

Passive reconnaissance

Intro

Physical:



Location Information

- Satellite images
- Drone recon
- Building layout (badge readers, break areas, security, fencing)



Job Information

- Employees (name, job title, phone number, manager, etc.)
- Pictures (badge photos, desk photos, computer photos, etc.)

Physical / Social

Web/Host:

Web / Host



Target Validation

WHOIS, nslookup, dnsrecon



Finding Subdomains

Google Fu, dig, Nmap, Sublist3r, Bluto, crt.sh, etc.



Fingerprinting

Nmap, Wappalyzer, WhatWeb, BuiltWith, Netcat



Data Breaches

HavelBeenPwned, Breach- Parse, WeLeakInfo

Identifying Target

Machine target is located at bugcrowd.com, which is a bug bounty program website. The name of the program is called "Tesla".

Lets make sure to stay within scope of target.

Quiz

1 / 6

Which of the following tools can be used for target validation? (multiple choice)

correct

WHOIS



correct

nslookup



correct

dnsrecon



Continue >

What can the browser add-on Wappalyzer potentially allow us to see? (multiple choice)

correct

CMS used



correct

Underlying OS



correct

Programming languages used



correct

◀ Back

Front-end technologies (such as jQuery)

Continue ▶



What search phrase can be used to find subdomains of a website?

subdomains:tcn-sec.com

sub:tcn-sec.com

correct

site:tcn-sec.com -www



site:tcn-sec.com -sub

◀ Back

Continue ▶

Discovering Email Addresses

Here are some websites, and application that could help gathering information passively.

1 - Hunter.io

This is a website where we can find email addresses for companies.

We can also find some of the technologies that are used by them.



Tesla



665 email addresses

Currently hiring

Description

Tesla is an automotive company that specializes in electric vehicles and clean energy technology.

Details

Industry:

Motor Vehicle Manufacturing

Size: 501-1000 employees

Address: Amsterdam, Netherlands

Website: tesla.com

Email pattern: {f}{last}@tesla.com

Keywords:

electric vehicles, technology,
clean energy, automotive

Social: [in](#) [@](#)

Email addresses

Technologies

Signals

665 results for your search

Filters

Nathan Crandall

ncrandall@tesla.com

99%

7 sources

Save as lead




Add to a campaign



Technologies used on **tesla.com**. Found with [TechLookup](#).


Analytics

 Google Analytics

CDN

 Akamai


CMS

 Drupal


Captchas

 reCAPTCHA

Issue trackers

 Sentry

Maps

 Google Maps

Marketing Automation

 Marketo

Programming Language

Programming Language

 PHP

Tag Managers

 Google Tag Manager

[Tell us what you think](#)



We can also find email addresses for specific departments.

2 - Phonebook.cz / Intelx.io

We can search Domains, URLs, and Email Addresses

3 - www.voilanorbert.com

Also email searching.

4 - Clearbit Connect - Must be used in Chrome (it requires plugin extension).

5 - EmailHippo - <https://tools.verifyemailaddress.io>

We can verify email addresses.

6 - <https://email-checker.net/validate>

Email address verifier.

Gathering Breached Credentials with Breach-Parse

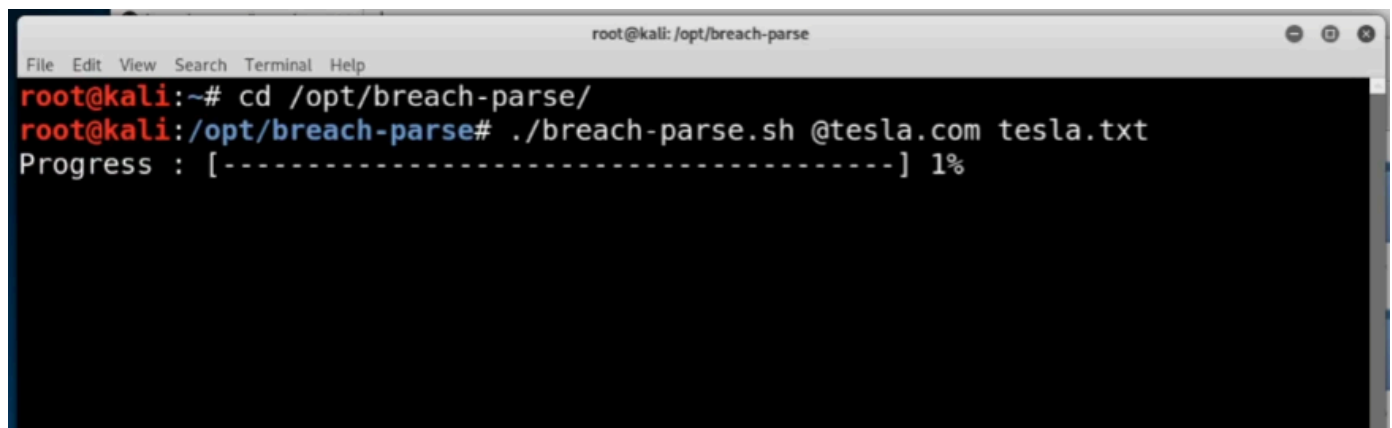
Another good tool is Heath's tool available in GitHub called "breach-parse".

GitHub page is "<https://github.com/hmaverickadams/breach-parse>"

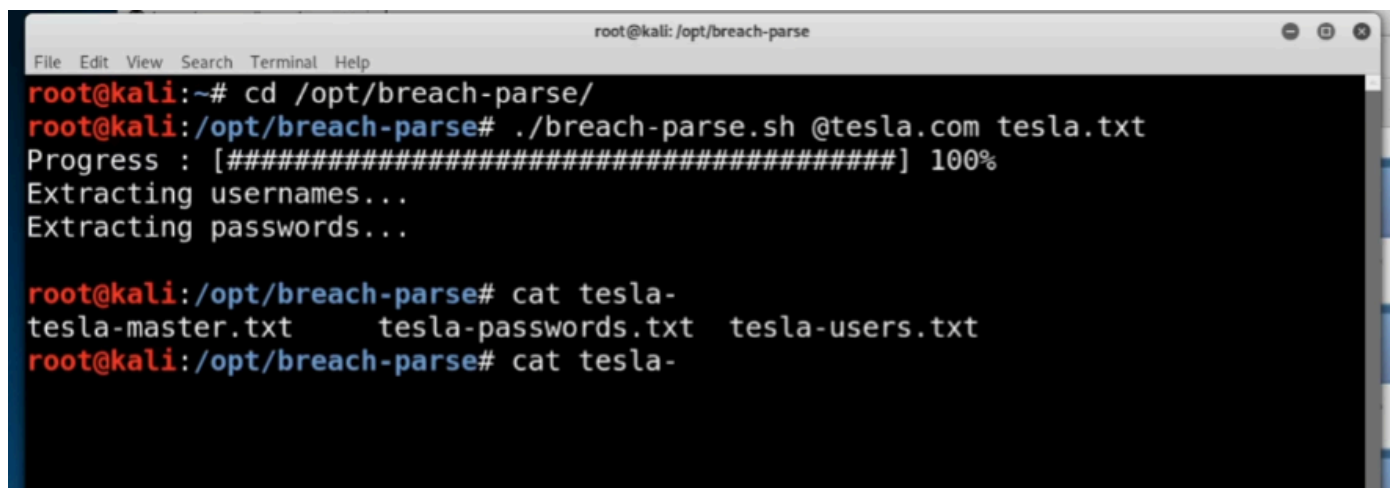
Usage: `#!/breach-parse.sh @gmail.com gmail.txt`

`#!/breach-parse.sh @tesla.com tesla.txt`

Dont forget to save the data at "/opt/breach-parse" otherwise we need to specify the path for the file containing the data. We can actually save everything to the mentioned directory. It is just a matter of preference.

A terminal window titled 'root@kali: /opt/breach-parse' showing the execution of the breach-parse.sh script. The user navigates to the directory and runs the script with '@tesla.com' and 'tesla.txt' as arguments. The progress bar shows 1% completion.

```
root@kali:~# cd /opt/breach-parse/
root@kali:/opt/breach-parse# ./breach-parse.sh @tesla.com tesla.txt
Progress : [-----] 1%
```

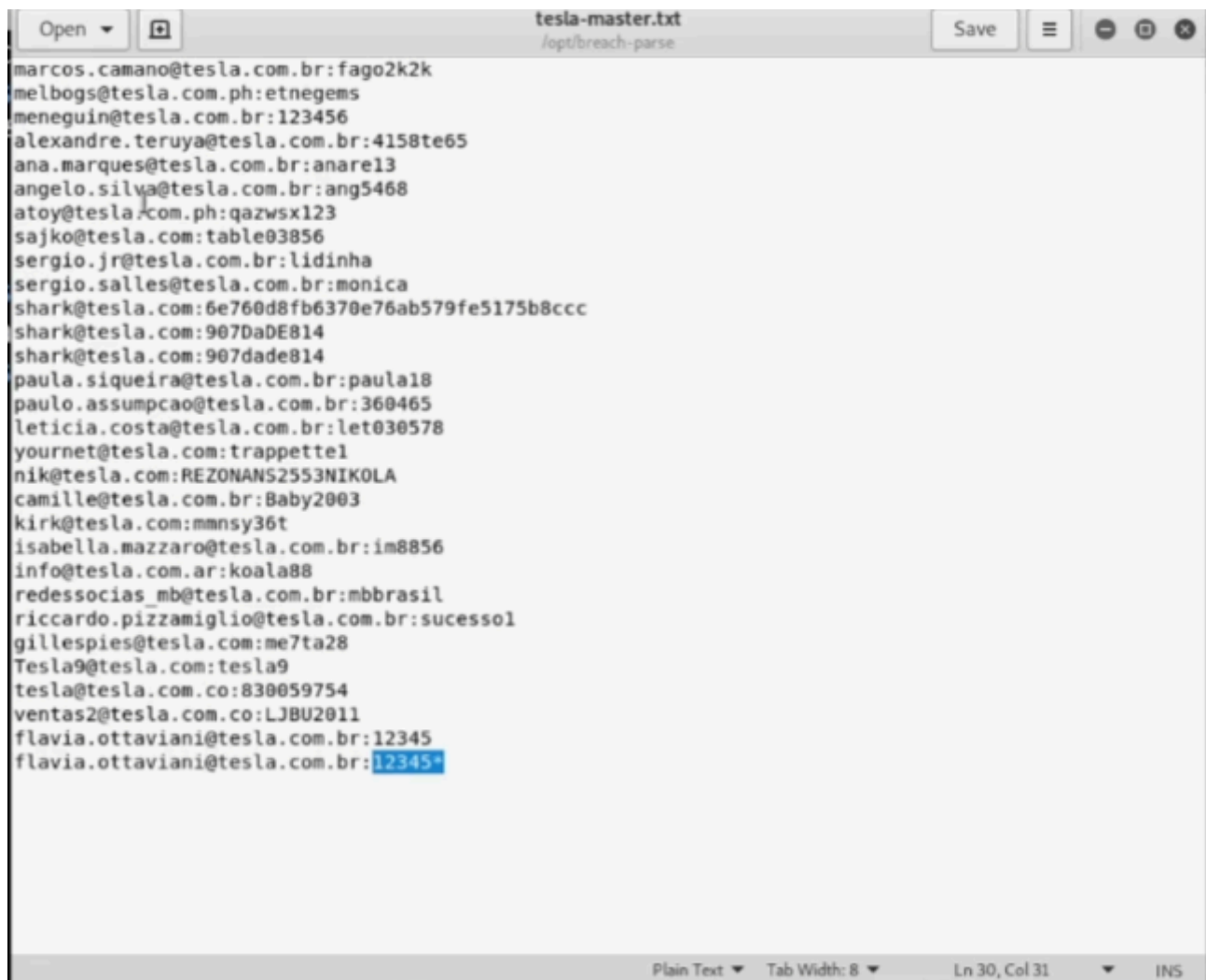
A terminal window titled 'root@kali: /opt/breach-parse' showing the completion of the breach-parse.sh script. The progress bar is at 100%, and the script has finished extracting usernames and passwords. The user then lists the files created in the directory.

```
root@kali:~# cd /opt/breach-parse/
root@kali:/opt/breach-parse# ./breach-parse.sh @tesla.com tesla.txt
Progress : [#####] 100%
Extracting usernames...
Extracting passwords...

root@kali:/opt/breach-parse# cat tesla-
tesla-master.txt      tesla-passwords.txt  tesla-users.txt
root@kali:/opt/breach-parse# cat tesla-
```

Here we are after the passwords. We can enumerate usernames patterns, and password patterns as well for users.

In particular, we are after the repeat offenders. Same username with more than 1 password found. Credentials stuffing is when you already have an idea of what the password could be to see if something sticks.



```
tesla-master.txt
/opt/bleach-parse

marcos.camano@tesla.com.br:fago2k2k
melbogs@tesla.com.ph:etnegems
meneguini@tesla.com.br:123456
alexandre.teruya@tesla.com.br:4158te65
ana.marques@tesla.com.br:anare13
angelo.silva@tesla.com.br:ang5468
atoy@tesla.com.ph:qazwsx123
sajko@tesla.com:table03856
sergio.jr@tesla.com.br:lidinha
sergio.salles@tesla.com.br:monica
shark@tesla.com:6e760d8fb6370e76ab579fe5175b8ccc
shark@tesla.com:907DaDE814
shark@tesla.com:907dade814
paula.siqueira@tesla.com.br:paula18
paulo.assumpcao@tesla.com.br:360465
leticia.costa@tesla.com.br:let030578
yournet@tesla.com:trappettel
nik@tesla.com:REZONANS2553NIKOLA
camille@tesla.com.br:Baby2003
kirk@tesla.com:mmnsy36t
isabella.mazzaro@tesla.com.br:im8856
info@tesla.com.ar:koala88
redessocias_mb@tesla.com.br:mbbrasil
riccardo.pizzamiglio@tesla.com.br:sucessol
gillespies@tesla.com:me7ta28
Tesla9@tesla.com:tesla9
tesla@tesla.com.co:830059754
ventas2@tesla.com.co:LJBU2011
flavia.ottaviani@tesla.com.br:12345
flavia.ottaviani@tesla.com.br:12345*
```

Plain Text ▾ Tab Width: 8 ▾ Ln 30, Col 31 ▾ INS

Hunting Breached Credentials with DeHashed

Hunting Breached Credentials with DeHashed.com

It is a paid one.

We can search by email, username, ip address, name, address, phone number, and VIN.

We can refine our search in multiple ways. If we know that particular person lives in a specific State, we can try searching with the state name to see if anything good comes back.

The idea here is to tie information together. If the same person is cited multiple times in a data breach, we can start tying information for the same person. How usually is their password. What other accounts have been breached, do we have any other account for that particular person, do we have any other password for other accounts.....

<https://hashes.org>, which is a good place where we could crack this hash.

Hunting Subdomains Part 1

Passively gathering information.

The first thing is finding subdomains for websites.

subdomains are ex: "dev.tesla.com", or "test.tesla.com".

Sublist3r is an awesome tool to find the subdomains.

```
#apt install sublist3r
```

Usage: #sublist3r -d tesla.com



```
root@kali:~# sublist3r -d tesla.com -t 100
```

This will passively search the subdomains for the website specified.

Another tool we can use is the " <https://crt.sh> "

Usage: "%tesla.com"

Use the wildcard to identify all subdomains. We are looking for sso, vpn, dev, epi-toolbox, sso-dev.....

Hunting Subdomains Part 2

Owasp amass - is another tool to find subdomains. Seems to be the latest and greatest. We need to install it on our kali linux.

Another one is the : tomnomnom httpprobe which is a probe tool. It is going to probe the different subdomains for a website to assess if they are alive or not.

Identifying Website Technologies

We are going to be investigating what a website is built with. That would be the framework that it is built with.

If we google and search "built with", and go to their website, we can find the type of tech that the particular company is running, as well as the widgets.

Here we are interested in learn the framework the website is written.

1- builtwith.com

2 - Wappalyzer is another good tool. It is between passive, and active bc we do need to go to the website, but we are not doing any type of scan, just as a normal user. It tells us the framework, the language, and version numbers.

3 - whatweb tool in kali linux. Another tool to investigate website framework, and behind the scenes

Information Gathering with Burp Suite

We can get headers, and api's requesting data.

In the "target" tab, we can see all the traffic that has been intercepted after request the website.

We can see a lot of info in the Response for the request.

Google Fu

We can search "Google Search Syntax" to know better what are the different operators, and how to refine searches on google.

Example here: If search for "tesla". We would get all info google has about "tesla". If we are looking for information regarding the website, then we could search for "site:tesla.com". Here we are not specifying the "www" part because that would limit our search to that specific domain. If we want to see less search results for the "www.tesla.com" domain, we could search for "site:tesla.com -www". If we do not want another specific domain to show on our search, we could also "subtract" that specific domain by searching "site:tesla.com -www -ir" if the domain name is called "ir".

We can also search for specific file types by searching "site:tesla.com filetype:docx" if we are looking for "docx" file type, or we could also search for "pdf", or maybe "xlsx", "csv". The idea is to find sensitive information from a specific company.

Utilizing Social Media

We can also look at linkedin, or twiter, or instagram, or facebook.

We can find badge pictures, desk pictures, software pictures posted by employees.

We want email addresses, anything that has been part of a credentials leak.

People are always the weakest link of an organization.