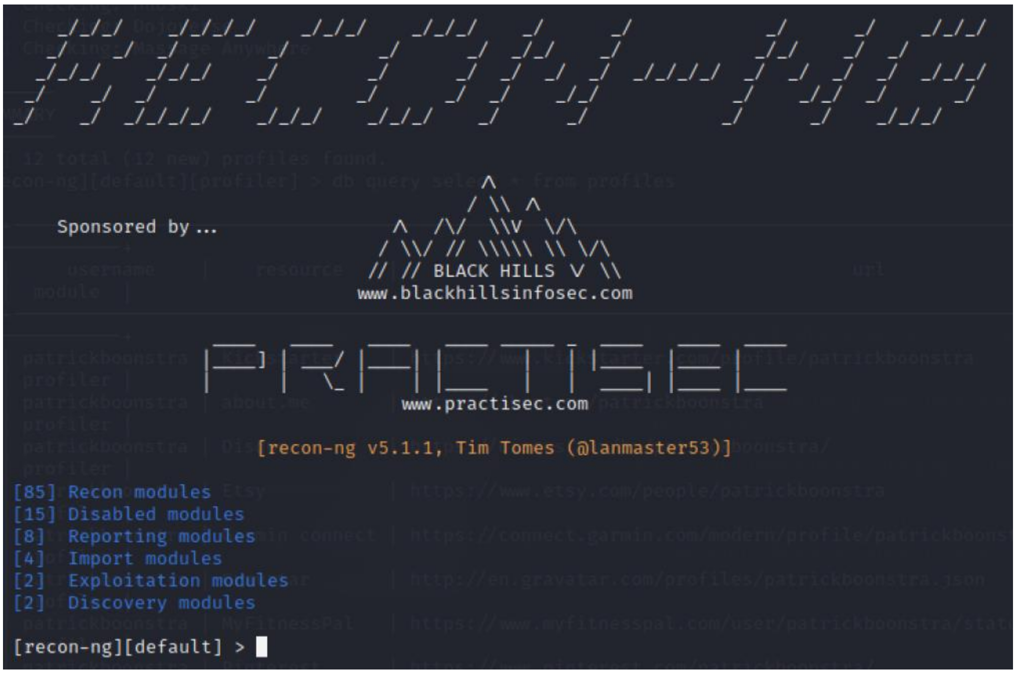
Recon-ng

This is a big one. Not a coincidence that Recon-ng is in Kali Linux by default. More than just an app it is the whole framework where you install other apps (they are called modules there) like Sherlock. So just one specific *recon-ng* module called *profiler* can do more than the whole *Sherlock*. And you have tens of those and you or the community can create others.

So, let's dive in. It's already installed on Kali Linux so type recon-ng in the terminal or start it using an icon in the graphical menu:



Once you start it, you probably get a ton of error messages about missing API keys for various services but that's OK.



```
    Sponsored by ...
                                    BLACK HILLS
                            www.blackhillsinfosec.com

                                 www.practisec.com

                    [recon-ng v5.1.1, Tim Tomes (@lanmaster53)]

[85] Recon modules
[15] Disabled modules
[8]  Reporting modules
[4]  Import modules
[2]  Exploitation modules
[2]  Discovery modules

[recon-ng][default] >
```

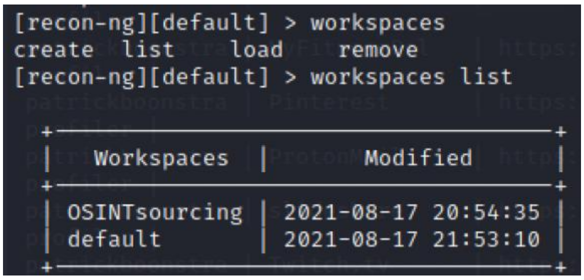
Now we are in the prompt of recon-ng. There are three layers to navigate through.

Workspaces -> Modules -> Database tables

The prompt on the screenshot is in the default workspace which is ok. We can create another one but why? Imagine that as the workspaces on your MacOS – you can have different apps throughout the different workspaces. It effects the database as well – one workspace might contain recon data from one company or a group of people and another project from a different one.

It will get a little while to get to used to that. What helps is that when you press Tab on your keyboard, it completes the command and also suggests you what are your options when you hit it again after you make a space after the command.

You can try to hit the command *workspaces* followed by a space and then hit Tab. After that I choose *list*.



```
[recon-ng][default] > workspaces
create  list  load  remove
[recon-ng][default] > workspaces list

  +---------------------------------------+
  |  Workspaces  |      Modified          |
  +---------------------------------------+
  | OSINTsourcing | 2021-08-17 20:54:35 |
  | default       | 2021-08-17 21:53:10 |
  +---------------------------------------+
```

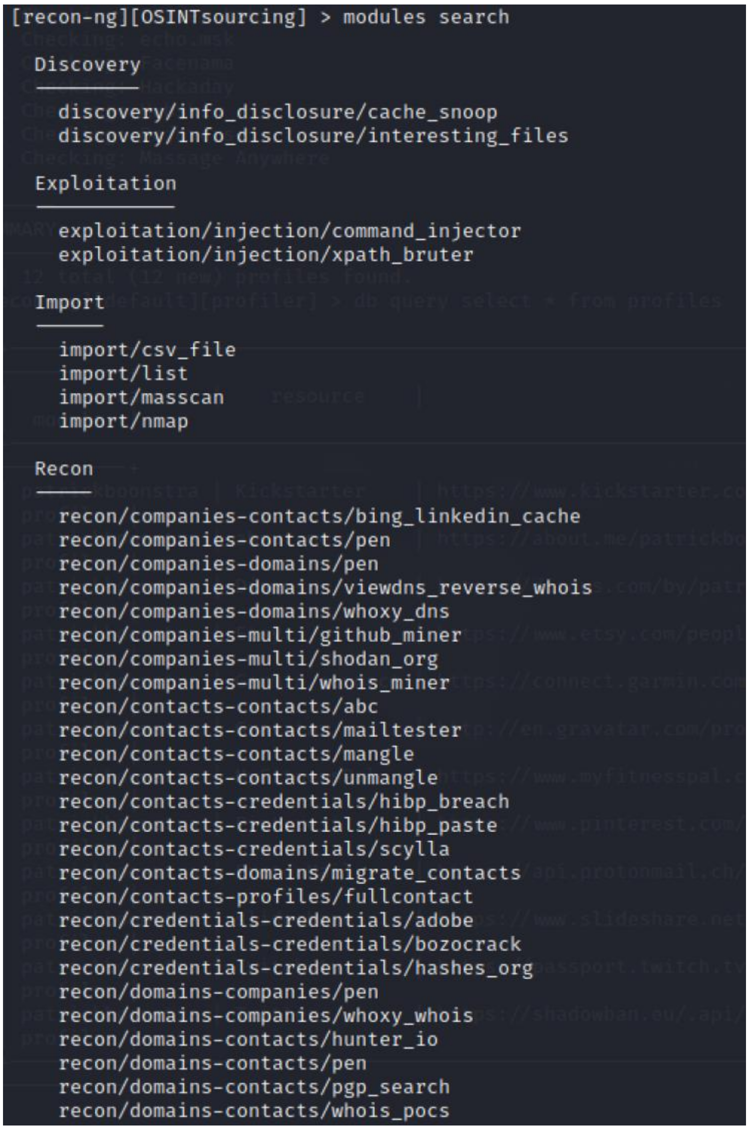
You can see that I've also created the workspace *OSINTsourcing* next to the default one.

Let's hit the following command to switch into the OSINTsourcing workspace:

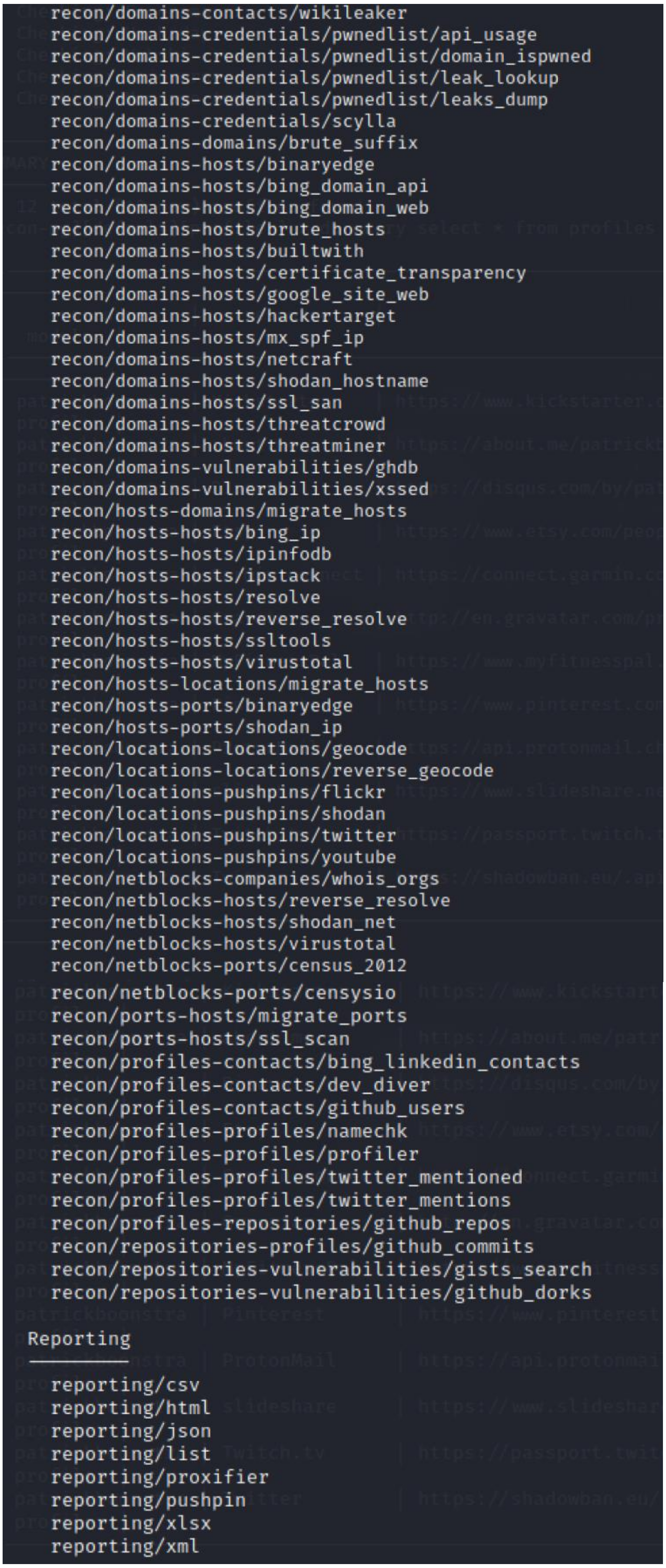
workspaces load OSINTsourcing

The command prompt has changed to *OSINTsourcing*.

Now let’s list the modules we can install. Hit the command *modules search*:



```
[recon-ng][OSINTsourcing] > modules search

  Discovery
  ---------
    discovery/info_disclosure/cache_snoop
    discovery/info_disclosure/interesting_files

  Exploitation
  ------------
    exploitation/injection/command_injector
    exploitation/injection/xpath_bruter

  Import
  ------
    import/csv_file
    import/list
    import/masscan
    import/nmap

  Recon
  -----
    recon/companies-contacts/bing_linkedin_cache
    recon/companies-contacts/pen
    recon/companies-domains/pen
    recon/companies-domains/viewdns_reverse_whois
    recon/companies-domains/whoxy_dns
    recon/companies-multi/github_miner
    recon/companies-multi/shodan_org
    recon/companies-multi/whois_miner
    recon/contacts-contacts/abc
    recon/contacts-contacts/mailtester
    recon/contacts-contacts/mangle
    recon/contacts-contacts/unmangle
    recon/contacts-credentials/hibp_breach
    recon/contacts-credentials/hibp_paste
    recon/contacts-credentials/scylla
    recon/contacts-domains/migrate_contacts
    recon/contacts-profiles/fullcontact
    recon/credentials-credentials/adobe
    recon/credentials-credentials/bozocrack
    recon/credentials-credentials/hashes_org
    recon/domains-companies/pen
    recon/domains-companies/whoxy_whois
    recon/domains-contacts/hunter_io
    recon/domains-contacts/pen
    recon/domains-contacts/pgp_search
    recon/domains-contacts/whois_pocs
```



```
    recon/domains-contacts/wikileaker
    recon/domains-credentials/pwnedlist/api_usage
    recon/domains-credentials/pwnedlist/domain_ispwned
    recon/domains-credentials/pwnedlist/leak_lookup
    recon/domains-credentials/pwnedlist/leaks_dump
    recon/domains-credentials/scylla
    recon/domains-domains/brute_suffix
    recon/domains-hosts/binaryedge
    recon/domains-hosts/bing_domain_api
    recon/domains-hosts/bing_domain_web
    recon/domains-hosts/brute_hosts
    recon/domains-hosts/builtwith
    recon/domains-hosts/certificate_transparency
    recon/domains-hosts/google_site_web
    recon/domains-hosts/hackertarget
    recon/domains-hosts/mx_spf_ip
    recon/domains-hosts/netcraft
    recon/domains-hosts/shodan_hostname
    recon/domains-hosts/ssl_san
    recon/domains-hosts/threatcrowd
    recon/domains-hosts/threatminer
    recon/domains-vulnerabilities/ghdb
    recon/domains-vulnerabilities/xssed
    recon/hosts-domains/migrate_hosts
    recon/hosts-hosts/bing_ip
    recon/hosts-hosts/ipinfodb
    recon/hosts-hosts/ipstack
    recon/hosts-hosts/resolve
    recon/hosts-hosts/reverse_resolve
    recon/hosts-hosts/ssltools
    recon/hosts-hosts/virustotal
    recon/hosts-locations/migrate_hosts
    recon/hosts-ports/binaryedge
    recon/hosts-ports/shodan_ip
    recon/locations-locations/geocode
    recon/locations-locations/reverse_geocode
    recon/locations-pushpins/flickr
    recon/locations-pushpins/shodan
    recon/locations-pushpins/twitter
    recon/locations-pushpins/youtube
    recon/netblocks-companies/whois_orgs
    recon/netblocks-hosts/reverse_resolve
    recon/netblocks-hosts/shodan_net
    recon/netblocks-hosts/virustotal
    recon/netblocks-ports/census_2012
    recon/netblocks-ports/censysio
    recon/ports-hosts/migrate_ports
    recon/ports-hosts/ssl_scan
    recon/profiles-contacts/bing_linkedin_contacts
    recon/profiles-contacts/dev_diver
    recon/profiles-contacts/github_users
    recon/profiles-profiles/namechk
    recon/profiles-profiles/profiler
    recon/profiles-profiles/twitter_mentioned
    recon/profiles-profiles/twitter_mentions
    recon/profiles-repositories/github_repos
    recon/repositories-profiles/github_commits
    recon/repositories-vulnerabilities/gists_search
    recon/repositories-vulnerabilities/github_dorks

  Reporting
  ---------
    reporting/csv
    reporting/html
    reporting/json
    reporting/list
    reporting/proxifier
    reporting/pushpin
    reporting/xlsx
    reporting/xml
```

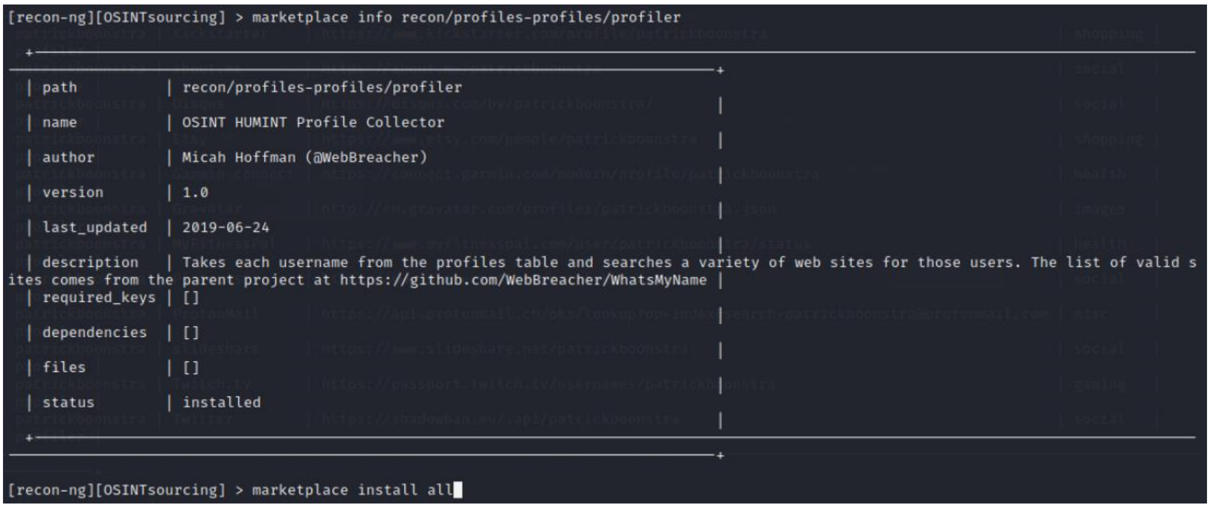
You can notice that you can use various known sources for the research such as Google, Bing, Github, Youtube, Namechk, Hunter, Fullcontant, Flickr but also specialized search engines such as Shodan or Cencys. And various network modules such as netcraft, whois, ssl, etc. Wikileaker and pwnedlist where you can search through leaked data sounds promising:-)

The reporting modules are used for the data export from the recon-ng database where there are really stored in the SQL database.

Let's say we want to replicate the functionality of Sherlock. We need to install the module profiler from the marketplace first.

We can check the information about the module we are interested and install all the modules we can at once by the commands:

marketplace info recon/profiles-profiles/profiler
marketplace install all

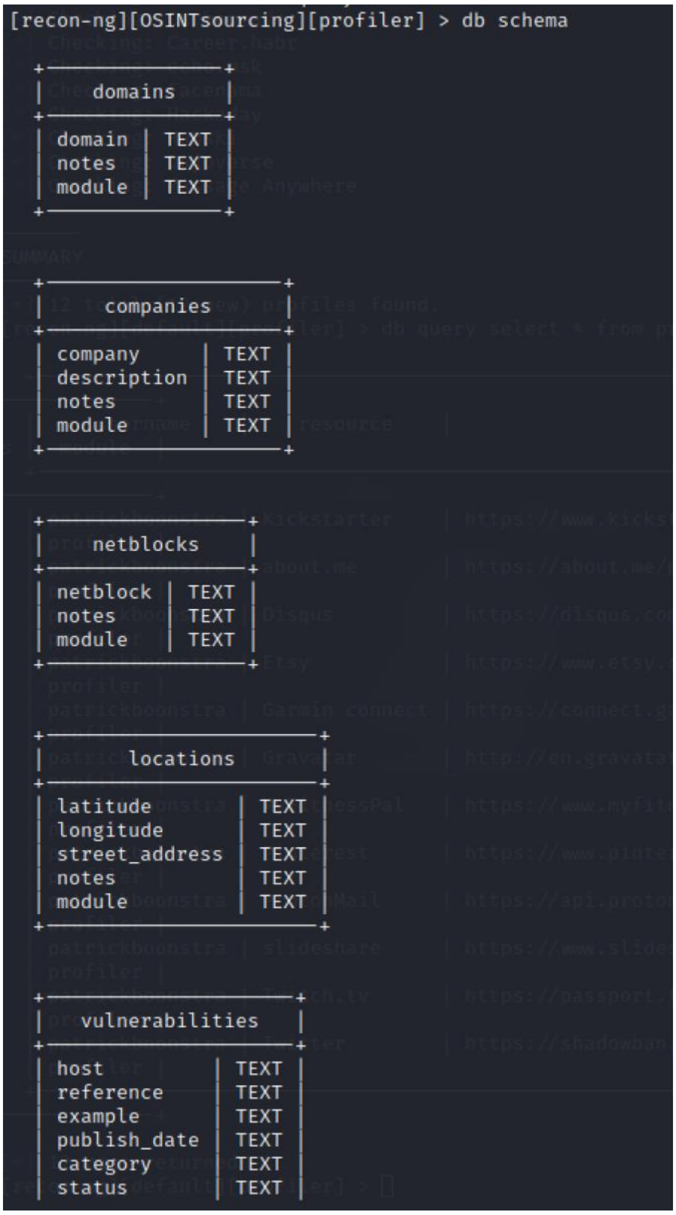


```
[recon-ng][OSINTsourcing] > marketplace info recon/profiles-profiles/profiler

  +---------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------+
  | path          | recon/profiles-profiles/profiler
  | name          | OSINT HUMINT Profile Collector
  | author        | Micah Hoffman (@WebBreacher)
  | version       | 1.0
  | last_updated  | 2019-06-24
  | description   | Takes each username from the profiles table and searches a variety of web sites for those users. The list of valid s
ites comes from the parent project at https://github.com/WebBreacher/WhatsMyName |
  | required_keys | []
  | dependencies  | []
  | files         | []
  | status        | installed
  +---------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------+

[recon-ng][OSINTsourcing] > marketplace install all
```

Now let's load the module *profiler*:

modules load recon/profiles-profiles/profiler

Before we start using it, we should explain how the database structure works. There are 13 tables in the database (domains, companies, netblocks, locations, vulnerabilities, ports, hosts, contacts, credentials, leaks, pushpins, profiles, repositories) and you can display them by the command *db schema*.



```
[recon-ng][OSINTsourcing][profiler] > db schema

  +-----------------+
  |     domains     |
  +-----------------+
  | domain  | TEXT  |
  | notes   | TEXT  |
  | module  | TEXT  |
  +-----------------+


  +----------------------+
  |      companies       |
  +----------------------+
  | company      | TEXT  |
  | description  | TEXT  |
  | notes        | TEXT  |
  | module       | TEXT  |
  +----------------------+


  +-------------------+
  |     netblocks     |
  +-------------------+
  | netblock  | TEXT  |
  | notes     | TEXT  |
  | module    | TEXT  |
  +-------------------+


  +-------------------------+
  |        locations        |
  +-------------------------+
  | latitude        | TEXT  |
  | longitude       | TEXT  |
  | street_address  | TEXT  |
  | notes           | TEXT  |
  | module          | TEXT  |
  +-------------------------+


  +-----------------------+
  |    vulnerabilities    |
  +-----------------------+
  | host          | TEXT  |
  | reference     | TEXT  |
  | example       | TEXT  |
  | publish_date  | TEXT  |
  | category      | TEXT  |
  | status        | TEXT  |
```

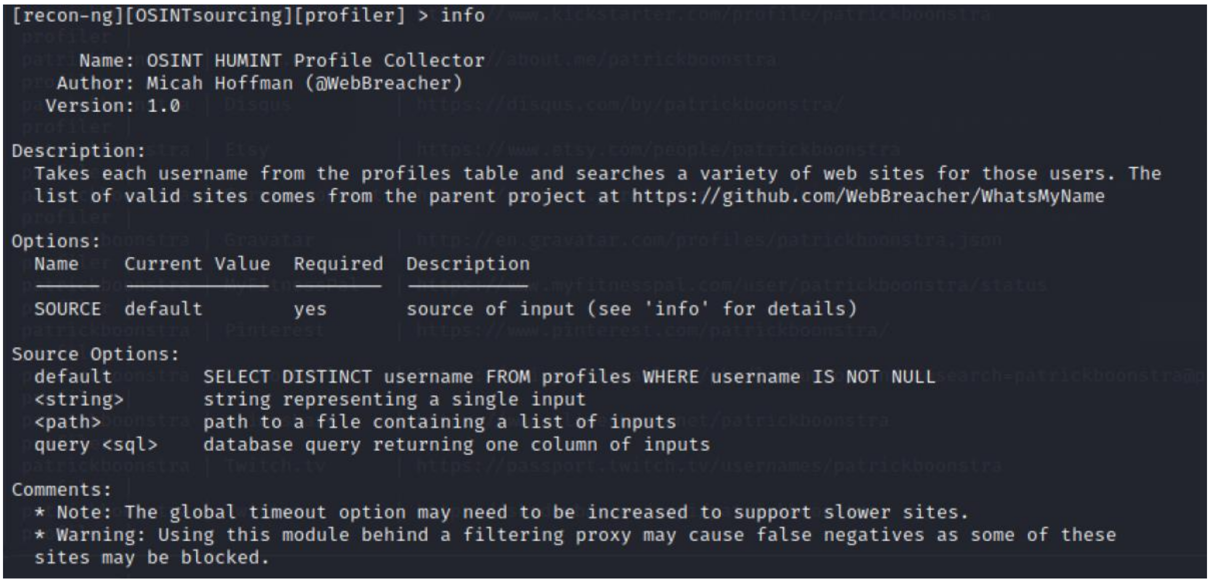
Note: The output is shortened.

These tables are empty at the beginning and this is where you data will end up but also some modules are reading data from there.

If you want to show the content of the specific table, e.g. profiles, execute this command:

*db query select * from profiles*

We are in the module profiler prompt mode, so we can type *info* just to find out what's expected as an input and output.



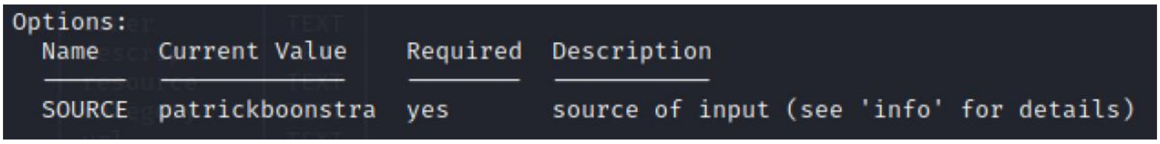
```
[recon-ng][OSINTsourcing][profiler] > info

      Name: OSINT HUMINT Profile Collector
    Author: Micah Hoffman (@WebBreacher)
   Version: 1.0

Description:
  Takes each username from the profiles table and searches a variety of web sites for those users. The
  list of valid sites comes from the parent project at https://github.com/WebBreacher/WhatsMyName

Options:
  Name    Current Value  Required  Description
  ------  -------------  --------  -----------
  SOURCE  default        yes       source of input (see 'info' for details)

Source Options:
  default        SELECT DISTINCT username FROM profiles WHERE username IS NOT NULL
  <string>       string representing a single input
  <path>         path to a file containing a list of inputs
  query <sql>    database query returning one column of inputs

Comments:
  * Note: The global timeout option may need to be increased to support slower sites.
  * Warning: Using this module behind a filtering proxy may cause false negatives as some of these
  sites may be blocked.
```

We can see that by default it’s going to take all usernames from the table *profiles* and apply this as an input. This might be the case when other module was writing some usernames into the *profiles* table or you can add them manually as well.

So, two options. One is that you set the value for the SOURCE parameter by the following command:

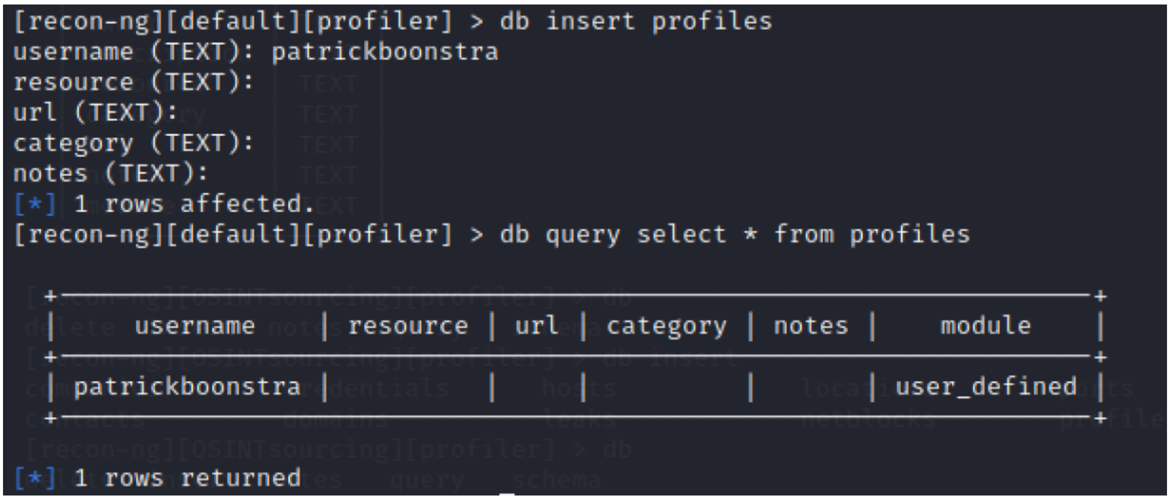
options set SOURCE patrickboonstra

When you hit info again, you should see that the value is there.



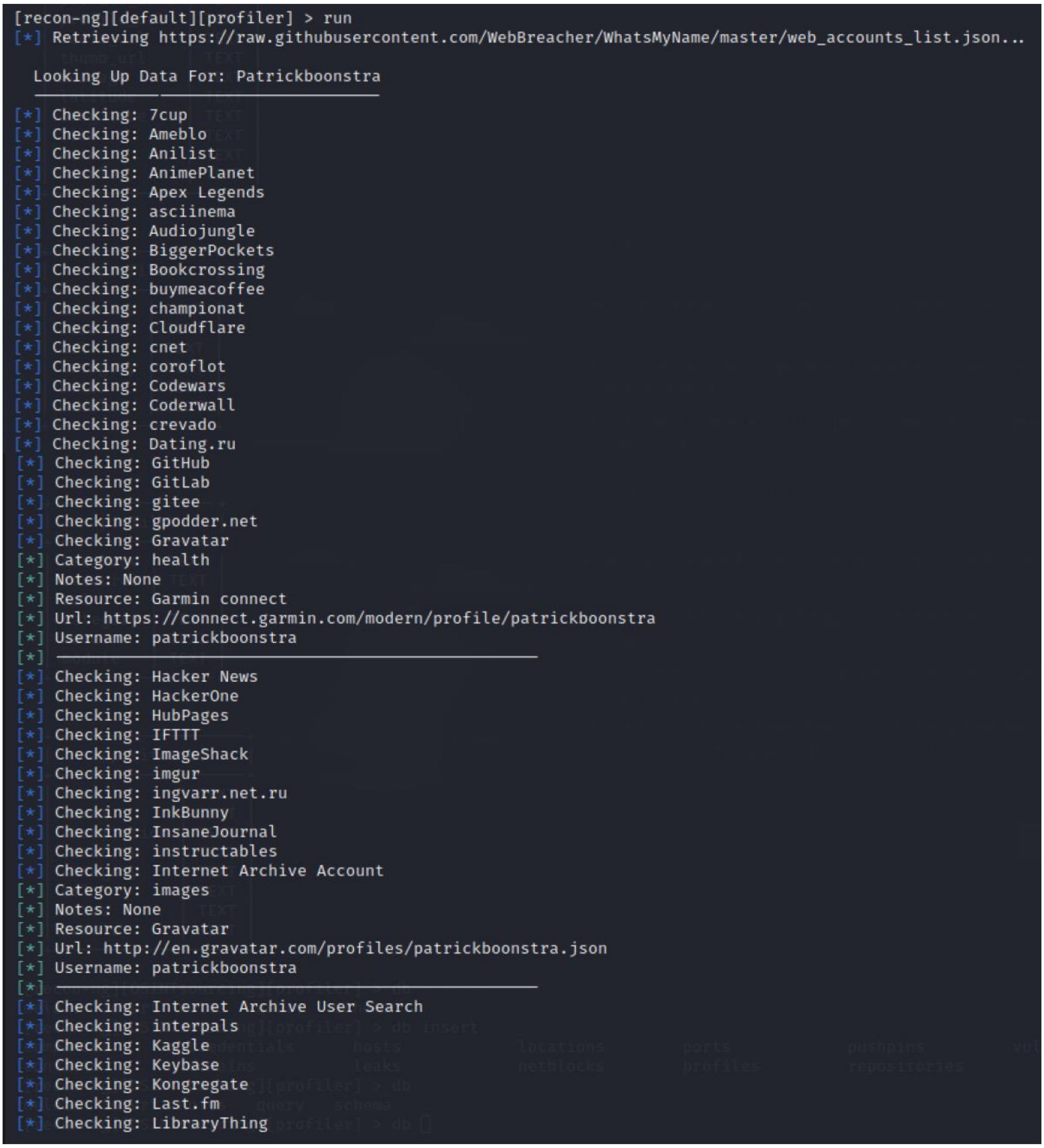
```
Options:
  Name    Current Value    Required  Description
  ------  ---------------  --------  -----------
  SOURCE  patrickboonstra  yes       source of input (see 'info' for details)
```

Or you add username(s) into the table and you don’t need to change any value then.



```
[recon-ng][default][profiler] > db insert profiles
username (TEXT): patrickboonstra
resource (TEXT):
url (TEXT):
category (TEXT):
notes (TEXT):
[*] 1 rows affected.
[recon-ng][default][profiler] > db query select * from profiles

  +---------------------------------------------------------------------------------+
  |     username    | resource | url | category | notes |    module    |
  +---------------------------------------------------------------------------------+
  | patrickboonstra |          |     |          |       | user_defined |
  +---------------------------------------------------------------------------------+

[*] 1 rows returned
```

Now we just need to run the module finally by the simple command *run*.



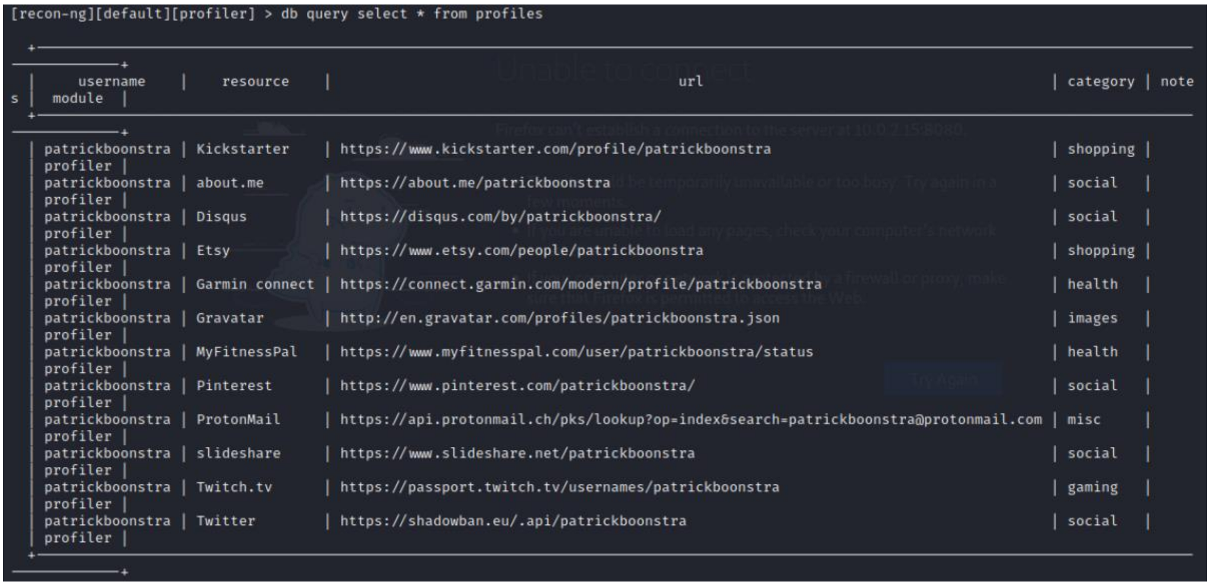
```
[recon-ng][default][profiler] > run
[*] Retrieving https://raw.githubusercontent.com/WebBreacher/WhatsMyName/master/web_accounts_list.json...

  Looking Up Data For: Patrickboonstra
  ───────────────────────────────────

[*] Checking: 7cup
[*] Checking: Ameblo
[*] Checking: Anilist
[*] Checking: AnimePlanet
[*] Checking: Apex Legends
[*] Checking: asciinema
[*] Checking: Audiojungle
[*] Checking: BiggerPockets
[*] Checking: Bookcrossing
[*] Checking: buymeacoffee
[*] Checking: championat
[*] Checking: Cloudflare
[*] Checking: cnet
[*] Checking: coroflot
[*] Checking: Codewars
[*] Checking: Coderwall
[*] Checking: crevado
[*] Checking: Dating.ru
[*] Checking: GitHub
[*] Checking: GitLab
[*] Checking: gitee
[*] Checking: gpodder.net
[*] Checking: Gravatar
[*] Category: health
[*] Notes: None
[*] Resource: Garmin connect
[*] Url: https://connect.garmin.com/modern/profile/patrickboonstra
[*] Username: patrickboonstra
[*] ──────────────────────────────────────────────────
[*] Checking: Hacker News
[*] Checking: HackerOne
[*] Checking: HubPages
[*] Checking: IFTTT
[*] Checking: ImageShack
[*] Checking: imgur
[*] Checking: ingvarr.net.ru
[*] Checking: InkBunny
[*] Checking: InsaneJournal
[*] Checking: instructables
[*] Checking: Internet Archive Account
[*] Category: images
[*] Notes: None
[*] Resource: Gravatar
[*] Url: http://en.gravatar.com/profiles/patrickboonstra.json
[*] Username: patrickboonstra
[*] ──────────────────────────────────────────────────
[*] Checking: Internet Archive User Search
[*] Checking: interpals
[*] Checking: Kaggle
[*] Checking: Keybase
[*] Checking: Kongregate
[*] Checking: Last.fm
[*] Checking: LibraryThing
```

The output is shortened so you cannot see the whole list of what all this is checking but it is pretty comprehensive. Luckily, for Patrick, we haven't found his account on redTube, Porn Hub and Dating.ru :-)

Where did we found Patrick's username?

The best way is to pull it from the database – from the table *profiles* in specific. If you have ever done dome SQL commands, this will be familiar to you:

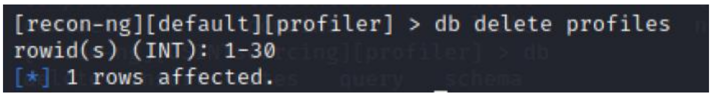
*db query select * from profiles*



```
[recon-ng][default][profiler] > db query select * from profiles

  +-------------------------------------------------------------------------------------------------------------------------------------
----------+
  |     username     |    resource     |                                  url                                                  | category | note
s |  module  |
  +-------------------------------------------------------------------------------------------------------------------------------------
----------+
  | patrickboonstra | Kickstarter     | https://www.kickstarter.com/profile/patrickboonstra                                   | shopping |
  | profiler |
  | patrickboonstra | about.me        | https://about.me/patrickboonstra                                                      | social   |
  | profiler |
  | patrickboonstra | Disqus          | https://disqus.com/by/patrickboonstra/                                                | social   |
  | profiler |
  | patrickboonstra | Etsy            | https://www.etsy.com/people/patrickboonstra                                           | shopping |
  | profiler |
  | patrickboonstra | Garmin connect  | https://connect.garmin.com/modern/profile/patrickboonstra                             | health   |
  | profiler |
  | patrickboonstra | Gravatar        | http://en.gravatar.com/profiles/patrickboonstra.json                                  | images   |
  | profiler |
  | patrickboonstra | MyFitnessPal    | https://www.myfitnesspal.com/user/patrickboonstra/status                              | health   |
  | profiler |
  | patrickboonstra | Pinterest       | https://www.pinterest.com/patrickboonstra/                                            | social   |
  | profiler |
  | patrickboonstra | ProtonMail      | https://api.protonmail.ch/pks/lookup?op=index&search=patrickboonstra@protonmail.com | misc     |
  | profiler |
  | patrickboonstra | slideshare      | https://www.slideshare.net/patrickboonstra                                            | social   |
  | profiler |
  | patrickboonstra | Twitch.tv       | https://passport.twitch.tv/usernames/patrickboonstra                                  | gaming   |
  | profiler |
  | patrickboonstra | Twitter         | https://shadowban.eu/.api/patrickboonstra                                             | social   |
  | profiler |
  +-------------------------------------------------------------------------------------------------------------------------------------
----------+
```

If you want to delete something or everything from the table *profiles*, just use the command:

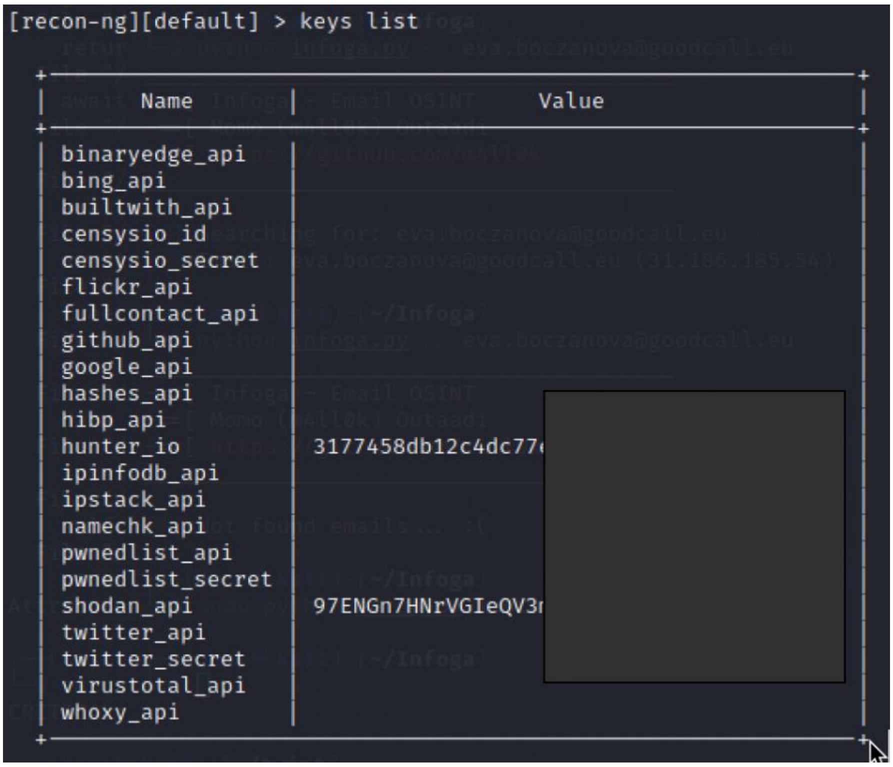
db delete profiles



```
[recon-ng][default][profiler] > db delete profiles
rowid(s) (INT): 1-30
[*] 1 rows affected.
```

Recon-ng will require API keys for the certain services. Type the following command to see the table with the keys:

keys list



```
[recon-ng][default] > keys list

  +------------------------------------------------------------+
  |       Name       |                 Value                   |
  +------------------------------------------------------------+
  | binaryedge_api   |                                         |
  | bing_api         |                                         |
  | builtwith_api    |                                         |
  | censysio_id      |                                         |
  | censysio_secret  |                                         |
  | flickr_api       |                                         |
  | fullcontact_api  |                                         |
  | github_api       |                                         |
  | google_api       |                                         |
  | hashes_api       |                                         |
  | hibp_api         |                                         |
  | hunter_io        | 3177458db12c4dc77 [redacted]            |
  | ipinfodb_api     |                                         |
  | ipstack_api      |                                         |
  | namechk_api      |                                         |
  | pwnedlist_api    |                                         |
  | pwnedlist_secret |                                         |
  | shodan_api       | 97ENGn7HNrVGIeQV3                       |
  | twitter_api      |                                         |
  | twitter_secret   |                                         |
  | virustotal_api   |                                         |
  | whoxy_api        |                                         |
  +------------------------------------------------------------+
```

If you want to add a new API key, you can do it with the following command:

keys add shodan_api 97ENGn7HNrVGIeQxxxxxxxxx

The API key is usually provided on the website of the specific service. It equals to your password so you shouldn't share it openly.

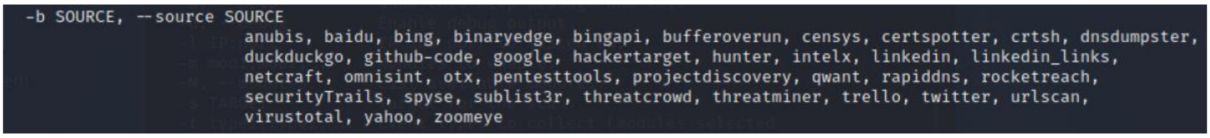
theHarvester

theHarvester is another Linux-based OSINT tool to research on the specific Internet domains. It can find things such as e-mail addresses, subdomains, IPs and more.

theHarvester is installed on Kali Linux by default. If you need to install it for example in Google Cloud Console, you would use the following commands:

git clone https://github.com/laramies/theHarvester
cd theHarvester
python3 -m pip install -r requirements/base.txt

The standard feature about this tool is that you can specify a source of research including Google, Bing, LinkedIn, Twitter, Baidu, IntelX, Hunter, Hackertarget, Github, Zoomeye and others.



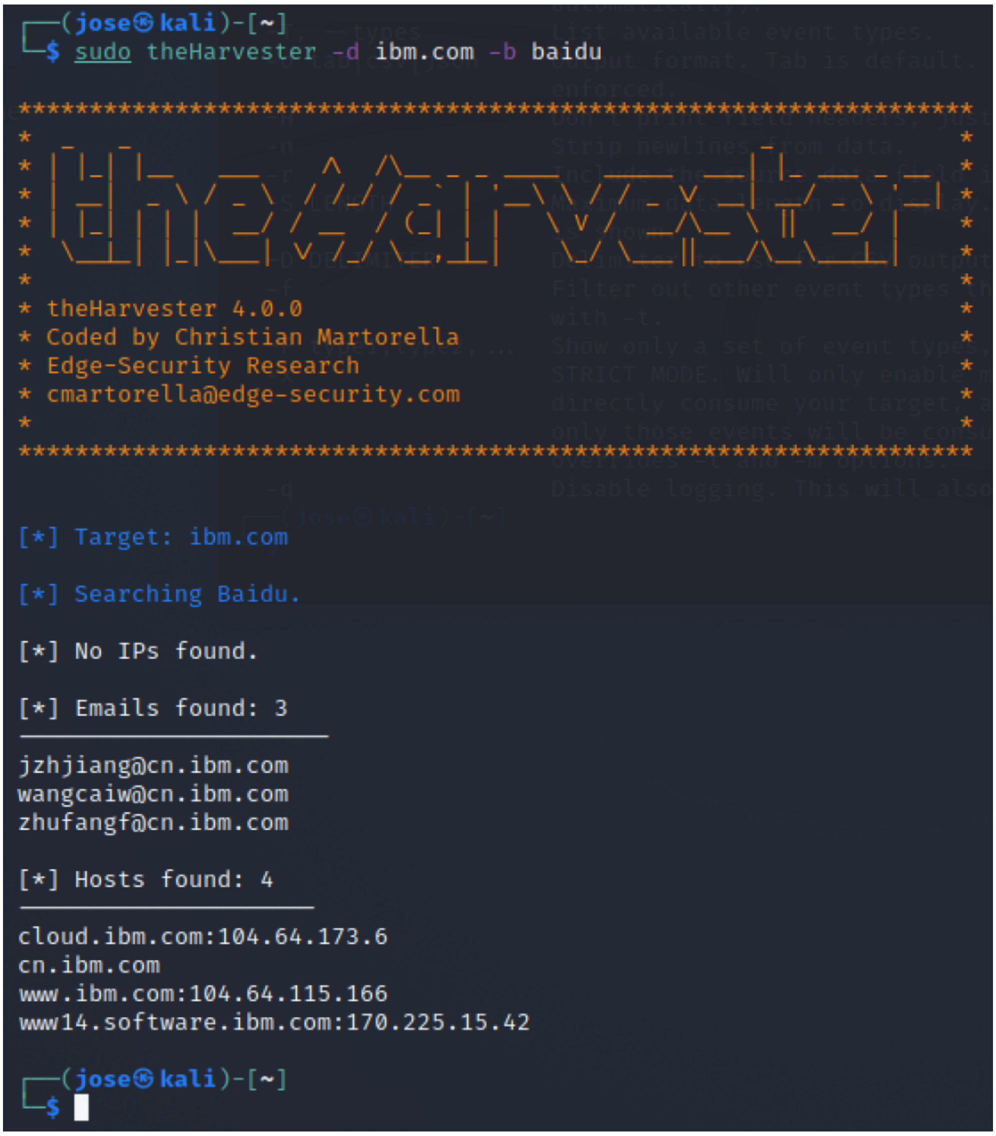
```
  -b SOURCE, --source SOURCE
                        anubis, baidu, bing, binaryedge, bingapi, bufferoverun, censys, certspotter, crtsh, dnsdumpster,
                        duckduckgo, github-code, google, hackertarget, hunter, intelx, linkedin, linkedin_links,
                        netcraft, omnisint, otx, pentesttools, projectdiscovery, qwant, rapiddns, rocketreach,
                        securityTrails, spyse, sublist3r, threatcrowd, threatminer, trello, twitter, urlscan,
                        virustotal, yahoo, zoomeye
```

You can also let search the found hosts through Shodan – specialized security search engine (I will focus on this tool in another publication or article). Or you can not only search the target by Google but also with some more extensive Google dorks:

sudo theHarvester -d ibm.com -b google -g

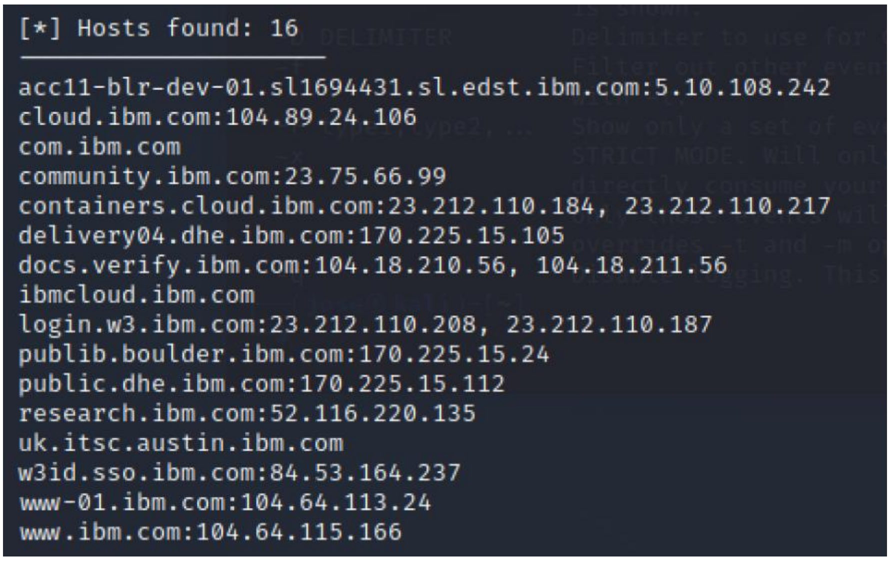
So, let's say I want to search the targeted domain name ibm.com on Baidu:

sudo theHarvester -d ibm.com -b baidu



```
┌──(jose㉿kali)-[~]
└─$ sudo theHarvester -d ibm.com -b baidu

*******************************************************************
*  _   _                                            _             *
* | |_| |__   ___    /\  /\__ _ _ ____   _____  ___| |_ ___ _ __  *
* | __|  _ \ / _ \  / /_/ / _` | '__\ \ / / _ \/ __| __/ _ \ '__| *
* | |_| | | |  __/ / __  / (_| | |   \ V /  __/\__ \ ||  __/ |    *
*  \__|_| |_|\___| \/ /_/ \__,_|_|    \_/ \___||___/\__\___|_|    *
*                                                                 *
* theHarvester 4.0.0                                              *
* Coded by Christian Martorella                                   *
* Edge-Security Research                                          *
* cmartorella@edge-security.com                                   *
*                                                                 *
*******************************************************************


[*] Target: ibm.com

[*] Searching Baidu.

[*] No IPs found.

[*] Emails found: 3
----------------------
jzhjiang@cn.ibm.com
wangcaiw@cn.ibm.com
zhufangf@cn.ibm.com

[*] Hosts found: 4
---------------------
cloud.ibm.com:104.64.173.6
cn.ibm.com
www.ibm.com:104.64.115.166
www14.software.ibm.com:170.225.15.42

┌──(jose㉿kali)-[~]
└─$ █
```

We found some e-mails and some hosts as well. Using other sources such as Google, otx Bufferoverrun, Hackertarget, sublist3r or urlscan you can find tens of thousands more subdomains which might lead you to further source of information. This is the output from the source Google only:



```
[*] Hosts found: 16
---------------------
acc11-blr-dev-01.sl1694431.sl.edst.ibm.com:5.10.108.242
cloud.ibm.com:104.89.24.106
com.ibm.com
community.ibm.com:23.75.66.99
containers.cloud.ibm.com:23.212.110.184, 23.212.110.217
delivery04.dhe.ibm.com:170.225.15.105
docs.verify.ibm.com:104.18.210.56, 104.18.211.56
ibmcloud.ibm.com
login.w3.ibm.com:23.212.110.208, 23.212.110.187
publib.boulder.ibm.com:170.225.15.24
public.dhe.ibm.com:170.225.15.112
research.ibm.com:52.116.220.135
uk.itsc.austin.ibm.com
w3id.sso.ibm.com:84.53.164.237
www-01.ibm.com:104.64.113.24
www.ibm.com:104.64.115.166
```

Note: Some of the sources such as GitHub, Censys, IntelX, Hunter, Zoomeye or Shodan will need your API key to be imported like for *Recon-ng*.

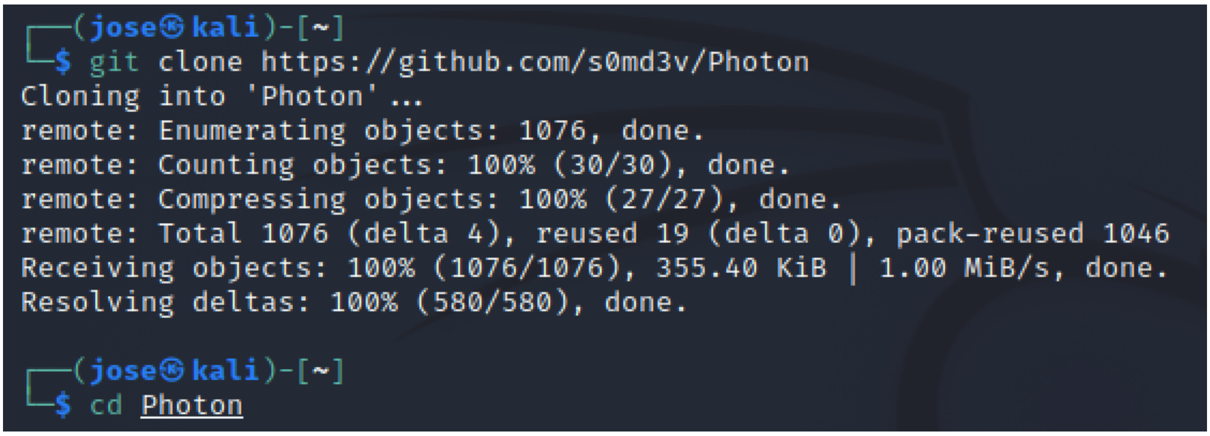
Photon

Photon is another OSINT tool to make research on a specific domain name. You can extract the following information:

- URLs (in-scope & out-of-scope)
- URLs with parameters (example.com/gallery.php?id=2)
- Intel (emails, social media accounts, etc.)
- Files (pdf, png, xml etc.)
- Secret keys (auth/API keys & hashes)
- JavaScript files & Endpoints present in them
- Strings matching custom regex pattern
- Subdomains & DNS related data

The installation is simple. Just clone it from git, skip into the directory *Photon* and you're good to go.

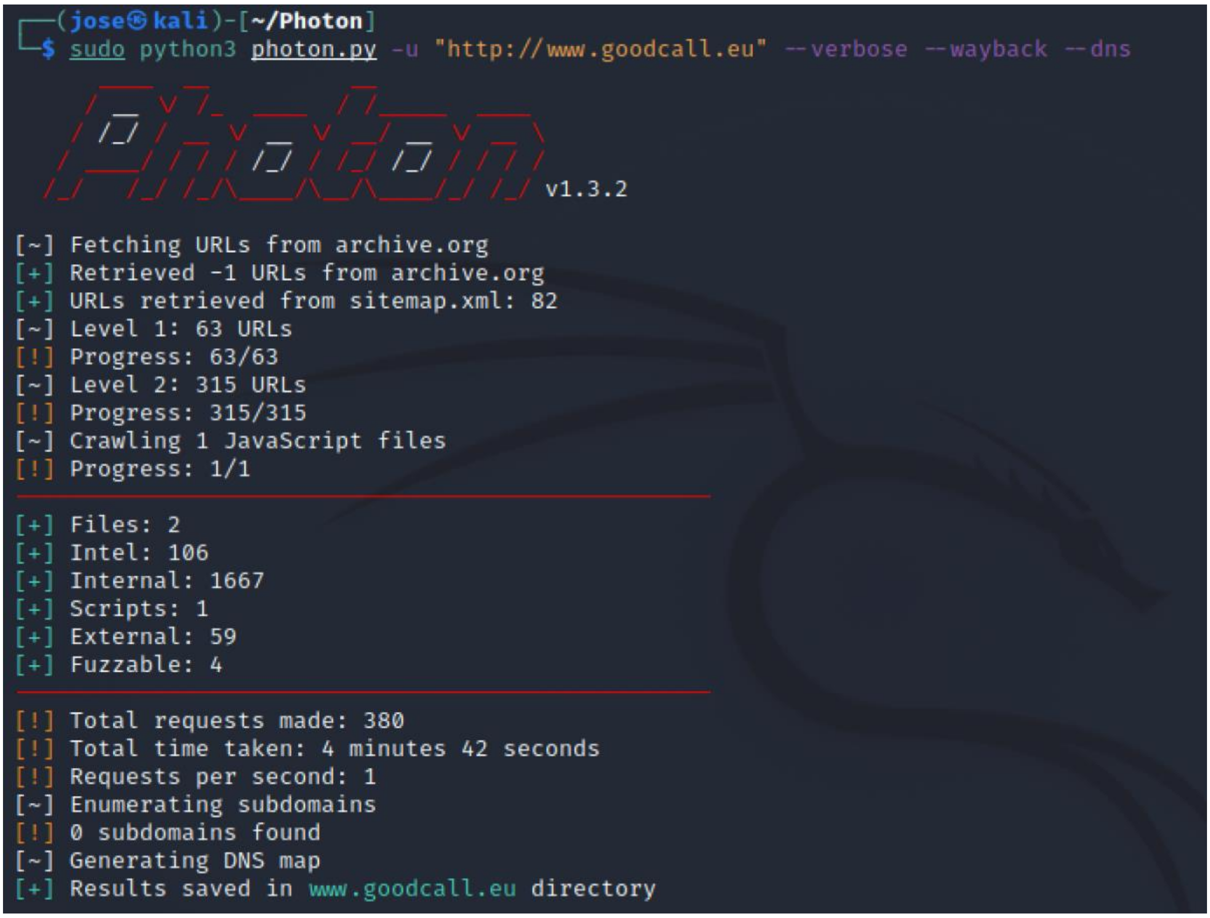
git clone https://github.com/s0md3v/Photon
cd Photon



```
┌──(jose㉿kali)-[~]
└─$ git clone https://github.com/s0md3v/Photon
Cloning into 'Photon' ...
remote: Enumerating objects: 1076, done.
remote: Counting objects: 100% (30/30), done.
remote: Compressing objects: 100% (27/27), done.
remote: Total 1076 (delta 4), reused 19 (delta 0), pack-reused 1046
Receiving objects: 100% (1076/1076), 355.40 KiB | 1.00 MiB/s, done.
Resolving deltas: 100% (580/580), done.

┌──(jose㉿kali)-[~]
└─$ cd Photon
```

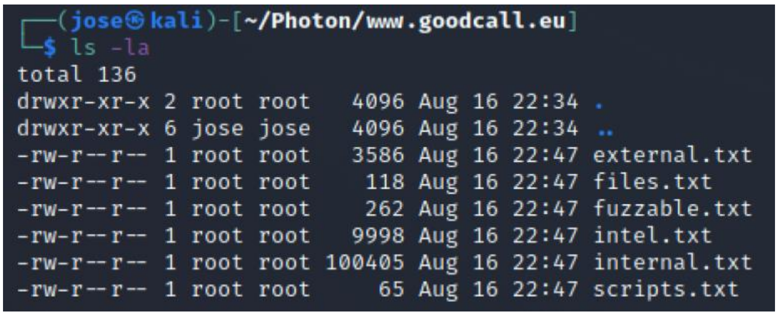
Let’s say I want to learn more about the company GoodCall by extracting information from their website.

sudo python3 photon.py -u "http://www.goodcall.eu" --verbose --wayback --dns



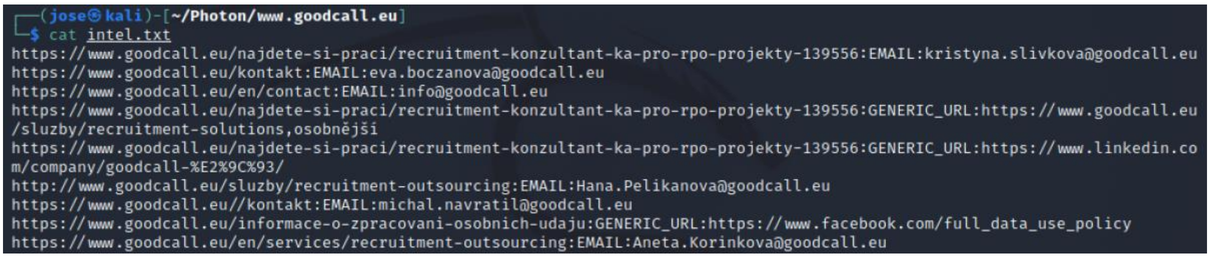
```
┌──(jose㉿kali)-[~/Photon]
└─$ sudo python3 photon.py -u "http://www.goodcall.eu" --verbose --wayback --dns
      ____  __          __
     / __ \/ /_  ____  / /_____  ____
    / /_/ / __ \/ __ \/ __/ __ \/ __ \
   / ____/ / / / /_/ / /_/ /_/ / / / /
  /_/   /_/ /_/\____/\__/\____/_/ /_/ v1.3.2

[~] Fetching URLs from archive.org
[+] Retrieved -1 URLs from archive.org
[+] URLs retrieved from sitemap.xml: 82
[~] Level 1: 63 URLs
[!] Progress: 63/63
[~] Level 2: 315 URLs
[!] Progress: 315/315
[~] Crawling 1 JavaScript files
[!] Progress: 1/1

[+] Files: 2
[+] Intel: 106
[+] Internal: 1667
[+] Scripts: 1
[+] External: 59
[+] Fuzzable: 4

[!] Total requests made: 380
[!] Total time taken: 4 minutes 42 seconds
[!] Requests per second: 1
[~] Enumerating subdomains
[!] 0 subdomains found
[~] Generating DNS map
[+] Results saved in www.goodcall.eu directory
```

The whole output is saved into a directory named by the researched domain name and organized into a few files.



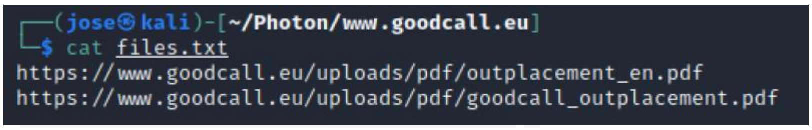
```
┌──(jose㉿kali)-[~/Photon/www.goodcall.eu]
└─$ ls -la
total 136
drwxr-xr-x 2 root root   4096 Aug 16 22:34 .
drwxr-xr-x 6 jose jose   4096 Aug 16 22:34 ..
-rw-r--r-- 1 root root   3586 Aug 16 22:47 external.txt
-rw-r--r-- 1 root root    118 Aug 16 22:47 files.txt
-rw-r--r-- 1 root root    262 Aug 16 22:47 fuzzable.txt
-rw-r--r-- 1 root root   9998 Aug 16 22:47 intel.txt
-rw-r--r-- 1 root root 100405 Aug 16 22:47 internal.txt
-rw-r--r-- 1 root root     65 Aug 16 22:47 scripts.txt
```

Let’s open the file *intel.txt* where you can find the e-mail addresses which appeared on the concrete pages.



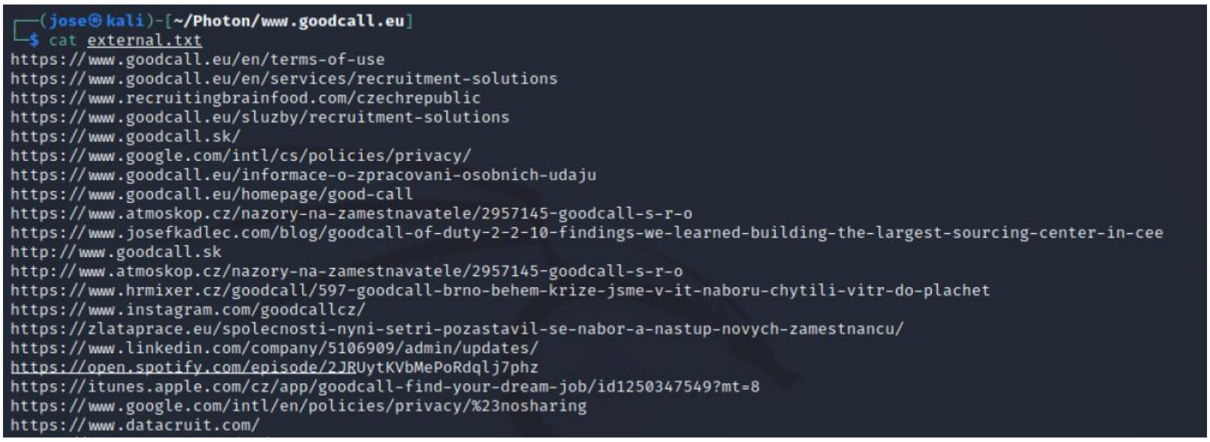
```
┌──(jose㉿kali)-[~/Photon/www.goodcall.eu]
└─$ cat intel.txt
https://www.goodcall.eu/najdete-si-praci/recruitment-konzultant-ka-pro-rpo-projekty-139556:EMAIL:kristyna.slivkova@goodcall.eu
https://www.goodcall.eu/kontakt:EMAIL:eva.boczanova@goodcall.eu
https://www.goodcall.eu/en/contact:EMAIL:info@goodcall.eu
https://www.goodcall.eu/najdete-si-praci/recruitment-konzultant-ka-pro-rpo-projekty-139556:GENERIC_URL:https://www.goodcall.eu
/sluzby/recruitment-solutions,osobnější
https://www.goodcall.eu/najdete-si-praci/recruitment-konzultant-ka-pro-rpo-projekty-139556:GENERIC_URL:https://www.linkedin.co
m/company/goodcall-%E2%9C%93/
http://www.goodcall.eu/sluzby/recruitment-outsourcing:EMAIL:Hana.Pelikanova@goodcall.eu
https://www.goodcall.eu//kontakt:EMAIL:michal.navratil@goodcall.eu
https://www.goodcall.eu/informace-o-zpracovani-osobnich-udaju:GENERIC_URL:https://www.facebook.com/full_data_use_policy
https://www.goodcall.eu/en/services/recruitment-outsourcing:EMAIL:Aneta.Korinkova@goodcall.eu
```

Files.txt can uncover some PDF and other files on the domain name:



```
┌──(jose㉿kali)-[~/Photon/www.goodcall.eu]
└─$ cat files.txt
https://www.goodcall.eu/uploads/pdf/outplacement_en.pdf
https://www.goodcall.eu/uploads/pdf/goodcall_outplacement.pdf
```

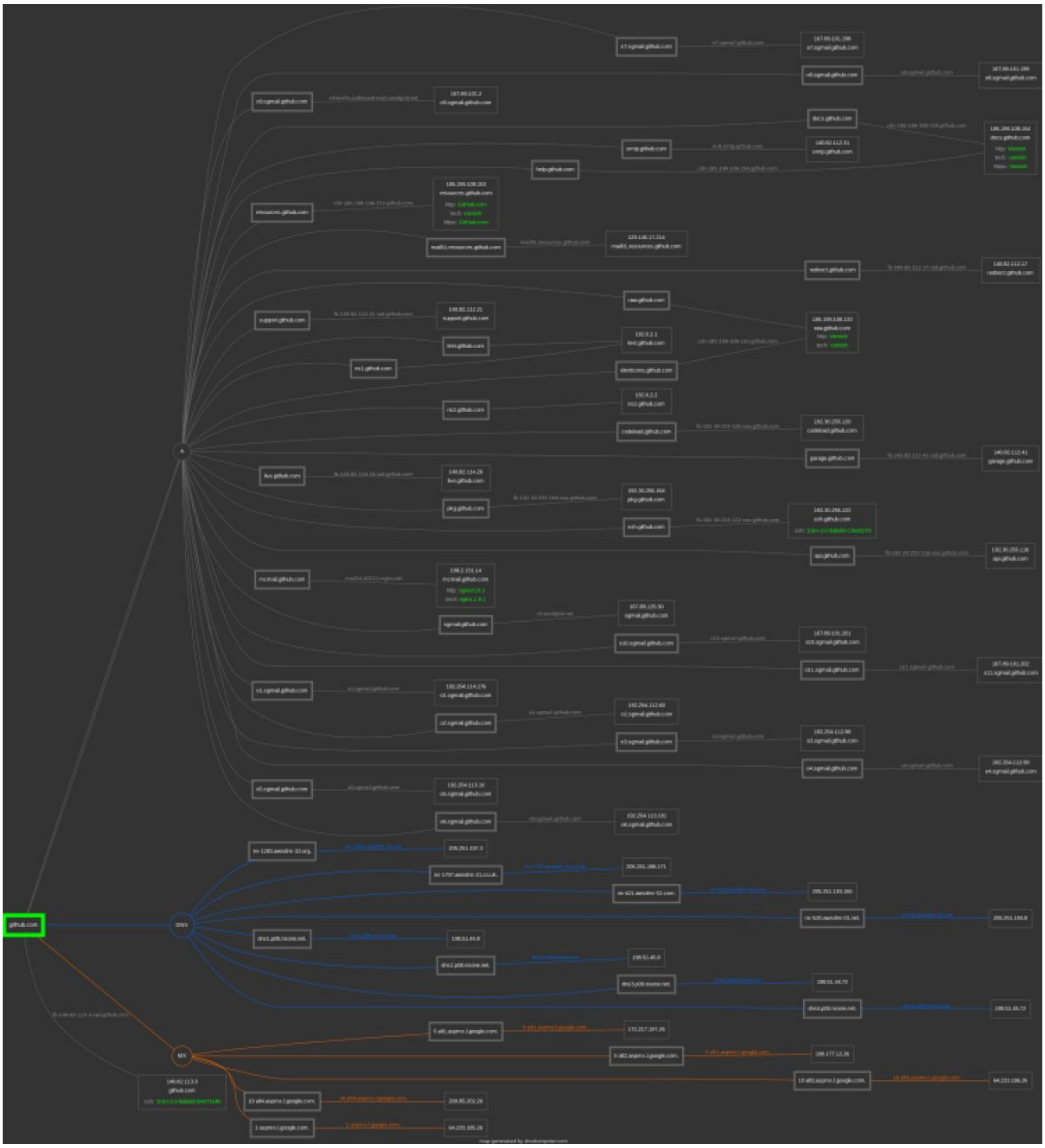
You can also find a list of internal and external links in *external.txt*:



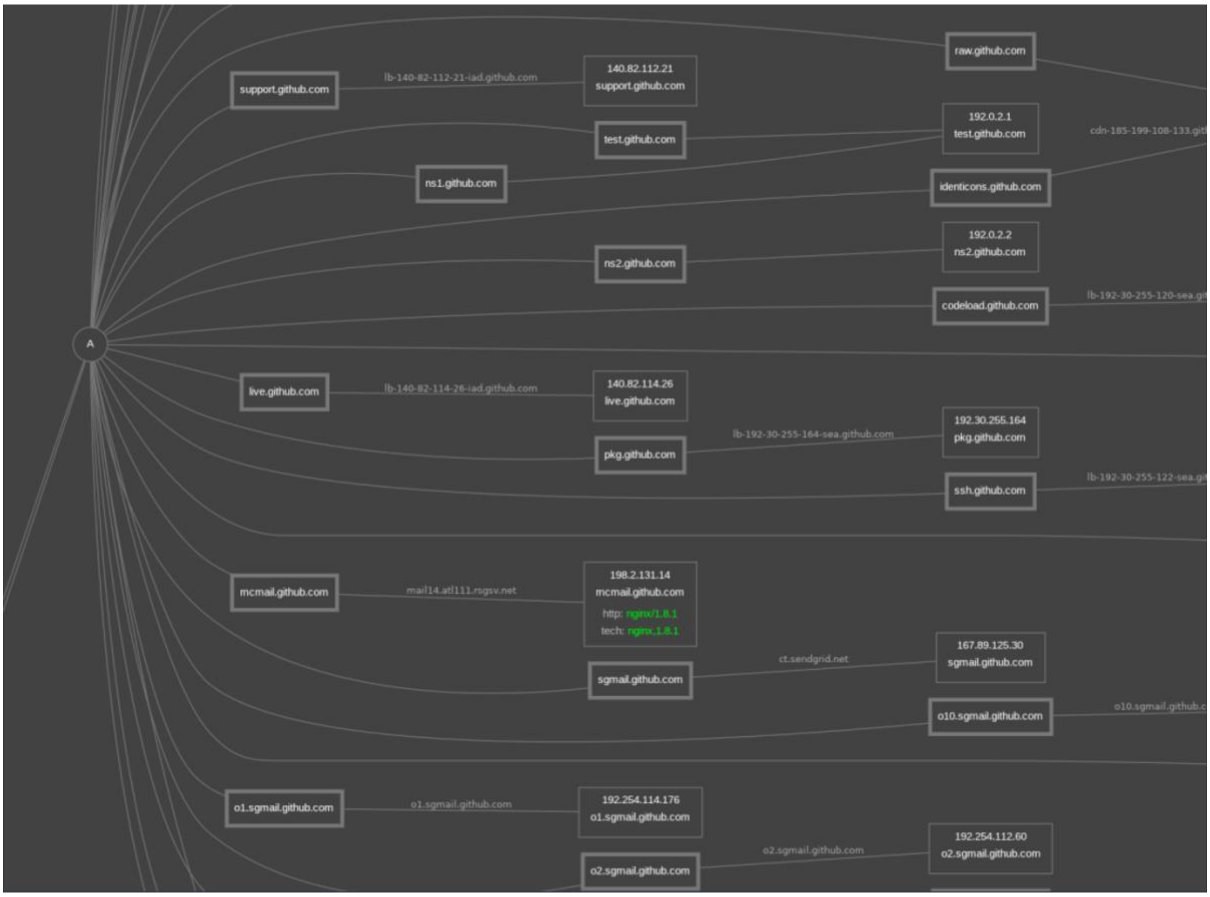
```
┌──(jose㉿kali)-[~/Photon/www.goodcall.eu]
└─$ cat external.txt
https://www.goodcall.eu/en/terms-of-use
https://www.goodcall.eu/en/services/recruitment-solutions
https://www.recruitingbrainfood.com/czechrepublic
https://www.goodcall.eu/sluzby/recruitment-solutions
https://www.goodcall.sk/
https://www.google.com/intl/cs/policies/privacy/
https://www.goodcall.eu/informace-o-zpracovani-osobnich-udaju
https://www.goodcall.eu/homepage/good-call
https://www.atmoskop.cz/nazory-na-zamestnavatele/2957145-goodcall-s-r-o
https://www.josefkadlec.com/blog/goodcall-of-duty-2-2-10-findings-we-learned-building-the-largest-sourcing-center-in-cee
http://www.goodcall.sk
http://www.atmoskop.cz/nazory-na-zamestnavatele/2957145-goodcall-s-r-o
https://www.hrmixer.cz/goodcall/597-goodcall-brno-behem-krize-jsme-v-it-naboru-chytili-vitr-do-plachet
https://www.instagram.com/goodcallcz/
https://zlataprace.eu/spolecnosti-nyni-setri-pozastavil-se-nabor-a-nastup-novych-zamestnancu/
https://www.linkedin.com/company/5106909/admin/updates/
https://open.spotify.com/episode/2JRUytKVbMePoRdqlj7phz
https://itunes.apple.com/cz/app/goodcall-find-your-dream-job/id1250347549?mt=8
https://www.google.com/intl/en/policies/privacy/%23nosharing
https://www.datacruit.com/
```

The --*dns* switch is going to make a map of subdomains, DNS and MX. I ran it for the domain *github.com* and get the following result.

This is the whole overview:



Let’s zoom in on the subdomains a little bit:



The switch --*wayback* can pull data from archive.org for a website which doesn't exist anymore for example.

It is also possible to crawl the websites or the parts of the website where you have to authenticate. With the switch -c you can specify the cookie from your authentication.

You can also increase the default number of 2 threads but don't forget it works like any other scraping and crawling, you can trigger security mechanisms preventing you from continuing. You can also set up a proxy or change the user agent (the website can see you as a Google bot or as an Android phone for instance).

It's probably no surprise that you can save the output as CSV or JSON by using --*export=csv* and --*export=json*.

Other similar command line tools like *theHarverster* and *Photon* include Dmitry, sublist3r, Datasploit, Belati, Fierce, DNStwist and Gas Mask. Also, the tool SRFramework which can provide more information on domains, usernames, e-mail addresses or telephone numbers and has a graphical output eventually as well.

Infoga

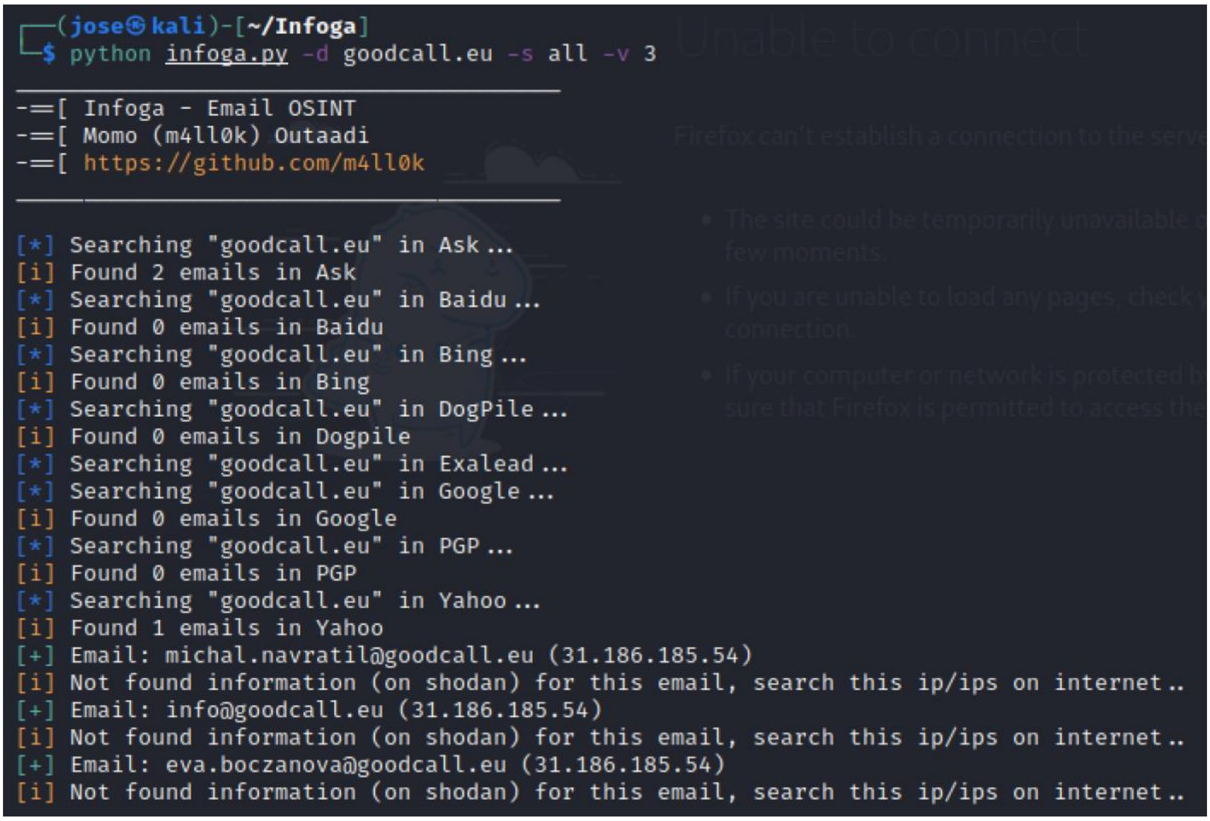
Infoga is another research tool focused on domain names and e-mail addresses eventually.

Download and install it from git by these commands:

git clone https://github.com/m4ll0k/Infoga.git
cd Infoga
sudo python setup.py install

Run the search by the following command covering all sources it can:

python infoga.py -d goodcall.eu -s all -v 3



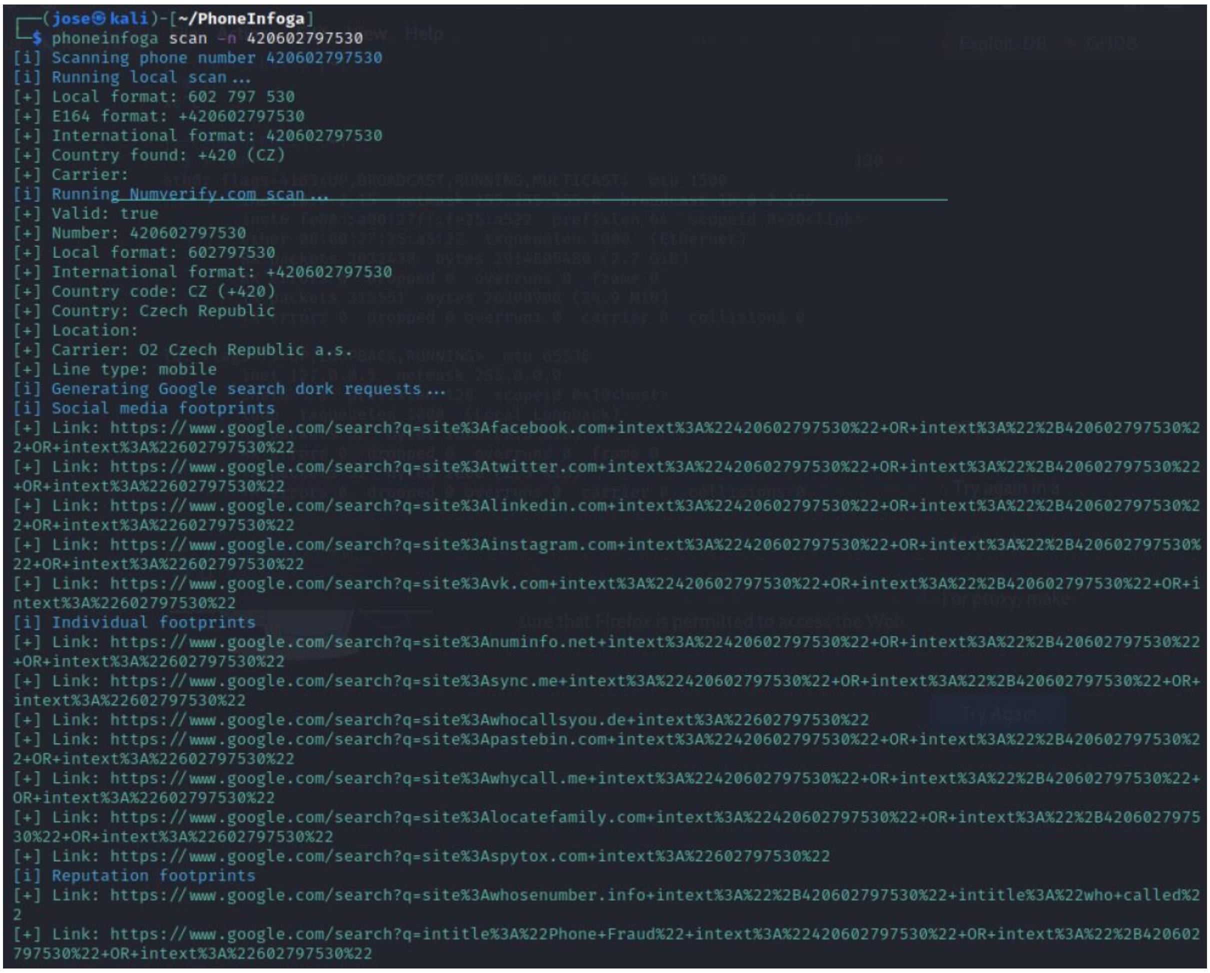
```
┌──(jose㉿kali)-[~/Infoga]
└─$ python infoga.py -d goodcall.eu -s all -v 3

-=[ Infoga - Email OSINT
-=[ Momo (m4ll0k) Outaadi
-=[ https://github.com/m4ll0k

[*] Searching "goodcall.eu" in Ask ...
[i] Found 2 emails in Ask
[*] Searching "goodcall.eu" in Baidu ...
[i] Found 0 emails in Baidu
[*] Searching "goodcall.eu" in Bing ...
[i] Found 0 emails in Bing
[*] Searching "goodcall.eu" in DogPile ...
[i] Found 0 emails in Dogpile
[*] Searching "goodcall.eu" in Exalead ...
[*] Searching "goodcall.eu" in Google ...
[i] Found 0 emails in Google
[*] Searching "goodcall.eu" in PGP ...
[i] Found 0 emails in PGP
[*] Searching "goodcall.eu" in Yahoo ...
[i] Found 1 emails in Yahoo
[+] Email: michal.navratil@goodcall.eu (31.186.185.54)
[i] Not found information (on shodan) for this email, search this ip/ips on internet..
[+] Email: info@goodcall.eu (31.186.185.54)
[i] Not found information (on shodan) for this email, search this ip/ips on internet..
[+] Email: eva.boczanova@goodcall.eu (31.186.185.54)
[i] Not found information (on shodan) for this email, search this ip/ips on internet..
```

Phoneinfoga

Phoneinfoga is a simple shell application (having a web GUI actually as well) to research on a specific phone number.

This tool doesn’t do any active OSINT by connecting to the networks or cracking any mobile network perimeters.

Let’s run the search on my cell phone number.



```
┌──(jose㉿kali)-[~/PhoneInfoga]
└─$ phoneinfoga scan -n 420602797530
[i] Scanning phone number 420602797530
[i] Running local scan ...
[+] Local format: 602 797 530
[+] E164 format: +420602797530
[+] International format: 420602797530
[+] Country found: +420 (CZ)
[+] Carrier:
[i] Running Numverify.com scan ...
[+] Valid: true
[+] Number: 420602797530
[+] Local format: 602797530
[+] International format: +420602797530
[+] Country code: CZ (+420)
[+] Country: Czech Republic
[+] Location:
[+] Carrier: O2 Czech Republic a.s.
[+] Line type: mobile
[i] Generating Google search dork requests ...
[i] Social media footprints
[+] Link: https://www.google.com/search?q=site%3Afacebook.com+intext%3A%22420602797530%22+OR+intext%3A%22%2B420602797530%2
2+OR+intext%3A%22602797530%22
[+] Link: https://www.google.com/search?q=site%3Atwitter.com+intext%3A%22420602797530%22+OR+intext%3A%22%2B420602797530%22
+OR+intext%3A%22602797530%22
[+] Link: https://www.google.com/search?q=site%3Alinkedin.com+intext%3A%22420602797530%22+OR+intext%3A%22%2B420602797530%2
2+OR+intext%3A%22602797530%22
[+] Link: https://www.google.com/search?q=site%3Ainstagram.com+intext%3A%22420602797530%22+OR+intext%3A%22%2B420602797530%
22+OR+intext%3A%22602797530%22
[+] Link: https://www.google.com/search?q=site%3Avk.com+intext%3A%22420602797530%22+OR+intext%3A%22%2B420602797530%22+OR+i
ntext%3A%22602797530%22
[i] Individual footprints
[+] Link: https://www.google.com/search?q=site%3Anuminfo.net+intext%3A%22420602797530%22+OR+intext%3A%22%2B420602797530%22
+OR+intext%3A%22602797530%22
[+] Link: https://www.google.com/search?q=site%3Async.me+intext%3A%22420602797530%22+OR+intext%3A%22%2B420602797530%22+OR+
intext%3A%22602797530%22
[+] Link: https://www.google.com/search?q=site%3Awhocallsyou.de+intext%3A%22602797530%22
[+] Link: https://www.google.com/search?q=site%3Apastebin.com+intext%3A%22420602797530%22+OR+intext%3A%22%2B420602797530%2
2+OR+intext%3A%22602797530%22
[+] Link: https://www.google.com/search?q=site%3Awhycall.me+intext%3A%22420602797530%22+OR+intext%3A%22%2B420602797530%22+
OR+intext%3A%22602797530%22
[+] Link: https://www.google.com/search?q=site%3Alocatefamily.com+intext%3A%22420602797530%22+OR+intext%3A%22%2B4206027975
30%22+OR+intext%3A%22602797530%22
[+] Link: https://www.google.com/search?q=site%3Aspytox.com+intext%3A%22602797530%22
[i] Reputation footprints
[+] Link: https://www.google.com/search?q=site%3Awhosenumber.info+intext%3A%22%2B420602797530%22+intitle%3A%22who+called%2
2
[+] Link: https://www.google.com/search?q=intitle%3A%22Phone+Fraud%22+intext%3A%22420602797530%22+OR+intext%3A%22%2B420602
797530%22+OR+intext%3A%22602797530%22
```

It can tell you a country and a carrier even if not correctly in my case thought. It says O2 but I have Vodafone because I moved while keeping the number. It’s followed by a series of Google dork search queries using the number. Sometimes you would expect even more phone variations with brackets and hyphens as it is common in the US for instance – (368) 500-1234. As I said it’s just a passive scanner.

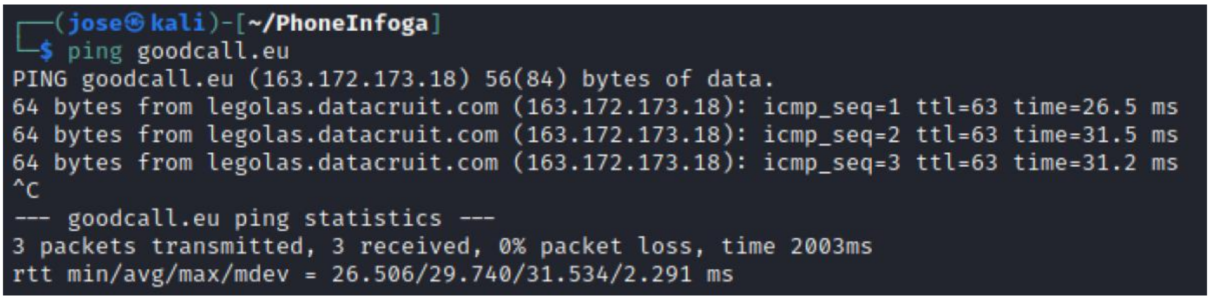
Nmap

Nmap is the very standard network scanner which I’ve been personally using for about 20 years. It belongs more to the group of system security tools rather than passive OSINT gathering tools but you can combine it with other already mentioned tools such as theHarvester, DataSploit and Photon to expand more on the found hosts, subdomains and IPs.

We can for example find out which OS is running the specific web address.

Nmap usually expects an IP address or a range or IP addresses rather than a DNS name such as www.GoodCall.eu. It's a bit lame but you can get the IP for example by a simple ping.

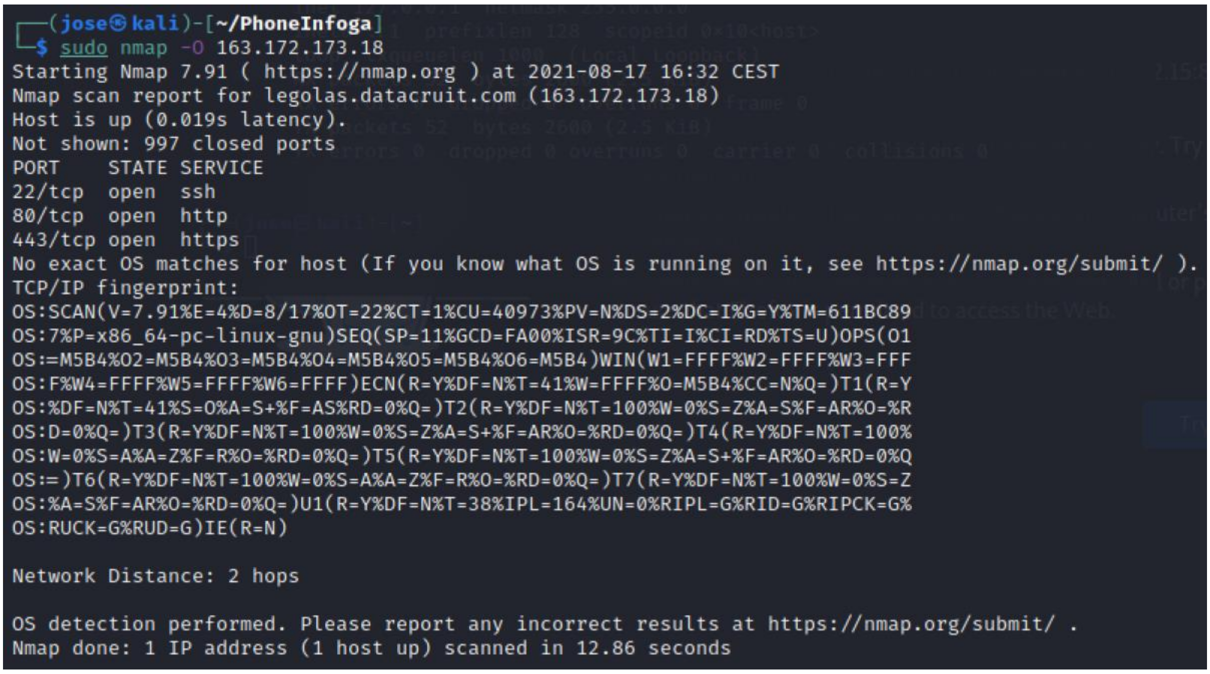
ping www.gooodcall.eu



```
┌──(jose㉿kali)-[~/PhoneInfoga]
└─$ ping goodcall.eu
PING goodcall.eu (163.172.173.18) 56(84) bytes of data.
64 bytes from legolas.datacruit.com (163.172.173.18): icmp_seq=1 ttl=63 time=26.5 ms
64 bytes from legolas.datacruit.com (163.172.173.18): icmp_seq=2 ttl=63 time=31.5 ms
64 bytes from legolas.datacruit.com (163.172.173.18): icmp_seq=3 ttl=63 time=31.2 ms
^C
--- goodcall.eu ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 26.506/29.740/31.534/2.291 ms
```

And then we run *nmap* with the root privileges so for example by using *sudo* command.

sudo nmap -O 163.172.173.18



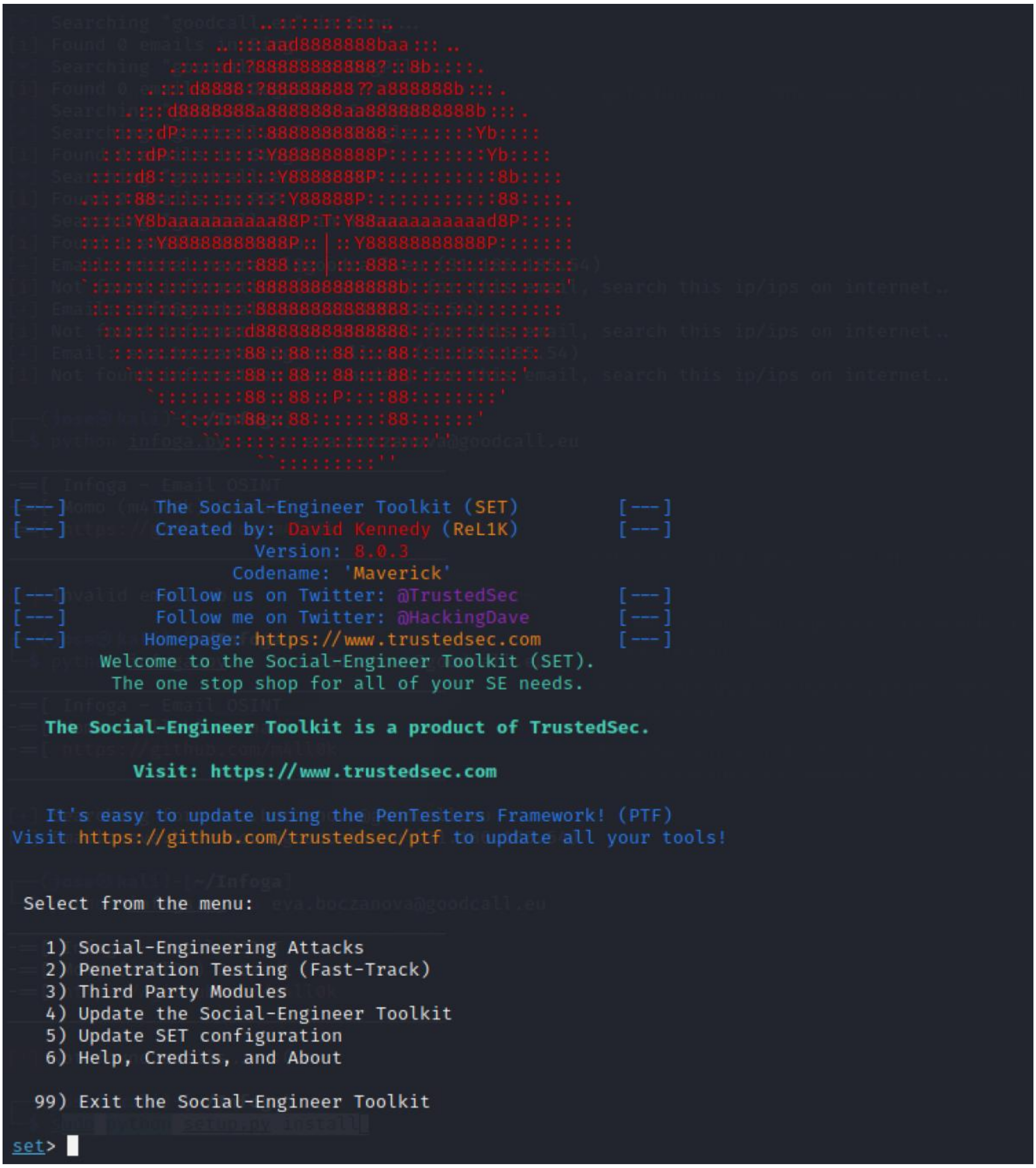
```
┌──(jose㉿kali)-[~/PhoneInfoga]
└─$ sudo nmap -O 163.172.173.18
Starting Nmap 7.91 ( https://nmap.org ) at 2021-08-17 16:32 CEST
Nmap scan report for legolas.datacruit.com (163.172.173.18)
Host is up (0.019s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE
22/tcp  open  ssh
80/tcp  open  http
443/tcp open  https
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.91%E=4%D=8/17%OT=22%CT=1%CU=40973%PV=N%DS=2%DC=I%G=Y%TM=611BC89
OS:7%P=x86_64-pc-linux-gnu)SEQ(SP=11%GCD=FA00%ISR=9C%TI=I%CI=RD%TS=U)OPS(O1
OS:=M5B4%O2=M5B4%O3=M5B4%O4=M5B4%O5=M5B4%O6=M5B4)WIN(W1=FFFF%W2=FFFF%W3=FFF
OS:F%W4=FFFF%W5=FFFF%W6=FFFF)ECN(R=Y%DF=N%T=41%W=FFFF%O=M5B4%CC=N%Q=)T1(R=Y
OS:%DF=N%T=41%S=O%A=S+%F=AS%RD=0%Q=)T2(R=Y%DF=N%T=100%W=0%S=Z%A=S%F=AR%O=%R
OS:D=0%Q=)T3(R=Y%DF=N%T=100%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)T4(R=Y%DF=N%T=100%
OS:W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T5(R=Y%DF=N%T=100%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q
OS:=)T6(R=Y%DF=N%T=100%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T7(R=Y%DF=N%T=100%W=0%S=Z
OS:%A=S%F=AR%O=%RD=0%Q=)U1(R=Y%DF=N%T=38%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%
OS:RUCK=G%RUD=G)IE(R=N)

Network Distance: 2 hops

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.86 seconds
```

It says that based on open protocols and footprints it cannot clearly state what OS is running there but we can see that it is some Linux OS from the footprint.

Beware that unlike some of other search engines such as Shodan which are passive scanners (the information is already preloaded in the search engine so you don't connect with the targeted server), *nmap* is doing the active search so you establish active connections between you and the target. The advantage is that you have really up-to-date data.

There are not only command line apps on Linux. You can use desktop applications (Creepy, Maltego) and some of the CLI apps have also web interface. For example, IVRE is a web app running on your localhost and can call CLI apps such as *nmap*. I will focus on them in a different publication or an article.

Also, there are many others command line apps you can use for OSINT on Linux. All or the vast majority of apps I described are supposed to be so-called passive recon tools. You can also use some active recon tools like Social Engineering Toolkit (SET) which you can also find on Kali Linux by default.



```
                 ..:::::::::..
             ..:::aad8888888baa:::..
          .::::d:?88888888888?::8b::::.
        .:::d8888:?88888888??a888888b:::.
      .:::d8888888a8888888aa8888888888b:::.
     ::::dP::::::::88888888888::::::::Yb::::
    ::::dP:::::::::Y888888888P:::::::::Yb::::
   ::::d8:::::::::::Y8888888P:::::::::::8b::::
  .::::88::::::::::::Y88888P::::::::::::88::::.
  :::::Y8baaaaaaaaaa88P:T:Y88aaaaaaaaaad8P:::::
  :::::::Y88888888888P::|::Y88888888888P:::::::
  ::::::::::::::::888:::|:::888::::::::::::::::
  `:::::::::::::::8888888888888b::::::::::::::'
   :::::::::::::::88888888888888::::::::::::::
    :::::::::::::d88888888888888:::::::::::::
     ::::::::::::88::88::88:::88::::::::::::
      `::::::::::88::88::88:::88::::::::::'
        `::::::::88::88::P::::88::::::::'
          `::::::88::88:::::::88::::::'
             ``:::::::::::::::::::''
                  ``:::::::::''

[---]        The Social-Engineer Toolkit (SET)         [---]
[---]        Created by: David Kennedy (ReL1K)         [---]
                      Version: 8.0.3
                    Codename: 'Maverick'
[---]        Follow us on Twitter: @TrustedSec         [---]
[---]        Follow me on Twitter: @HackingDave        [---]
[---]       Homepage: https://www.trustedsec.com       [---]
        Welcome to the Social-Engineer Toolkit (SET).
         The one stop shop for all of your SE needs.

   The Social-Engineer Toolkit is a product of TrustedSec.

           Visit: https://www.trustedsec.com

   It's easy to update using the PenTesters Framework! (PTF)
Visit https://github.com/trustedsec/ptf to update all your tools!


 Select from the menu:

   1) Social-Engineering Attacks
   2) Penetration Testing (Fast-Track)
   3) Third Party Modules
   4) Update the Social-Engineer Toolkit
   5) Update SET configuration
   6) Help, Credits, and About

  99) Exit the Social-Engineer Toolkit

set> 
```

You can run any apps on the web or Google Chrome plugins as well of course.

Sourcer in the Shell

You could get the idea why to use Linux and shell applications and scripts for various services you are normally used to running on the web.

When you get the ropes of the Linux commands, you can be really efficient in digging for the OSINT info. Usually, you have also more options which can use – for example to choose the format of the output. Or you edit the behavior by editing them in the case of scripts in Python and other scripting languages.

The next level is incorporating the Linux apps or their output into your own shell, Python or Perl scripts.

In the next issues I will focus on some other aspects of OSINT covering some specific searching verticals such as visual sourcing or going through deep, leaked and dark web data.

FOLLOW JOSÉ FOR MORE UPDATES

José Kadlec is a former ethical hacker, digital forensic examiner and hardcore Linux engineer who went head over into the talent sourcing industry utilizing his cross-field experience.

Based on the OSINT techniques co-founded a holding of companies *Datacruit*, *GoodCall* and *Recruitment Academy*. They made made it to the FT1000 as the 415th fastest growing company in Europe by Financial Times.

Follow José on: **LinkedIn** | **Twitter** | **Facebook** | **Instagram**

www.JoseKadlec.com

⚠ DISCLAIMER: In no event shall the author of this ebook be liable for any special, consequential, incidental or indirect damages whatsoever (including, without limitation, damages for loss of business profits, business interruption, loss of business information, or any other pecuniary loss or damage) arising out of the use of this product.