Osint with DataSploit

Author = Behrouz Mansoori

Email : mr.mansoori@yahoo.com

Detecting Behavioral Personas with Email OSINT

In this article we'll learn how to associate a behavioral persona to an email account with the help of an OSINT technique.

For those of you who are not very familiar with the concepts involved, OSINT stands for "Open-Source INTelligence" and basically refers to collecting publicly available information on the Internet for further analysis. On the other hand, a behavioral persona represents what a user does rather than what he or she is, for example: "Bob is a married lawyer who uses Dropbox, LinkedIn and Facebook, and he has a Gmail account."

A Privacy Issue Concerning Emails

Nowadays, it is not very uncommon that web apps will leak the fact that their users' usernames or email accounts exist in their database. Let's say for example they are okay assuming there's no risk that anybody will know that bob@genericdomain.com is a user of a certain extramarital dating site.

Think a bit about it. A malicious attacker can easily guess if a given email account is registered into such vulnerable applications by just interacting with the authentication mechanism.

They will just try to log in with a wrong email or password and see how it responds.

Use Case	Message
Nonexistent email	No account was found for this email
Existent email with wrong password	The password is not correct, please try again

These illustrative examples literally tell us if this or that user is registered into the system. However, not every potentially vulnerable application responds that way; some times, they leak the information by sending the browser an HTTP status code other than 200, or by redirecting to another page from which the leaking information can be extracted.

Why Would Apps Do That?

A usability design decision is often the reason why, and probably to a certain extent, big players like Microsoft or Google had an impact on how developers and designers are writing apps today.

Service	Use Case	Message
Outlook	Nonexistent user	That Microsoft account doesn’t exist
Gmail	Nonexistent user	Couldn’t find your Google Account

Web devs might well think:

If they did it why can't we did it?

But be aware of hackers. Even though the messages above are user-friendly, and certainly help understand users what exactly is going on, it may not be the right thing to do in terms of web security, especially if the site is particularly targeted to an audience.

Companies should protect the data of apps addressed to people of a particular religion, race, health condition, sexual orientation or political ideology. The same thing goes for dating sites, gambling sites, and any other type of app that could reveal private behavioral patterns of its users.

More specifically, here are some threats about leaking emails:

- Users' accounts can potentially be accessed by brute force
- Valid email accounts can be extracted for spamming purposes
- OSINT can be performed to deduce a user's persona from the sites they are registered

What Is the Solution?

From a developer's perspective the answer is not to provide any specific clues about what is running under the hood, as it is shown below.

Use Case	Message
Nonexistent email	The credentials provided are not valid
Existent email with wrong password	The credentials provided are not valid

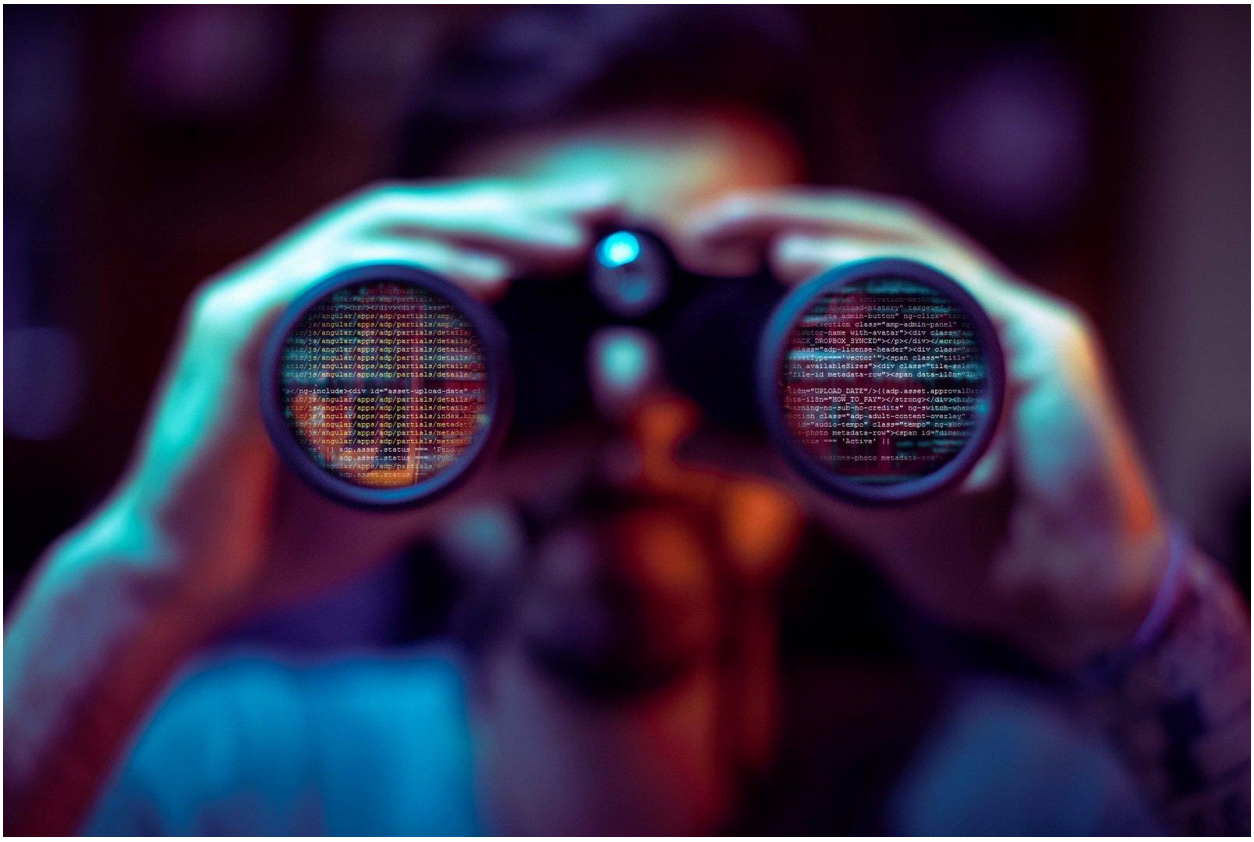
Twitter is an example where usability is sacrificed for the sake of security:

Service	Use Case	Message
Twitter	Nonexistent user	The email and password that you entered did not match our records

The solution is really simple actually: Users just shouldn't be able to log in the app when providing a wrong email/password pair. Error messages must be displayed in such a way that nobody can know what's wrong with what went wrong.

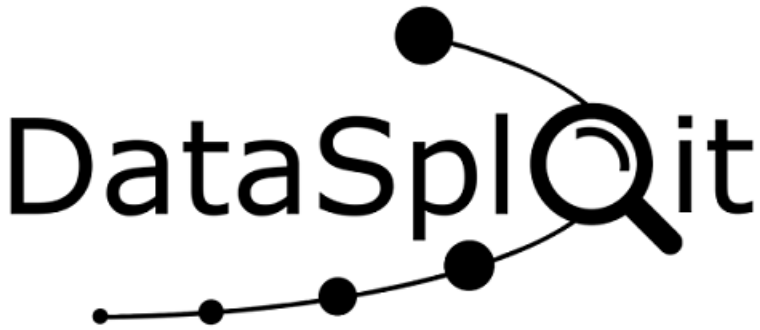
Introducing Email OSINT

From small to medium-sized custom apps written in PHP, JavaScript or Python, to widely used websites all over the world, including open source ERPs and CRMs. There's a plethora of websites of all sizes and shapes leaking emails from their databases.



Remember, hackers with enough time and motivation can do some manual pen testing and write their own tools with the clear purpose of identifying emails with behavioral personas.

Automated OSINT with DataSploit



Regarding automated penetration testing tools, DataSploit is a Python OSINT framework that performs automated OSINT on domains, emails, usernames and phones. You can use DataSploit as something to pull from in Python apps or as a standalone tool in the command line.

Here is how to install DataSploit:

```
git clone https://github.com/datasploit/datasploit /etc/datasploit
cd /etc/datasploit/
pip install -r requirements.txt
mv config_sample.py config.py
```

If the installation goes okay you will get this output on your console:

```
python datasploit.py
True

Single target or file input required to run
```

```
usage: datasploit.py [-h] [-i SINGLE_TARGET] [-f FILE_TARGET] [-a] [-q]
                     [-o OUTPUT]

  _____/ /_____ _ / /_ _____ _ _____ _____   / /____   (_)/ /_
 / __  // __ `// __// __ `// ___// __ \ / // __ \ / // __/
/ /_/ // /_/ // /_ / /_/ /(__  )/ /_/ // // /_/ // // /_
\__,_/ \__,_/ \__/ \__,_//____// .___//_/ \____//_/ \__/
                              /_/

          Open Source Assistant for #OSINT
             www.datasploit.info

optional arguments:
  -h, --help            show this help message and exit
  -i SINGLE_TARGET, --input SINGLE_TARGET
                        Provide Input
  -f FILE_TARGET, --file FILE_TARGET
                        Provide Input
  -a, --active          Run Active Scan attacks
  -q, --quiet           Run scans in automated manner accepting default
                        answers
  -o OUTPUT, --output OUTPUT
                        Provide Destination Directory

            Connect at Social Media: @datasploit
```

This is how to do basic email OSINT in the command line out of the box:

```
python emailOsint.py bob@genericdomain.org
[-] Skipping Hacked because it is marked as disabled.

---> Finding Paste(s)..

[-] google_cse_key not configured
```

[-] google_cse_key and google_cse_cx not configured. Skipping paste(s) search.
Please refer to http://datasploit.readthedocs.io/en/latest/apiGeneration/.

---> Searching Scribd Docs

---> Checking breach status in HIBP (@troyhunt)

Pwned at 4 Instances

Title: Adobe
BreachDate: 2013-10-04
PwnCount: 152445165
Description: In October 2013, 153 million Adobe accounts were breached with each containing an internal ID, username, email, encrypted password and a password hint in plain text. The password cryptography was poorly done and many were quickly resolved back to plain text. The unencrypted hints also disclosed much about the passwords adding further to the risk that hundreds of millions of Adobe customers already faced.
DataClasses: Email addresses, Password hints, Passwords, Usernames

Title: Dropbox
BreachDate: 2012-07-01
PwnCount: 68648009
Description: In mid-2012, Dropbox suffered a data breach which exposed the stored credentials of tens of millions of their customers. In August 2016, they forced password resets for customers they believed may be at risk. A large volume of data totalling over 68 million records was subsequently traded online and included email addresses and salted hashes of passwords (half of them SHA1, half of them bcrypt).
DataClasses: Email addresses, Passwords

Title: LinkedIn
BreachDate: 2012-05-05
PwnCount: 164611595
Description: In May 2016, LinkedIn had 164 million email addresses and passwords exposed. Originally hacked in 2012, the data remained out of sight until being offered for sale on a dark market site 4 years later. The passwords in the breach were stored as SHA1 hashes without salt, the vast majority of which were quickly cracked in the days following the release of the data.
DataClasses: Email addresses, Passwords

Title: MySpace
BreachDate: 2008-07-01
PwnCount: 359420698
Description: In approximately 2008, MySpace suffered a data breach that exposed almost 360 million accounts. In May 2016 the data was offered up for sale on the "Real Deal" dark market website and included email addresses, usernames and SHA1 hashes of the first 10 characters of the password converted to lowercase and stored without a salt. The exact breach date is unknown, but analysis of the data suggests it was 8 years before being made public.
DataClasses: Email addresses, Passwords, Usernames

---> Checking Fullcontact..

[-] fullcontact_api not configured

[-] Full-Contact API Key not configured. Skipping Fullcontact Search.
Please refer to http://datasploit.readthedocs.io/en/latest/apiGeneration/.

---> Searching Clearbit

[-] clearbit_apikey not configured
[False, 'INVALID_API']

[-] Clearbit API Key not configured. Skipping Clearbit Search.
Please refer to http://datasploit.readthedocs.io/en/latest/apiGeneration/.

---> Basic Email Check(s)..

[-] mailboxlayer_api not configured
[-] mailboxlayer_api not configured

[-] MailBoxLayer_API Key not configured. Skipping basic checks.
Please refer to http://datasploit.readthedocs.io/en/latest/apiGeneration/.

---> Searching Slideshare

---> Searching Whoismind for associated domains

Fair enough! With a very basic query we could learn that bob@genericdomain.org uses Adobe, Dropbox, LinkedIn and MySpace.

Note that DataSploit heavily relies on third-party services and APIs for reporting. The information above is retrieved through a public service named haveibeenpwned.com (HIBP) which checks if a particular email account is been compromised in a data breach.

As you can see, this is the HIBP section in DataSploit's report:

---> Checking breach status in HIBP (@troyhunt)

For a more detailed analysis in the other sections it is necessary to get an API key with the corresponding service provider supported by DataSploit:

- https://account.shodan.io/register
- https://www.censys.io/register
- https://dashboard.clearbit.com/signup
- https://hunter.io/users/sign_up
- https://dashboard.fullcontact.com/signup
- https://console.developers.google.com/
- https://www.zoomeye.org/accounts/register

Then, the environment has to be set up accordingly in the config.py file.

Conclusion

Today we brought some awareness on information leaking and OSINT since there are still plenty of applications out there informing about which email account exists in their databases.

Some of them are about dating or gambling, just to give some examples, or are exposed to audiences of a particular religion, sexual orientation or political ideology. Hackers with time and motivation can take advantage of the vulnerability with manual pen testing to an extent that it is even possible to detect behavioral personas.

Also, DataSploit is an OSINT framework that allows to automate the process of email research with emailOsint.py. More specifically, we showed how easy it is to figure out a bunch of widely-known sites being used by a target email address.

mr.mansoori@yahoo.com

Instagram.com/Behrouz_mansoori