

OSINT tools for security auditing

Open Source Intelligence with python tools

José Manuel Ortega

@jmortegac



<http://jmortega.github.io>



EUROPYTHON

2016 Bilbao, 17-24 July

Ethical hacking with Python tools

JOSE MANUEL ORTEGA
@JMORTEGAC

https://github.com/jmortega/osint_tools_security_auditing

Shodan

TOR

emails

google+API

images

ip_map_position

links

maltego_python

mapping

metadata

panoramio

twitter

youtube

BuiltWith.py

GeoLite2-City.mmdb

README.md

censys_data.py

checkFullContactAPI.py

checkIpDetails.py

checkLinkedLinProfile.py

check_social_networks.py

github_repositories.py



Agenda

- **OSINT introduction**
- **Server information(Censys,Shodan)**
- **OSINT tools developed with python**
- **Geolocation,Metadata**
- **Twitter,Footprinting,FullContact**

OSINT

- **Define a specific target and data you wish to obtain**
- **Technical-Accounts,servers,services,software**
- **Social-Social Media,Email,Photos**
- **Physical-Address,Home IP address,Footprinting**
- **Logical-Network,Operational intelligence**

OSINT

- **GeoLocation**
- **IP address**
- **Email address**
- **Telephone Number**
- **Username in social network profiles**
- **Metadata information from images**
- **Server information & vulnerabilities**

Censys.io

[About](#)[Search](#)[Reports](#)[API](#)[Raw Data](#)[Search](#) ▾[IPv4 Snapshots](#)[Historical Data](#)

This dataset contains the latest information we know about each public host in the IPv4 address space. Each record describes a single host and matches the data that is available in the IPv4 search index. All protocols are updated at least weekly.

Latest Snapshot

Our last snapshot was taken at 2016-09-15 06:48:05.

Name	ipv4-2016-09-15.json.lz4
File Size	685.5 GB
SHA-256 Checksum	23769379fabdbc88e9ef121d13ef75dff2fdcbff19816b8b570beda07778399b

Censys.io

https://www.censys.io/api/v1/view/ipv4/ip_address

<https://www.censys.io/api/v1/view/websites/domain>

```
def get_censys_data(ip, domain):  
  
    if ip is not None:  
        res = requests.get(API_URL + "/view/ipv4/"+ip, auth=(UID,SECRET))  
  
    if domain is not None:  
        res = requests.get(API_URL + "/view/websites/"+domain, auth=(UID,SECRET))  
  
    if res.status_code != 200:  
        print("error occurred: %s" % res.json()["error"])  
        sys.exit(1)  
  
    print(json.dumps(res.json(), indent=4))  
  
    with open('data_censys.txt', 'w') as file:  
        json.dump(res.json(), file, ensure_ascii=False, indent=4)
```


Censys.io

```
python censys_data.py -d python.org
```

```
    "mail.python.org"  
  ],  
  "organization": [  
    "Python Software Foundation"  
  ],  
  "locality": [  
    "Beaverton"  
  ]  
},  
"issuer": {  
  "country": [  
    "IL"  
  ],  
  "organization": [  
    "StartCom Ltd."  
  ],  
  "common_name": [  
    "StartCom Class 2 Primary Intermediate Server CA"  
  ],  
  "organizational_unit": [  
    "Secure Digital Certificate Signing"  
  ]  
}
```

```
  "organization": "Rackspace Hosting",  
  "asn": 27357,  
  "rir": "unknown",  
  "description": "RACKSPACE - Rackspace Hosting,US",  
  "name": "RACKSPACE",  
  "routed_prefix": "104.130.0.0/18"  
},  
"updated_at": "2016-09-24T10:30:10+00:00",  
"location": {  
  "province": "Texas",  
  "registered_country_code": "US",  
  "postal_code": "78218",  
  "latitude": 29.4889,  
  "registered_country": "United States",  
  "country": "United States",  
  "longitude": -98.3987,  
  "country_code": "US",  
  "city": "San Antonio",  
  "continent": "North America",  
  "timezone": "America/Chicago"  
}
```

Shodan

Shodan Developers Blog View All...

SHODAN fomcat Explore Downloads Reports Enterprise Access Contact Us

The search engine for Buildings

Shodan is the world's first search engine for Internet-connected devices.

Create a Free Account Getting Started

Welcome

Shodan lets you search for devices that are connected to the Internet. And a Shodan account means you get more access, more features and the ability to check out the latest developments.

More Results

With a free Shodan account you can access more results!

Developer API

The Shodan API makes it easy to access the data from within your own scripts.

New Filters

Once you're logged in you have access to a lot more filters that help you find exactly what you're looking for.

Sign in with Shodan

Username

Password

Log in [Forgot your password?](#)

Sign in with

TWITTER FACEBOOK

Google WINDOWS LIVE

Shodan

```
import shodan
```

```
SHODAN_API_KEY = "insert your API key here"  
api = shodan.Shodan(SHODAN_API_KEY)
```

```
# Lookup the host  
host = api.host(hostname)  
  
# Print general info  
print ""  
        IP: %s  
        Organization: %s  
        Operating System: %s  
    "" % (host['ip_str'], host.get('org', 'n/a'), host.get('os', 'n/a'))  
  
# Print all banners  
for item in host['data']:  
    print ""Port: %s  
    Banner: %s"" % (item['port'], item['data'])
```

Shodan

```
aster/Shodan $ python demoShodanSearch.py -target fosdem.org
```

```
IP: 31.22.22.135  
Organization: Tigron BVBA  
Operating System: None
```

```
Port: 80
```

```
Banner: HTTP/1.1 200 OK
```

```
Server: nginx/1.10.1
```

```
Date: Sat, 28 Jan 2017 21:12:01 GMT
```

```
Content-Type: text/html
```

```
Content-Length: 612
```

```
Last-Modified: Sun, 25 Sep 2016 19:13:43 GMT
```

```
Connection: keep-alive
```

```
ETag: "57e821e7-264"
```

```
Accept-Ranges: bytes
```

```
Port: 873
```

```
Banner: @RSYNCD: 31.0
```


```
video Recorded videos from the FOSDEM conferences
```

```
@RSYNCD: FXTT
```

Shodan

- Checking data with ip address

<https://www.shodan.io/host/31.22.22.135>



31.22.22.135 www-public.fosdem.org

Country	Belgium
Organization	Tigron BVBA
ISP	Tigron BVBA
Last Update	2017-01-28T21:12:06.837628
Hostnames	www-public.fosdem.org
ASN	AS56837

Ports

80 443 873

Services

80 tcp http
nginx Version: 1.10.1
HTTP/1.1 200 OK
Server: nginx/1.10.1
Date: Sat, 28 Jan 2017 21:12:01 GMT
Content-Type: text/html
Content-Length: 612

Shodan CVE vulns

```
# Print vuln information
for item in host['vulns']:
    CVE = item.replace('!', '')
    print 'Vulns: %s' % item
    exploits = api.exploits.search(CVE)
    for item in exploits['matches']:
        if item.get('cve')[0] == CVE:
            print item.get('description')
```

Shodan Developer API

<https://developer.shodan.io/api>

REST API Documentation

The base URL for all of these methods is:

`https://api.shodan.io`

Shodan Methods

GET /shodan/host/{ip}

GET /shodan/host/count

GET /shodan/host/search

GET /shodan/host/search/tokens

GET /shodan/ports

GET /shodan/protocols

POST /shodan/scan

POST /shodan/scan/internet

GET /shodan/scan/{id}

GET /shodan/services

GET /shodan/query

GET /shodan/query/search

GET /shodan/query/tags

Recon-ng

- <https://bitbucket.org/LaNMaSteR53/recon-ng>
- **Open Source OSINT toolkit written in python**
- **Actively maintained**
- **Uses modules and saves all recollected information in databases**

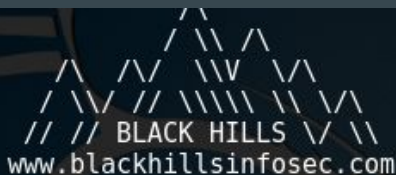
Recon-ng dependences

- dnspython - <http://www.dnspython.org/>
- dicttoxml - <https://github.com/quandyfactory/dicttoxml/>
- jsonrpclib - <https://github.com/joshmarshall/jsonrpclib/>
- lxml - <http://lxml.de/>
- slowaes - <https://code.google.com/p/slowaes/>
- XlsxWriter - <https://github.com/jmcnamara/XlsxWriter/>
- Mechanize
- PyPDF2
- sqlite3



Recon-ng modules

Sponsored by...



[recon-ng v4.8.1, Tim Tomes (@LaNMaSteR53)]

```
[76] Recon modules
[7] Reporting modules
[2] Import modules
[2] Exploitation modules
[2] Discovery modules
```

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

```
Discovery
```

```
-----
```

```
discovery/info_disclosure/cache_snoop
discovery/info_disclosure/interesting_files
```

```
Exploitation
```

```
-----
```

```
exploitation/injection/command_injector
exploitation/injection/xpath bruter
```

Recon-ng modules

Recon-ng / modules /

- ..
- discovery/info_disclosure
- exploitation/injection
- import
- recon
- reporting

Recon-ng / modules / recon /

- ..
- companies-contacts
- companies-multi
- contacts-contacts
- contacts-credentials
- contacts-domains
- contacts-profiles
- credentials-credentials
- domains-contacts
- domains-credentials/pwnedlist

Recon-ng modules

```
Description:
Leverages the freegeoip.net API to geolocate a host by IP address. Updates
'hosts' table with
the results.

Options:
  Name          Current Value          Required  Description
  -----
  SERVERURL     http://freegeoip.net         yes       overwrite server url (e.g. for
  l installations)
  SOURCE        default                       yes       source of input (see 'show i
  r details')

Source Options:
  default      SELECT DISTINCT ip_address FROM hosts WHERE ip_address I
  ULL
  <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:
  * Allows up to 10,000 queries per hour by default. Once this limit is r
  all requests will
  result in HTTP 403, forbidden, until the quota is cleared.

[recon-ng][default][freegeoip] > set source 31.22.22.135
SOURCE => 31.22.22.135
[recon-ng][default][freegeoip] > run
[*] 31.22.22.135 - 50.85,4.35 - Belgium
[recon-ng][default][freegeoip] > █
```

Recon-ng subdomains

```
class Module(BaseModule):
    meta = {
        'name': 'HackerTarget Lookup',
        'author': 'Michael Henriksen (@michenriksen)',
        'description': 'Uses the HackerTarget.com API to find host names. Updates the \'hosts\' table with the results.',
        'query': 'SELECT DISTINCT domain FROM domains WHERE domain IS NOT NULL',
    }

    def module_run(self, domains):
        for domain in domains:
            self.heading(domain, level=0)
            url = 'https://api.hackertarget.com/hostsearch/'
            payload = {'q': domain}
            resp = self.request(url, payload=payload)
            if resp.status_code is not 200:
                self.error('Got unexpected response code: %i' % resp.status_code)
                continue
            if resp.text == '':
                self.output('No results found.')
                continue
            for line in resp.text.split("\n"):
                line = line.strip()
                if line == '':
                    continue
                host, address = line.split(",")
                self.add_hosts(host=host, ip_address=address)
```

Recon-ng Shodan API

```
-----  
FOSDEM.ORG  
-----
```

```
[*] Searching Shodan API for: hostname:fosdem.org  
[*] [port] 31.22.22.135 (80) - www-public.fosdem.org  
[*] [host] www-public.fosdem.org (31.22.22.135)  
[*] [port] 51.15.48.19 (9100) - streambackend3.video.fosdem.org  
[*] [host] streambackend3.video.fosdem.org (51.15.48.19)  
[*] [port] 31.22.22.130 (143) - apeiron.fosdem.org  
[*] [host] apeiron.fosdem.org (31.22.22.130)  
[*] [port] 31.22.22.132 (5432) - phronesis.fosdem.org  
[*] [host] phronesis.fosdem.org (31.22.22.132)  
[*] [port] 51.15.49.18 (80) - streambackend1.video.fosdem.org  
[*] [host] streambackend1.video.fosdem.org (51.15.49.18)  
[*] [port] 51.15.42.23 (9100) - control0.video.fosdem.org  
[*] [host] control0.video.fosdem.org (51.15.42.23)  
[*] [port] 2001:67c:1808::4 (5432) - phronesis.fosdem.org  
[*] [host] phronesis.fosdem.org (2001:67c:1808::4)  
[*] [port] 31.22.22.135 (873) - www-public.fosdem.org  
[*] [host] www-public.fosdem.org (31.22.22.135)  
[*] [port] 51.15.50.139 (22) - streamfrontend1.video.fosdem.org  
[*] [host] streamfrontend1.video.fosdem.org (51.15.50.139)  
[*] [port] 163.172.142.127 (80) - crap.video.fosdem.org  
[*] [host] crap.video.fosdem.org (163.172.142.127)  
[*] [port] 51.15.49.176 (9100) - streambackend0.video.fosdem.org  
[*] [host] streambackend0.video.fosdem.org (51.15.49.176)  
[*] [port] 51.15.49.18 (9100) - streambackend1.video.fosdem.org  
[*] [host] streambackend1.video.fosdem.org (51.15.49.18)  
[*] [port] 51.15.49.176 (80) - streambackend0.video.fosdem.org  
[*] [host] streambackend0.video.fosdem.org (51.15.49.176)
```

The harvester

<https://github.com/laramies/theHarvester>

```
Aplicaciones ▾ Lugares ▾ Terminal ▾ dom 01:59 1 es ▾
root@kali: ~
Archivo Editar Ver Buscar Terminal Ayuda
* Edge-Security Research *
* cmartorella@edge-security.com *
*****
VBOXADDITIONS_
5.0.4.102546
Usage: theharvester options

-d: Domain to search or company name
-b: data source: google, googleCSE, bing, bingapi, pgp,
    linkedin, google-profiles, people123, jigsaw,
    twitter, googleplus, all

-s: Start in result number X (default: 0)
-v: Verify host name via dns resolution and search for virtual hosts
-f: Save the results into an HTML and XML file
-n: Perform a DNS reverse query on all ranges discovered
-c: Perform a DNS brute force for the domain name
-t: Perform a DNS TLD expansion discovery
-e: Use this DNS server
-l: Limit the number of results to work with(bing goes from 50 to 50 results,
-h: use SHODAN database to query discovered hosts
    google 100 to 100, and pgp doesn't use this option)

Examples:
theharvester -d microsoft.com -l 500 -b google
theharvester -d microsoft.com -b pgp
theharvester -d microsoft -l 200 -b linkedin
theharvester -d apple.com -b googleCSE -l 500 -s 300

root@kali:~#
```

The harvester modules

```
asksearch.py
bingsearch.py
bingsearch.pyc
DNS
dnssearch.py
dnssearch.pyc
dnssearch-threads.py
dogpilesearch.py
dogpilesearch.pyc
exaleadsearch.py
exaleadsearch.pyc
googleCSE.py
googleCSE.pyc
googleplussearch.py
googleplussearch.pyc
googlesearch.py
googlesearch.pyc
googlesets.py
googlesets.pyc
__init__.py
__init__.pyc
IPy.py
IPy.pyc
jigsaw.py
jigsaw.pyc
linkedinsearch.py
linkedinsearch.pyc
people123.py
people123.pyc
pgpsearch.py
pgpsearch.pyc
shodan
shodansearch.py
shodansearch.pyc
twittersearch.py
twittersearch.pyc
yandexsearch.py
yandexsearch.pyc
```


Python modules

- `httplib`
- `socket`
- `requests`
- `shodan`

```
import string
import httplib
import sys
import os
from socket import *
import re
import getopt

try:
    import requests
except:
    print "Request library not found,
    sys.exit()

from discovery import *
from lib import htmlExport
from lib import hostchecker
```

The harvester

```
[-] Searching in Bing:
```

```
root@kali:~# theharvester -d pycon.org -l 500 -b google
```

```
*****
```

```
*
```

```
* THE HARVESTER *
```

```
* TheHarvester Ver. 2.6  
* Coded by Christian Martorella  
* Edge-Security Research  
* cmartorella@edge-security.com
```

```
*****
```

```
[-] Searching in Google:
```

```
Searching 0 results...  
Searching 100 results...  
Searching 200 results...  
Searching 300 results...  
Searching 400 results...  
Searching 500 results...
```

```
[+] Emails found:
```

```
-----  
contacto2016@es.pycon.org  
info@pycon.org  
swc@za.pycon.org
```

```
[+] Hosts found in search engines:
```

```
-----
```

```
[-] Resolving hostnames IPs...
```

```
148.251.113.227:www.pycon.org  
144.76.246.116:2016.es.pycon.org  
69.164.212.224:za.pycon.org  
151.101.60.223:us.pycon.org  
82.166.0.152:il.pycon.org  
151.101.60.133:na.pycon.org  
106.187.52.176:tw.pycon.org  
144.76.246.116:2015.es.pycon.org  
205.147.96.46:in.pycon.org  
188.166.192.28:ua.pycon.org  
52.50.6.31:cz.pycon.org  
94.23.84.75:es.pycon.org  
87.204.59.165:pl.pycon.org  
106.187.52.176:Tw.pycon.org  
158.69.9.197:ar.pycon.org
```

OSR framework










- `pip install osrframework`
- Developed in python 2.7
- Integrates with maltego transforms
- <https://pypi.python.org/pypi/osrframework/0.13.2>
- <https://github.com/i3visio/osrframework>



OSR python modules

- **BeautifulSoup**
- **Requests**
- **Mechanize**
- **pyDNS**→resolving name servers
- **python-whois**→to recover the whois info from a domain
- **tweepy**→for connecting with Twitter API
- **Skype4Py**→ for connecting with Skype API
- **Python-emailahoy**→for checking email address
- **Multiprocessing**→import Process, Queue, Pool

OSR python scripts

 alias_generator.py	Added osrfconsole script to control de utilities in the framework to ...
 domainfy.py	Check information against DNS instead of making HTTP GET queries
 entify.py	Added osrfconsole script to control de utilities in the framework to ...
 enumeration.py	Added osrfconsole script to control de utilities in the framework to ...
 mailfy.py	Added osrfconsole script to control de utilities in the framework to ...
 osrfconsole.py	Fix issue #166 by adding a split to information in PLATFORMS
 phonefy.py	Added osrfconsole script to control de utilities in the framework to ...
 searchfy.py	Added osrfconsole script to control de utilities in the framework to ...
 usufy.py	Added osrfconsole script to control de utilities in the framework to ...

OSR python scripts

```
python searchfy.py -q "fosdem" -p "twitter"
```

i3visio_uri	i3visio_alias	i3visio_platform
http://twitter.com/ruby_fosdem	ruby_fosdem	Twitter
http://twitter.com/PerlFosdem	PerlFosdem	Twitter
http://twitter.com/FosdemBar	FosdemBar	Twitter
http://twitter.com/yaloki	yaloki	Twitter
http://twitter.com/fosdemlive	fosdemlive	Twitter
http://twitter.com/fosdem13home	fosdem13home	Twitter
http://twitter.com/PythonFOSDEM	PythonFOSDEM	Twitter
http://twitter.com/FosdemBot	FosdemBot	Twitter
http://twitter.com/go_fosdem	go_fosdem	Twitter
http://twitter.com/PHP_FOSDEM	PHP_FOSDEM	Twitter

OSR python scripts

```
python usufy.py -n "fosdem"
```

i3visio_uri	i3visio_alias	i3visio_platform
http://favstar.fm/users/fosdem	fosdem	Favstar
http://www.instagram.com/fosdem	fosdem	Instagram
http://www.slideshare.net/fosdem	fosdem	Slideshare
http://es.badoo.com/fosdem	fosdem	Badoo
https://gytorrents.com/user/fosdem/	fosdem	Gytorrents
http://www.klout.com/fosdem	fosdem	Klout
https://www.facebook.com/fosdem	fosdem	Facebook
http://twitter.com/fosdem	fosdem	Twitter
http://web.tv/user/fosdem	fosdem	Webtv
http://www.metacafe.com/channels/fosdem	fosdem	Metacafe
https://github.com/fosdem	fosdem	Github
http://twicsy.com/u/fosdem	fosdem	Twicsy
http://foroCompraventa.com/member.php?username=fosdem	fosdem	ForoCompraventa
http://www.nubelo.com/perfil/fosdem	fosdem	Nubelo

OSR python scripts

checkIfEmailWasHacked.py	Added Tor search platform to searchfy.py
checkIfHashIsCracked.py	Added Tor search platform to searchfy.py
checkInSkype.py	Checked that mailfy.py work on Linux.
checkIpDetails.py	Added Tor search platform to searchfy.py
checkIpFromAlias.py	Checked that mailfy.py work on Linux.
checkPhoneDetails.py	Added Tor search platform to searchfy.py
getBitcoinAddressDetails.py	Added Tor search platform to searchfy.py

```
python checkIpDetails.py -q fosdem.org
```

```
try:
    apiURL = "http://ip-api.com/json/" + query

    # Accessing the ip-api.com RESTful API
    data = urllib2.urlopen(apiURL).read()

    # Reading the text data onto python structures
    apiData = json.loads(data)

    # i3visio structure to be returned
    jsonData = []
```

```
[
  {
    "attributes": [],
    "type": "i3visio.location.city",
    "value": "Zaventem"
  },
  {
    "attributes": [],
    "type": "i3visio.location.postalcode",
    "value": "1930"
  },
  {
    "attributes": [
      {
        "attributes": [],
        "type": "i3visio.text",
        "value": "BE"
      }
    ],
    "type": "i3visio.location.country",
    "value": "Belgium"
  },
  {
    "attributes": [
      {
        "attributes": [],
        "type": "i3visio.text",
        "value": "Flanders"
      }
    ],
    "type": "i3visio.location.province",
    "value": "VLG"
  },
  {
    "attributes": [],
    "type": "i3visio.text",
    "value": "Tigron BVBA"
  },
  {

```


SpiderFoot-modules




- Python 2.7
- BeautifulSoup
- DNSPython
- Socks
- Socket
- SSL
- CherryPy
- M2MCrypto
- Netaddr
- pyPDF

📁 adblockparser	Added Adblock parser module.
📁 bs4	Spidering to utilise BeautifulSoup parsing and adjusted configuration.
📁 dns	Added the DNSPython library.
📁 exifread	Identify image meta data and extract software used from images, offic...
📁 gexf	PyGEXF added
📁 ispell	Identifying names in content.
📁 metapdf	Extracting docx, xlsx, pptx and pdf meta data.
📁 openxmllib	Extracting docx, xlsx, pptx and pdf meta data.
📁 phonenumbers	Support for identifying phone numbers.
📁 pyPdf	Extracting docx, xlsx, pptx and pdf meta data.
📁 pythonwhois	Pythonwhois external module added.
📁 stem	Stem license.
📄 __init__.py	Module re-shuffle.
📄 socks.py	TOR integration support.

SpiderFoot-data sources

Source	Location	Notes
abuse.ch	http://www.abuse.ch	Various malware trackers.
AdBlock	https://easylst-downloads.adblockplus.org/easylst.txt	AdBlock pattern matches
AlienVault	https://reputation.alienvault.com	AlienVault's IP reputation database.
Autoshun.org	http://www.autoshun.org	Blacklists.
AVG Site Safety Report	http://www.avgthreatlabas.com	Site safety checker.
Bing	http://www.bing.com	Scraping but future version to also use API.
Blocklist.de	http://lists.blocklist.de	Blacklists.
Checkusernames.com	http://www.checkusernames.com	Look up username availability on popular sites.
DNS	Your configured DNS server.	Defaults to your local DNS but can be configured to whatever IP address you supply SpiderFoot.
DomainTools	http://www.domaintools.com	
DroneBL	http://www.dronebl.org	
Facebook	http://www.facebook.com	Scraping but future version to also use API.
FreeGeolP	http://freegeolp.net	
Google	http://www.google.com	Scraping but future version to also use API.
Google+	http://plus.google.com	Scraping but future version to also use API.
Google Safe Browsing	http://www.google.com/safebrowsing	Site safety checker.
LinkedIn	http://www.linkedin.com	Scraping but future version to also use API.
malc0de.com	http://malc0de.com	Blacklists.
malwaredomainlist.com	http://www.malwaredomainlist.com	Blacklists.

SpiderFoot-Results

 SpiderFoot New Scan Scans Settings

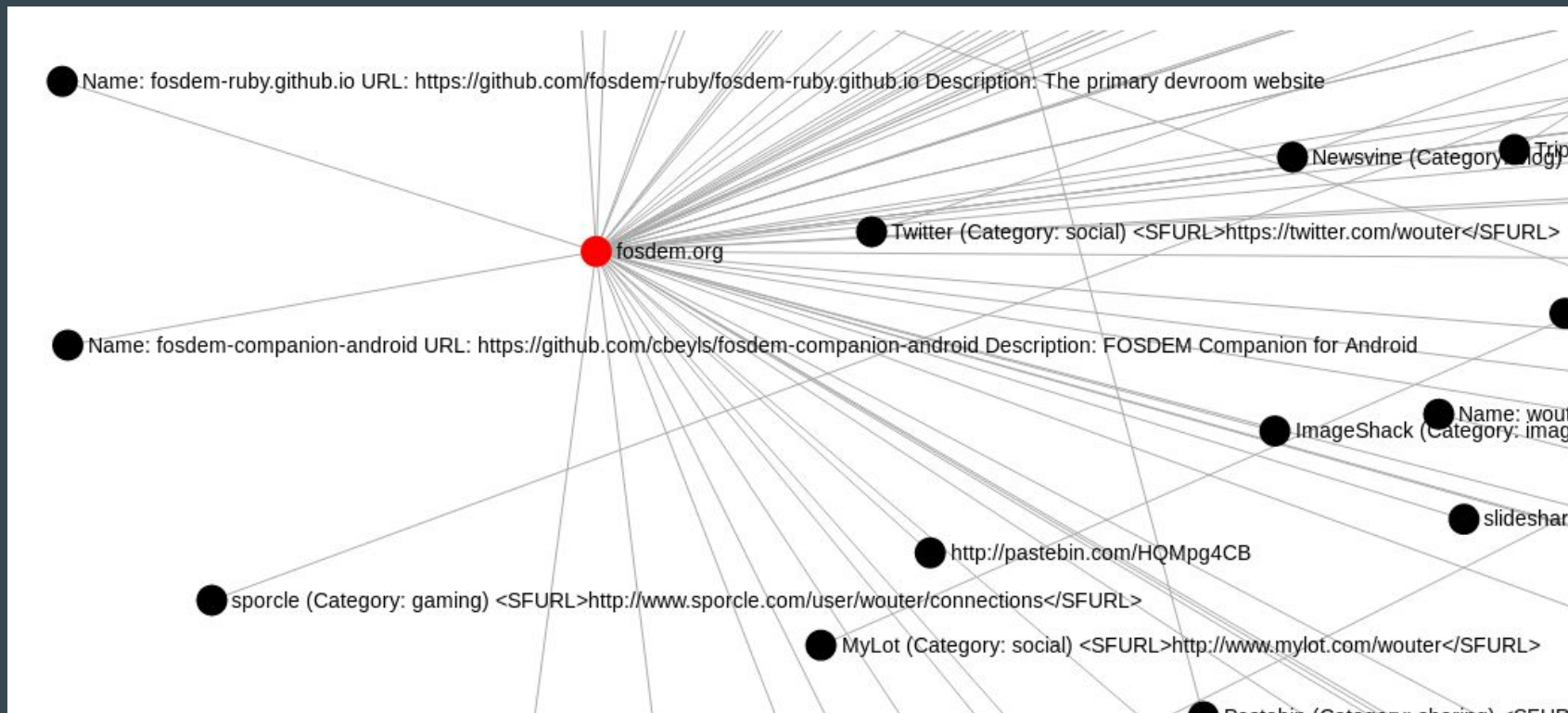
fosdem

Status Browse Graph Scan Settings Log

Browse > Public Code Repository

<input type="checkbox"/>	Data Element	
<input type="checkbox"/>	Name: avdyk.github.io URL: https://github.com/avdyk/avdyk.github.io Description: Personal website	
<input type="checkbox"/>	Name: chri URL: https://github.com/mikanyman/chri Description: Cultural Heritage Research Infrastructure	
<input type="checkbox"/>	Name: chris URL: https://github.com/beyondyuefei/chris Description: chris是一个支持JDBC层数据库读写分离的框架	

SpiderFoot-Results



Github repositories

```
GITHUB_URL_REPOS = 'https://api.github.com/repos'
GITHUB_URL_USERS = 'https://api.github.com/users'

import argparse
import requests
import json

def search_repositories(author, search_for):
    url = "%s/%s/repos" %(GITHUB_URL_USERS, author)|
    print("Searching Repositories URL: %s" %url)
    result = requests.get(url)
    if(result.ok):
        repo_info = json.loads(result.text or result.content)
        print("Github repository info for: %s" %author)
        result = "No result found!"
        keys = []
        for value in repo_info:
            for key,value2 in value.items():#python 2-->iteritems()
                if str(search_for)in str(key):
                    result = value
                    return result

if __name__ == '__main__':
    parser = argparse.ArgumentParser(description='Github search')
    parser.add_argument('--author', action="store", dest="author",required=True)
    parser.add_argument('--repo', action="store", dest="repo",required=False)
    parser.add_argument('--search_for', action="store",dest="search_for", default='owner', required=False)
    given_args = parser.parse_args()
    if given_args.repo is not None:
        result = search_repository(given_args.author, given_args.repo,given_args.search_for)
    else:
        result = search_repositories(given_args.author,given_args.search_for)
```

Github repositories

```
Github repository info for: fosdem
Got result for 'owner'...
issues_url => https://api.github.com/repos/FOSDEM/frontdesk-t-shirt_tracker/issues{/number}
deployments_url => https://api.github.com/repos/FOSDEM/frontdesk-t-shirt_tracker/deployments
stargazers_count => 0
forks_url => https://api.github.com/repos/FOSDEM/frontdesk-t-shirt_tracker/forks
mirror_url => None
subscription_url => https://api.github.com/repos/FOSDEM/frontdesk-t-shirt_tracker/subscriptions
notifications_url => https://api.github.com/repos/FOSDEM/frontdesk-t-shirt_tracker/notifications
collaborators_url => https://api.github.com/repos/FOSDEM/frontdesk-t-shirt_tracker/collaborators
updated_at => 2016-12-05T17:01:34Z
private => False
pulls_url => https://api.github.com/repos/FOSDEM/frontdesk-t-shirt_tracker/pulls{/number}
issue_comment_url => https://api.github.com/repos/FOSDEM/frontdesk-t-shirt_tracker/issues/{number}/comments{/number}
labels_url => https://api.github.com/repos/FOSDEM/frontdesk-t-shirt_tracker/labels{/name}
has_wiki => True
full_name => FOSDEM/frontdesk-t-shirt_tracker
```

Extract Metadata

- PDF→PyPDF2,PDFMiner
- Images→Pillow,pyexiv2(python 2.7),gexiv2(python 3)

```
[+] Metadata for file: images\img.jpg
Metadata: 42016 - Value: 2BF3A9E97BC886678DE12E6EB8835720
Metadata: YResolution - Value: (300, 1)
Metadata: ResolutionUnit - Value: 2
Metadata: Copyright - Value: Frank Noort
Metadata: Artist - Value: Frank Noort
Metadata: Make - Value: Canon
Metadata: GPSInfo - Value: {'Lat': 32.07874722222222, 'Lng': -131.46757777777778}
Metadata: XResolution - Value: (300, 1)
Metadata: ExifOffset - Value: 146
Metadata: ExifVersion - Value: 0220
Metadata: DateTimeOriginal - Value: 2002:10:28 11:05:09
Metadata: Model - Value: Canon EOS-5
Metadata: DateTime - Value: 2008:03:09 22:00:01
Metadata: Software - Value: Adobe Photoshop CS2 windows
```

GeoLocation

<http://dev.maxmind.com/geoip/geoip2/geolite2/>

```
import geoip2
import geoip2.database
```

```
def getGeo(self, ip):
    response = ''
    if not os.path.exists('GeoLite2-City.mmdb'):
        msg = "[*] GeoLite2-City.mmdb is not found ..."
        logging.error(msg)
    else:
        try:
            reader = geoip2.database.Reader('GeoLite2-City.mmdb')
            response = reader.city(ip)
        except:
            # *** need fixing exception messages ***
            msg = "[*] AddressNotFoundError for ip: %s" % (ip)
            logging.error(msg)
    return response
```

```
python GeoIP.py --target 8.8.8.8
```

```
[*] city: Mountain View
[*] region_code: CA
[*] area_code: 650
[*] time_zone: America/Los_Angeles
[*] dma_code: 807
[*] metro_code: San Francisco, CA
[*] country_code3: USA
[*] latitude: 37.3845
[*] postal_code: 94040
[*] longitude: -122.0881
[*] country_code: US
[*] country_name: United States
[*] continent: NA
[*] city: Mountain View
[*] region_code: CA
[*] area_code: 650
```


FootPrinting tools

- **Orb(Python 2.x)**
 - <https://github.com/epsylon/orb>
 - python-whois - Python module for retrieving WHOIS information
 - python-dnspython - DNS toolkit for Python
 - python-nmap - Python interface to the Nmap port scanner

- **InstaRecon(Python 2.x)**
 - <https://github.com/vergl4s/instarecon>
 - Dnspython,ipaddress
 - ipwhois,python-whois
 - requests,shodan

InstaRecon

```
python instarecon.py -v -s <SHODAN_API_KEY> python.org
```

```
# _____ Scanning python.org _____ #  
[-] WARNING: PTR lookup failed for 23.253.135.79 - NXDOMAIN  
[*] Domain: python.org  
  
[*] IPs & reverse DNS:  
23.253.135.79  
  
[*] NS records:  
ns1.p11.dynect.net  
    208.78.70.11 - ns1.p11.dynect.net  
ns2.p11.dynect.net  
    204.13.250.11 - ns2.p11.dynect.net  
ns3.p11.dynect.net  
    208.78.71.11 - ns3.p11.dynect.net  
ns4.p11.dynect.net  
    204.13.251.11 - ns4.p11.dynect.net  
  
[*] MX records:  
mail.python.org  
    188.166.95.178 - mail.python.org
```

InstaRecon

```
status.python.org
  52.35.72.202 - ec2-52-35-72-202.us-west-2.compute.amazonaws.com
  54.149.249.224 - ec2-54-149-249-224.us-west-2.compute.amazonaws.com
  54.148.61.28 - ec2-54-148-61-28.us-west-2.compute.amazonaws.com
  https://status.python.org/
testpypi.python.org
  151.101.12.175
  https://testpypi.python.org/
warehouse.python.org
  151.101.60.175
  https://warehouse.python.org/
  https://warehouse.python.org/project/whitenoise/
wiki.python.org
  140.211.10.69 - virt-y8pzvf.psf.osuosl.org
  https://wiki.python.org/jython
  https://wiki.python.org/jython/JythonDeveloperGuide
  https://wiki.python.org/jython/JythonFaq
  https://wiki.python.org/jython/JythonUsers
  https://wiki.python.org/jython/PackageScanning
  https://wiki.python.org/jython/PoiExample
  https://wiki.python.org/jython/ReadlineSetup
  https://wiki.python.org/jython/RoadMap
  https://wiki.python.org/jython/whyJython
```

Python modules

- **BeautifulSoup** for parsing web information
- **Requests,urllib3** for synchronous requests
- **Asyncio,aiohttp** for asynchronous requests
- **Robobrowser,Scrapy** for web crawling
- **PyGeolP,geoip2,geojson** for GeoLocation
- **python-twitter,tweepy** for connecting with twitter
- **Shodan** for obtain information for servers
- **DNSPython,netaddr** for resolving ip address



Wig-WebApp Information gatherer

```
root@kali:~/wig# ./wig.py --help
```

```
usage: wig.py [-h] [-l INPUT_FILE] [-n STOP_AFTER] [-a] [-m] [-u]
             [--no_cache_load] [--no_cache_save] [-N] [--verbosity]
             [--proxy PROXY] [-w OUTPUT_FILE]
             [url]
```

WebApp Information Gatherer

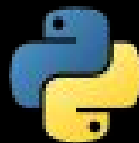
positional arguments:

url The url to scan e.g. http://example.com

optional arguments:

-h, --help show this help message and exit
-l INPUT_FILE File with urls, one per line.
-n STOP_AFTER Stop after this amount of CMSs have been detected. Default:
1
-a Do not stop after the first CMS is detected
-m Try harder to find a match without making more requests
-u User-agent to use in the requests
--no_cache_load Do not load cached responses
--no_cache_save Do not save the cache for later use
-N Shortcut for --no_cache_load and --no_cache_save
--verbosity, -v Increase verbosity. Use multiple times for more info
--proxy PROXY Tunnel through a proxy (format: localhost:8080)
-w OUTPUT_FILE File to dump results into (JSON)

```
root@kali:~/wig#
```



python 3

Wig-WebApp Information gatherer

<https://github.com/jekyc/wig>

```
wig - WebApp Information Gatherer
```

```
Redirected to https://fosdem.org
```

```
Continue? [Y|n]:y
```

```
Scanning https://fosdem.org...
```

SITE INFO

IP	Title
31.22.22.135	FOSDEM 2017 - Home

VERSION

Name	Versions	Type
nginx	1.10.1	Platform
FreeBSD	10 11	OS
OpenBSD	6.0	OS
openSUSE	tumbleweed	OS

SUBDOMAINS

Name	Page Title	IP
http://m.fosdem.org:80	FOSDEM	31.22.22.130
http://mail.fosdem.org:80	FOSDEM	31.22.22.130
https://mail.fosdem.org:443	FOSDEM	31.22.22.130

```
Time: 164.6 sec
```

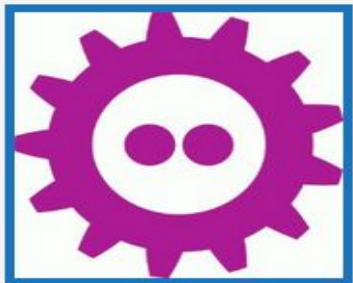
```
Urls: 610
```

```
Fingerprints: 40401
```

Tinfoleak-fosdem

```
python tinfoleak.py -u fosdem -i -s --sdate 2016-01-01 --hashtags --mentions --meta --media media --social --top 10 --conv -o report.html
```

tinfoleak



FOSDEM

FOSDEM is a free and non-commercial event organized by the community, for the community.

Followers: 7,928 | Following: 187 | Likes: 235

Tweets: 1,215 (0.36 tweets/day)

Screen Name: fosdem

Account Created at: 11/28/2007

Verified: False

Twitter ID: 10680142

URL: <http://fosdem.org/>

Location: Europe

Time Zone: Brussels

Geo enabled: False

Listed count: 351

Language: en

Tinfoleak-python dependences

- `import tweepy`→**Twitter API library for Python**
- `from PIL import Image, ExifTags, ImageCms`→**metadata from images**
- `import pyexiv2`→**metadata from images**
- `import urllib2`→**requests**
- `from OpenSSL import SSL`
- `from jinja2 import Template, Environment, FileSystemLoader`→**report**



Tinfoleak-parameters

optional arguments:

```
-h, --help          show this help message and exit
-v, --version       show program's version number and exit
-t TWEETS, --tweets TWEETS
                    number of tweets to be analysed (default: 200)
-i, --info          general information about a user
-s, --sources       show the client applications used to publish the
                    tweets
--sdate SDATE       filter the results for this start date (format:
                    yyyy/mm/dd)
--edate EDATE       filter the results for this end date (format:
                    yyyy/mm/dd)
--stime STIME       filter the results for this start time (format:
                    HH:MM:SS)
--etime ETIME       filter the results for this end date (format:
                    HH:MM:SS)
--hashtags          show info about hashtags included in tweets
--mentions          show info about user mentions
--meta              show metadata information from user images
--media [D]         [no value]: show user images and videos, [d]: download
                    user images to "username" directory
--friend FRTENDNAME show friendship with the FRTENDNAME user
```

Tinfoleak

HASHTAG DETAIL

Date (since)	Date (until)	RT's	Likes	Count	#Hashtag
01/13/2016	01/31/2016	159	122	14	#FOSDEM
01/31/2016	01/31/2016	22	7	1	#fodem
01/30/2016	01/31/2016	14	5	2	#line71
01/31/2016	01/31/2016	9	5	1	#funwithpublictransport
01/30/2016	01/30/2016	7	10	1	#hsbxl
01/30/2016	01/30/2016	7	10	1	#ByteNight
01/24/2016	01/30/2016	64	32	3	#FOSDEM2016
01/27/2016	01/27/2016	24	10	1	#opensource

TOP MENTIONS

Date (since)	Date (until)	RT's	Likes	Count	Name	Mention
01/26/2016	02/05/2016	191	177	12	FOSDEM	@fosdem
01/30/2016	02/11/2016	171	93	8	Peter Van Eynde	@pvaneynd
02/02/2016	11/28/2016	1	5	5	JJ Merelo	@jjmerelo
01/04/2016	11/28/2016	0	0	4	Alessio Fattorini	@ale_fattorini
01/26/2016	01/26/2016	0	0	4	Simon Phipps	@webmink
01/26/2016	01/26/2016	0	0	4	christian	@csgui
11/27/2016	12/07/2016	1	4	3	Kenneth Hoste	@kehoste
01/29/2016	01/30/2016	0	4	3	Iefred	@Iefred
01/04/2016	01/04/2016	1	0	3	Rob Bosch	@robb_nl
01/04/2016	01/04/2016	0	0	3	Su-Shee	@sheeshee

Tinfoleak-get auth configuration

```
class Configuration():
    """Configuration information"""
    # -----
    def __init__(self):
        try:
            # Read tinfoleak configuration file ("tinfoleak.conf")
            config = ConfigParser.RawConfigParser()
            config.read('tinfoleak.conf')
            self.color = config.get('colors', 'INFO')
            self.color_hdr = config.get('colors', 'HEADER')
            self.color_fun = config.get('colors', 'FUNCTION')

            CONSUMER_KEY = config.get('Twitter OAuth', 'CONSUMER_KEY')
            CONSUMER_SECRET = config.get('Twitter OAuth', 'CONSUMER_SECRET')
            ACCESS_TOKEN = config.get('Twitter OAuth', 'ACCESS_TOKEN')
            ACCESS_TOKEN_SECRET = config.get('Twitter OAuth', 'ACCESS_TOKEN_SECRET')
```

Tinfoleak-Geolocation

```
# -----  
def set_geofile_information(self, tweet, user):  
    try:  
        tweet_geo = 0  
        place = ""  
        geo = ""  
  
        # Get place from tweet  
        if tweet.place:  
            place = tweet.place.name.encode('utf-8')  
  
        # Get coordinates from tweet  
        if tweet.geo:  
            geo = tweet.geo['coordinates']  
            tweet_geo = 1
```

Tinfoleak-Geolocation

```
if tweet.geo:
    sgeo = tweet.geo['coordinates']
    add = 1
    lat = str(sgeo[0])[:str(sgeo[0]).find(".") + 4]
    lon = str(sgeo[1])[:str(sgeo[1]).find(".") + 4]
    location = "[" + lat + ", " + lon + "]"
    for i in range(1, 20 - len(location)):
        location += " "
    location = location + "\t" + splace
```

FullContact API

- We know we have a valid email address
- What other profiles are associated with this address?
- Go to fullcontact.com for an API key.....



FullContact API



Version 2 API

INTRODUCTION

PERSON API

COMPANY API

CARD READER API

EMAIL API

Email API Overview

Detect Disposable Email
Addresses

Disposable Email API Diagram

NAME API

LOCATION API

BATCH PROCESS

ACCOUNT STATS

LIBRARIES

Version 3 API

BETA

OVERVIEW

METHODS

Email API Overview

Do you run a service in which you would like to reduce the number of anonymous subscribers helping to reduce userbase contamination? This API allows you to identify disposable email addresses or one time use email addresses so that you can take action at the time of user signup. The API detects known domains associated with disposable email addresses, and in addition detects sub addressing for domains where the behavior is known.

Detect Disposable Email Addresses

Use the disposable method for identifying email addresses that either use sub addressing or are associated with known one time use or disposable email addresses. NOTE: Please take efforts to not expose your private api key in client side applications of this API.

JSON

```
curl -H"X-FullContact-APIKey:$your_key" "https://api.fullcontact.com/v2/email/disposable.json?email=joe+tag@sharklasers.com"
```

XML

```
curl -H"X-FullContact-APIKey:$your_key" "https://api.fullcontact.com/v2/email/disposable.xml?email=joe+tag@sharklasers.com"
```

FullContact API

```
def get_fullcontact(email):  
    api_key = 'a82ad9009f6b1b1'  
    base_url = 'https://api.fullcontact.com/v2/person.json'  
    payload = {'email':email, 'apiKey':api_key}  
    resp = requests.get(base_url, params=payload)  
    if resp.status_code == 200:  
        # parse contact information  
        if 'contactInfo' in resp.json():
```


FullContact API

```
python checkFullContactAPI.py -e gvanrossum@gmail.com
```

```
Guido van Rossum - gvanrossum@gmail.com  
Staff Engineer at Dropbox  
Employee at Google  
Employee at Elemental Security  
Employee at Zope Corporation  
Employee at CNRI  
Employee at BeOpen.com  
Employee at CWI  
Employee at SARA  
San Francisco Bay Area  
Github - https://github.com/gvanrossum  
Twitter - https://twitter.com/gvanrossum  
Quora - http://www.quora.com/Guido-van-Rossum  
Gravatar - https://gravatar.com/gvanrossum  
Flickr - https://www.flickr.com/people/gvanrossum  
Yelp - http://gvanrossum.yelp.com  
Blogger - http://blogger.com/profile/12821714508588242516  
GooglePlus - https://plus.google.com/u/0/115212051037621986145  
Klout - http://klout.com/gvanrossum  
LinkedIn - https://www.linkedin.com/pub/guido-van-rossum/0/756/4a0  
Confidence: 88%
```

FullContact API

```
python checkFullContactByDomain.py -d fosdem.org
```

```
    "links" : [ {
      "url" : "http://fosdem.org/2017/rss.xml",
      "label" : "rss"
    }, {
      "url" : "http://fosdem.org/2016/rss.xml",
      "label" : "rss"
    } ]
  },
  "socialProfiles" : [ {
    "bio" : "FOSDEM is a free and non-commercial event organized by the community for the community. The goal is to provide Free Software developers and communities a place to meet to: * get in touch with other developers and projects; * be informed about the Free Software and Open Source world; * attend interesting talks and presentations held in large conference rooms by Free Software project leaders and committers on various topics; and * to promote the development and the benefits of Free Software and Open Source Software. Participation and attendance is totally free, though the organization gratefully accepts donationals and sponsorships.",
    "typeId" : "linkedincompany",
    "typeName" : "LinkedIn",
    "url" : "https://www.linkedin.com/company/fosdem",
    "username" : "fosdem",
    "id" : "349978"
  } ],
  "traffic" : {
    "topCountryRanking" : [ {
      "rank" : 466884,
      "locale" : "us"
    } ],
    "ranking" : [ {
      "rank" : 631870,
      "locale" : "global"
    }, {
      "rank" : 466884,
      "locale" : "us"
    } ]
  }
}
```

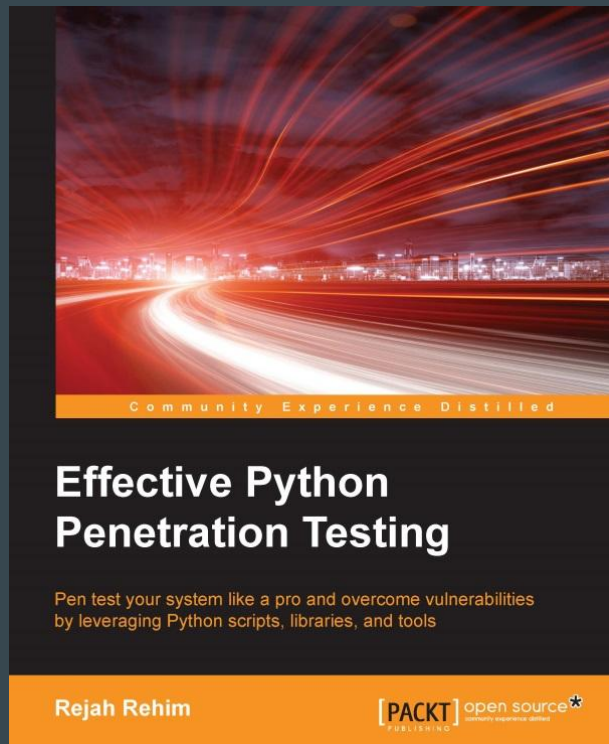
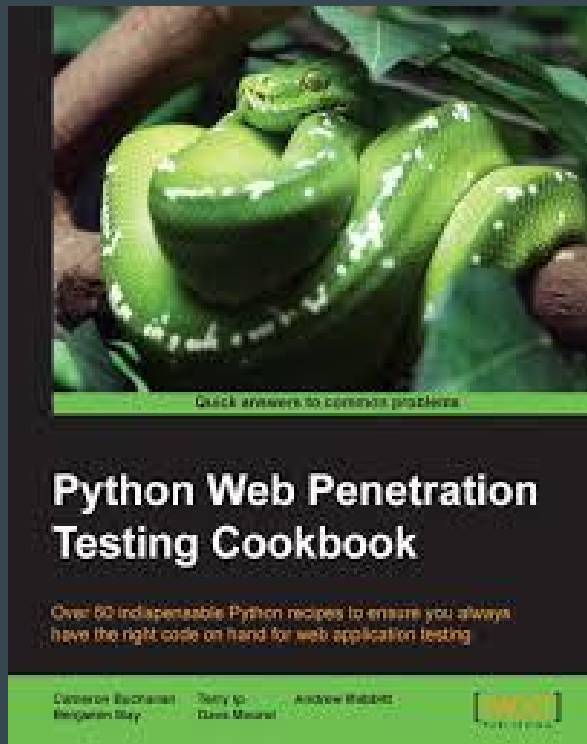
Kali Linux



References

- <http://osintframework.com>
- <https://sourceforge.net/projects/spiderfoot>
- <http://www.edge-security.com/theharvester.php>
- <https://developer.shodan.io/api>
- <http://www.clips.ua.ac.be/pattern>
- http://www.pentest-standard.org/index.php/PTES_Technical_Guidelines#OSINT
- <http://www.vicenteaguileradiaz.com/tools>
- https://github.com/automatingosint/osint_public
- <http://www.automatingosint.com/blog/>

Books



Thanks!

@jmortegac

