

# maxmind\_db\_ip\_geolocator.py

Original Post: [Python Basics for Hackers, Part 4: How to Find the Exact Location of any IP Address](#)

```
#!/usr/bin/python

#Hello fellow hackers! My name is Defalt
#I built a very basic version of this tool a long time ago and recently did a re-write
#The first re-write had some awkward usage of the argparse module, so this update is going to fix it
#Original version: http://pastebin.com/J5NLnThL
#This will query the MaxMind database to get an approximate geolocation of an IP address
#Happy hacking! -Defalt


import sys
import socket
import urllib
import gzip
import os

try:
    import pygeoip
except ImportError:
    print '[!] Failed to Import pygeoip'
    try:
        choice = raw_input('[*] Attempt to Auto-install pygeoip? [y/N] ')
    except KeyboardInterrupt:
        print '\n[!] User Interrupted Choice'
        sys.exit(1)
    if choice.strip().lower()[0] == 'y':
        print '[*] Attempting to Install pygeoip... ',
        sys.stdout.flush()
        try:
            import pip
            pip.main(['install', '-q', 'pygeoip'])
```

```

import pygeoip
print '[DONE]'
except Exception:
    print '[FAIL]'
    sys.exit(1)
elif choice.strip().lower()[0] == 'n':
    print '[*] User Denied Auto-install'
    sys.exit(1)
else:
    print '[!] Invalid Decision'
    sys.exit(1)

class Locator(object):
    def __init__(self, url=False, ip=False, datfile=False):
        self.url = url
        self.ip = ip
        self.datfile = datfile
        self.target = ''
    def check_database(self):
        if not self.datfile:
            self.datfile = '/usr/share/GeoIP/GeoLiteCity.dat'
        else:
            if not os.path.isfile(self.datfile):
                print '[!] Failed to Detect Specified Database'
                sys.exit(1)
            else:
                return
        if not os.path.isfile(self.datfile):
            print '[!] Default Database Detection Failed'
        try:
            choice = raw_input('[*] Attempt to Auto-install Database? [y/N] ')
        except KeyboardInterrupt:
            print '\n[!] User Interrupted Choice'
            sys.exit(1)
        if choice.strip().lower()[0] == 'y':
            print '[*] Attempting to Auto-install Database... ',
            sys.stdout.flush()
            if not os.path.isdir('/usr/share/GeoIP'):
                os.makedirs('/usr/share/GeoIP')
            try:
                urllib.urlretrieve('http://geolite.maxmind.com/download/geoip/database/GeoLiteCity.dat.gz',

```

```

/usr/share/GeoIP/GeoLiteCity.dat.gz')
except Exception:
    print '[FAIL]'
    print '[!] Failed to Download Database'
    sys.exit(1)
try:
    with gzip.open('/usr/share/GeoIP/GeoLiteCity.dat.gz', 'rb') as compressed_dat:
        with open('/usr/share/GeoIP/GeoLiteCity.dat', 'wb') as new_dat:
            new_dat.write(compressed_dat.read())
except IOError:
    print '[FAIL]'
    print '[!] Failed to Decompress Database'
    sys.exit(1)
os.remove('/usr/share/GeoIP/GeoLiteCity.dat.gz')
print '[DONE]\n'
elif choice.strip().lower()[0] == 'n':
    print '[!] User Denied Auto-Install'
    sys.exit(1)
else:
    print '[!] Invalid Choice'
    sys.exit(1)
def query(self):
    if not not self.url:
        print '[*] Translating %s: ' %(self.url),
        sys.stdout.flush()
        try:
            self.target += socket.gethostbyname(self.url)
            print self.target
        except Exception:
            print '\n[!] Failed to Resolve URL'
            return
        else:
            self.target += self.ip
        try:
            print '[*] Querying for Records of %s...\n' %(self.target)
            query_obj = pygeoip.GeoIP(self.datfile)
            for key, val in query_obj.record_by_addr(self.target).items():
                print '%s: %s' %(key, val)
            print '\n[*] Query Complete!'
        except Exception:
            print '\n[!] Failed to Retrieve Records'

```

```
    return
```

```
if __name__ == '__main__':
```

```
    import argparse
```

```
    parser = argparse.ArgumentParser(description='IP Geolocation Tool')
```

```
    parser.add_argument('--url', help='Locate an IP based on a URL', action='store', default=False, dest='url')
```

```
        parser.add_argument('-t', '--target', help='Locate the specified IP', action='store', default=False, dest='ip')
```

```
        parser.add_argument('--dat', help='Custom database filepath', action='store', default=False, dest='datfile')
```

```
    args = parser.parse_args()
```

```
    if ((not not args.url) and (not not args.ip)) or ((not args.url) and (not args.ip)):
```

```
        parser.error('invalid target specification')
```

```
    try:
```

```
        locate = Locator(url=args.url, ip=args.ip, datfile=args.datfile)
```

```
        locate.check_database()
```

```
        locate.query()
```

```
    except Exception:
```

```
        print '\n[!] An Unknown Error Occured'
```

---

Revision #1

Created 25 September 2022 10:29:58 by Admin

Updated 25 September 2022 10:33:04 by Admin