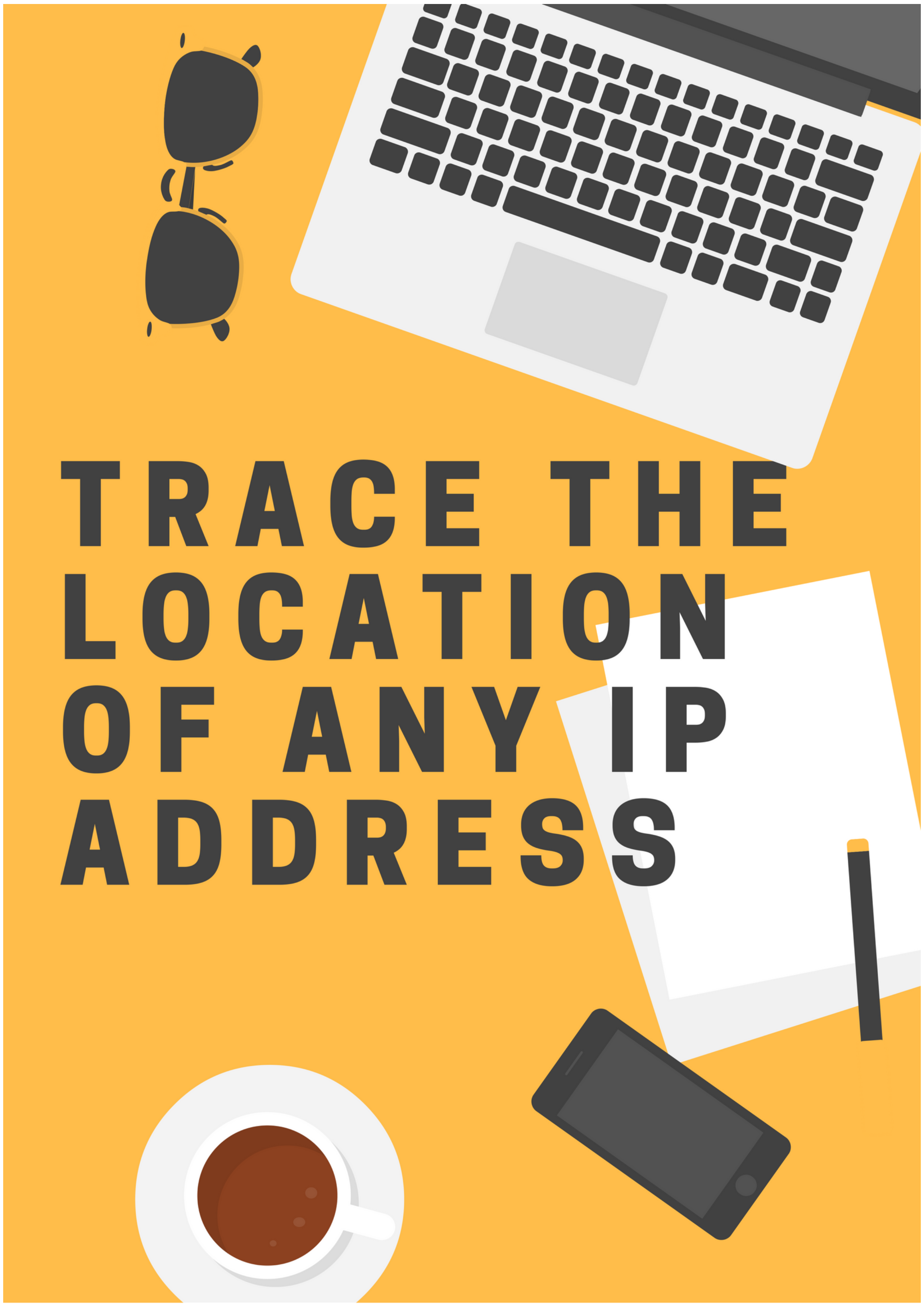


TRACE THE LOCATION OF ANY IP ADDRESS

Hello everyone! We're going to be demonstrating how to find and trace the location of any IP address! We'll start with a quick introduction to this topic, and then we'll jump right in.

I'm sure most of you know this already, but for those that don't, each network that is connected to the Internet is given one public IP address (this is due to the implication of NAT). This means that each user on the Internet has a unique address that doesn't belong to anyone else. This address is what allows us to communicate and access resources on the Internet, such as this very site!

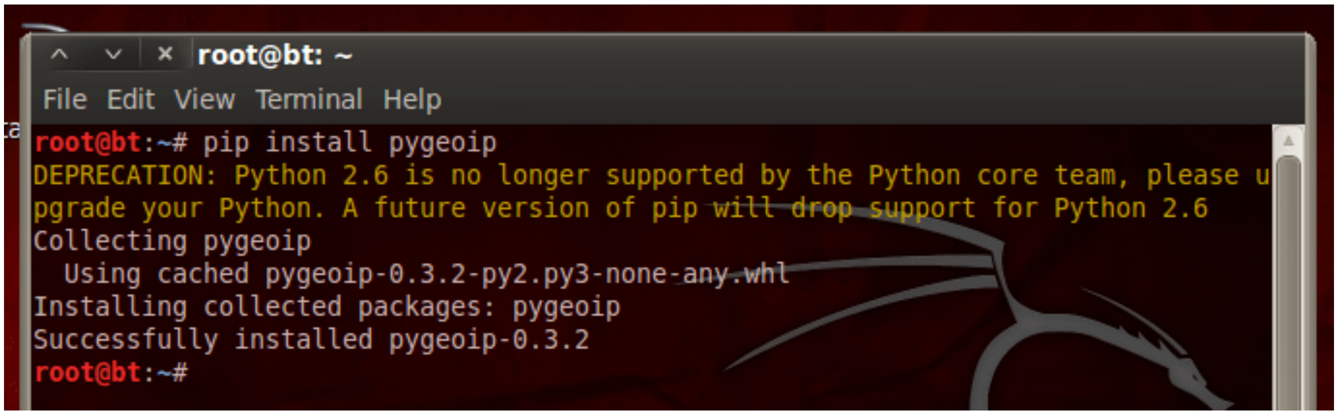
Now that we know the usage for IP over the Internet for the average user, we can explain how it applies to hackers. If you remember our article on hping3, you already know how important it is for hackers to hide their IP. In this article, we'll be demonstrating just how easy it is to track and trace an IP address using IP geolocation. It all comes down to this; if hackers leave their public IP address lying around the place, law enforcement **will** find it and they **will** track it. Once they find our IP, it'll only be a matter of days before they're breaking down the door!

Now that we've discussed the use of IP for both users and law enforcement, we can jump into learning how to trace an IP address for ourselves! First things first, let's discuss the forces that enable us to do this in the first place. There is a company by the name of MaxMind that maintains a database of the location of nearly every known IP address. In order to trace an IP using MaxMind we'd normally have to pay a fee in order to query their database. But our kind friends over at MaxMind also have a developer version of the database available for free download. The only real issue with this is, we don't have a way to read the developer database once we download it. Or do we?

This brings us to our second factor as play, code. Someone released a Python module named pygeoip. This module will allow us to query the database for almost any IP we want! Be warned, the results we will get are not exact. If we want pinpoint accuracy, we'll need to pay for MaxMind's service. But, if we want to track IPs for free, we'll have to sacrifice a bit of accuracy (a fair trade overall). Now that we know about the software we'll be using, we can begin the process of tracking an IP. So, let's get started!

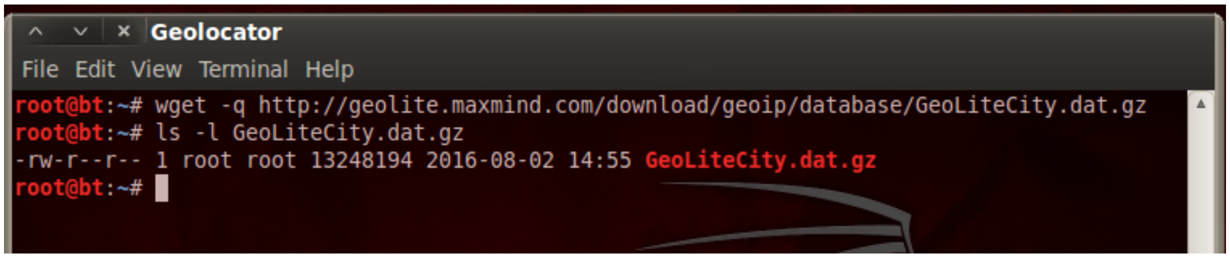
Step 1: Download and Install the Needed Files

Our fist step in tracing IPs is a fairly obvious one. We need to download the database and install the pygeoip module. We'll start by installing pygeoip, since it's the simpler of the two. All we need to do in order to install pygeoip is to use the Python package manager **pip** for an easy and quick installation. (If your distribution of Linux does not come with pip, you can install it via this command: **apt-get install pip**. If this method does not work, you can manually install pip by download and executing this script.) Let’s take a look at installing pygeoip via pip:



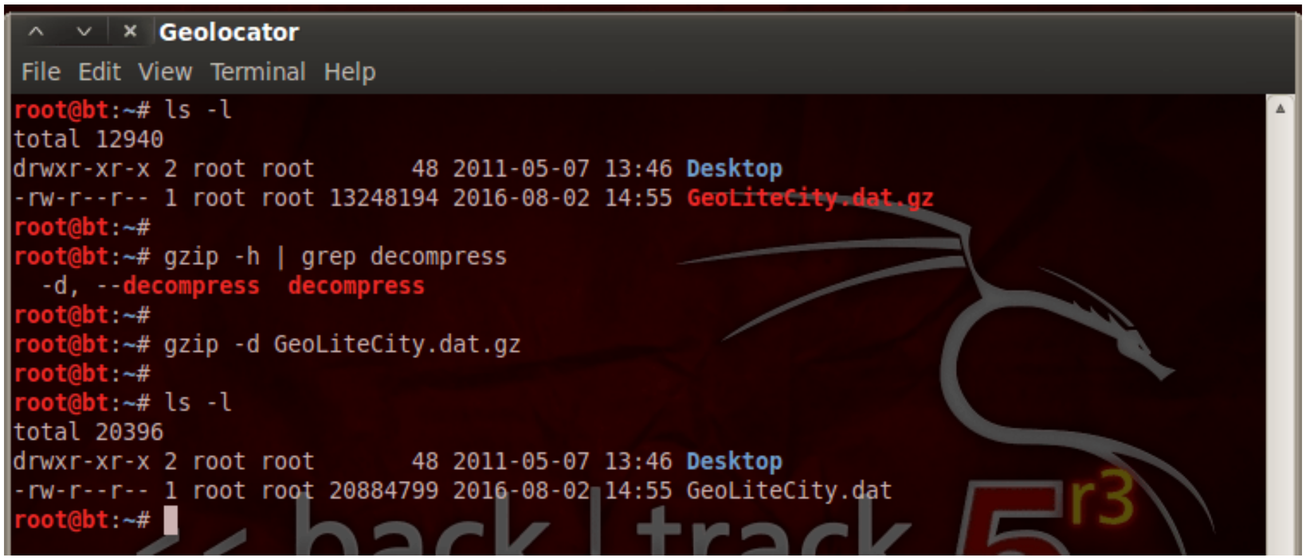
```
root@bt:~# pip install pygeoip
DEPRECATION: Python 2.6 is no longer supported by the Python core team, please u
pgrade your Python. A future version of pip will drop support for Python 2.6
Collecting pygeoip
  Using cached pygeoip-0.3.2-py2.py3-none-any.whl
Installing collected packages: pygeoip
Successfully installed pygeoip-0.3.2
root@bt:~#
```

Simple enough, right? Well here comes the harder part, getting our hands on the database. This is still a very simple process, albeit more complicated than installing pygeoip. First we need to download the compressed file using wget (I’ll put the link here to make it easier for you: http://geolite.maxmind.com/download/geoip/database/GeoLiteCity.dat.gz):



```
root@bt:~# wget -q http://geolite.maxmind.com/download/geoip/database/GeoLiteCity.dat.gz
root@bt:~# ls -l GeoLiteCity.dat.gz
-rw-r--r-- 1 root root 13248194 2016-08-02 14:55 GeoLiteCity.dat.gz
root@bt:~#
```

I’ve used the **-q** flag to hide all output in order to prevent output from spewing all over the place. Regardless, we now have a file named GeoLiteCity.dat.gz. For those of you that don’t know, a dat file is a file that contains raw data, and a gz file is a type of compressed file. So, in order to get at the dat file within, we need to decompress this file using the **gzip** command:

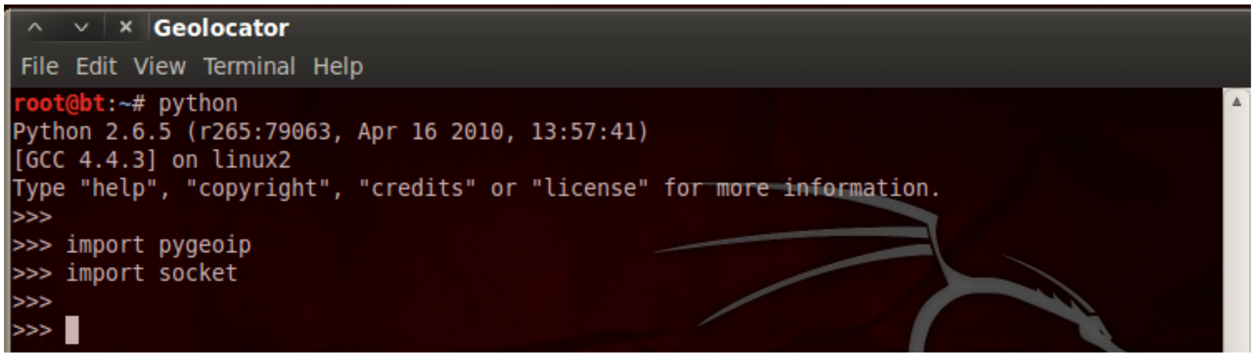


```
root@bt:~# ls -l
total 12940
drwxr-xr-x 2 root root       48 2011-05-07 13:46 Desktop
-rw-r--r-- 1 root root 13248194 2016-08-02 14:55 GeoLiteCity.dat.gz
root@bt:~#
root@bt:~# gzip -h | grep decompress
  -d, --decompress  decompress
root@bt:~#
root@bt:~# gzip -d GeoLiteCity.dat.gz
root@bt:~#
root@bt:~# ls -l
total 20396
drwxr-xr-x 2 root root       48 2011-05-07 13:46 Desktop
-rw-r--r-- 1 root root 20884799 2016-08-02 14:55 GeoLiteCity.dat
root@bt:~# 
```

We can see here that by using the **-d** flag with gzip, we've decompressed the file and now have GeoLiteCity.dat. We finally have the database that we need. Now that we have both of these downloaded and installed, we can finally get to tracking an IP!

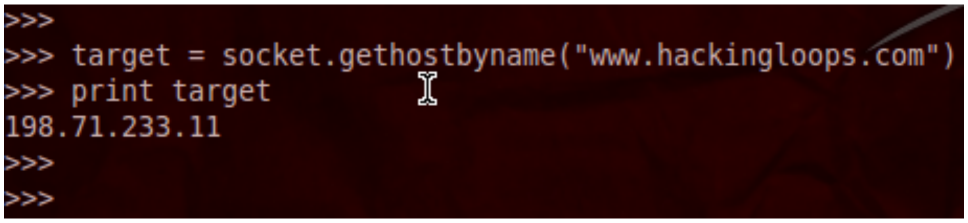
Step 2: Trace the IP using Pygeoip

Now that we have all that we need, we can use them to trace an IP. We'll be using a live Python interpreter to query the database through the command line. We're also going to need to import two modules to allow us to query the database. We're going to import pygeoip (which we just installed), and socket (that comes pre-packaged with Python). Let's go ahead and fire up our interpreter and import these modules:



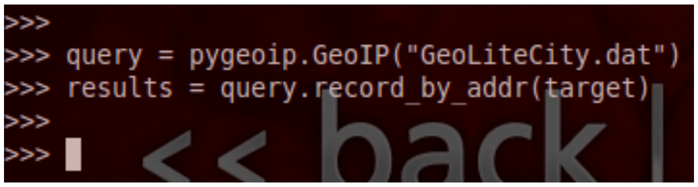
```
root@bt:~# python
Python 2.6.5 (r265:79063, Apr 16 2010, 13:57:41)
[GCC 4.4.3] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>>
>>> import pygeoip
>>> import socket
>>>
>>> 
```

Now that we have our modules, let's put them to good use! First things first, if we're going to track and IP address, we need an IP to track! We can use the socket module to find an IP address from a domain name. We're going to be tracking the IP of the one and only hackingloops.com! So, let's use the socket module to find the IP of www.hackingloops.com :



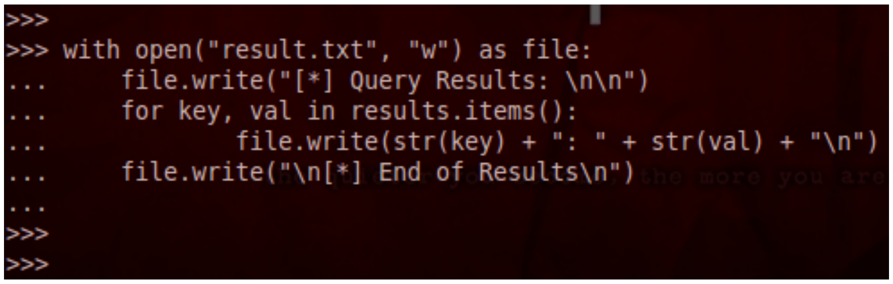
```
>>>
>>> target = socket.gethostbyname("www.hackingloops.com")
>>> print target
198.71.233.11
>>>
>>>
```

Now that we have the IP address we're going to track, we need to query the database for it. For this, we use the pygeoip module. This part is very simple, so let's see it:



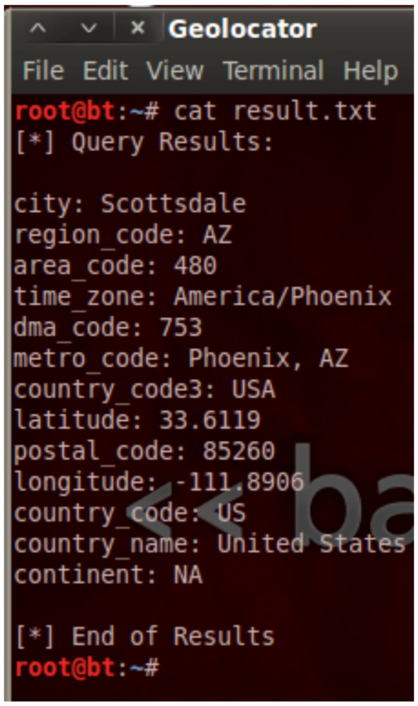
```
>>>
>>> query = pygeoip.GeoIP("GeoLiteCity.dat")
>>> results = query.record_by_addr(target)
>>>
>>> 
```

Here we can see that we've made a query and stored it under results. Now this part is a bit more complicated; we have the results right now, but we're going to write them to a text file so that we don't have to query more than once. Let's take a look at the code and then we'll break it down:



```
>>>
>>> with open("result.txt", "w") as file:
...     file.write("[*] Query Results: \n\n")
...     for key, val in results.items():
...             file.write(str(key) + ": " + str(val) + "\n")
...     file.write("\n[*] End of Results\n")
...
>>>
>>>
```

What we've done here is still simple, yet relatively complicated compared to what we did before. We've opened a new file named results.txt under the alias of "file." We then loop through the contents of the results and write them to the file along with some starting and ending text. Once this has executed we should have a new text file that has our results in it. Let's go ahead and use the **cat**command to read our results from this new file:



```
root@bt:~# cat result.txt
[*] Query Results:

city: Scottsdale
region_code: AZ
area_code: 480
time_zone: America/Phoenix
dma_code: 753
metro_code: Phoenix, AZ
country_code3: USA
latitude: 33.6119
postal_code: 85260
longitude: -111.8906
country_code: US
country_name: United States
continent: NA

[*] End of Results
root@bt:~#
```

There we have it! We can see by the contents of our results file above that www.hackingloops.com is hosted on servers somewhere near Scottsdale, Arizona. We did it!

Having this understanding of how to **IP Trace** can help us to avoid being tracked. It can also prove to be a very useful tool to have in the case that we end up needing to track down an attacker ourselves for whatever reason.