

Information Gathering (Reconnaissance)

Passive Recon

Physical

- Satellite images
- Drone Recon
- Building Layout(badge readers, break areas, security, fencing)

Social

- Employees (Names, Job Title, Phone Number, Manager, etc.)
- Pictures Badge photos, desk photos, computer photos, etc.

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** - HaveIBeenPwned, Breach-Parse, WeLeakInfo

Identifying our target

Finding a client to "attack". Find a client using [BugCrowd](#). He goes over [Tesla](#) in his video, I will be going over [Humble Bundle](#). No matter the target, be sure to stay in scope on any type of engagement.

Discovering Email Addresses

Helpful Tools

I have my on list of various helpful tools on email OSINT [here](#). Most of these tools were already added there.

Finding emails/username is very helpful for later on in the engagement as you can gather them for credential stuffing and other various types of attacks.

- [Hunter.io](#) - Discover email addresses by company name. This requires you to sign up with an email or Gmail)

Domain Search

humblebundle.com 13 results x Filters

Type Department Show only results with

13 results for your search Export Find by name

Michelle Anderson michelle@humblebundle.com 99%	Media Contact	Save as lead	1 source
Libby Kindle kindle@humblebundle.com 99%		Save as lead	15 sources
Kelley Allen kelley@humblebundle.com 98%	+1 908 294 1818 Writing & Communication	Save as lead	6 sources
support@humblebundle.com 97%	Support	Save as lead	3 sources

Company

Humble Bundle
Humble Bundle sells games, books, software, and more. Our mission is to support charity while providing awesome content to cust... more

Email pattern: {first}@humblebundle.com
Accept all: NO
Industry: Game
Country: United States, New York

Technologies

We see various people with their job title and the source of where they found the email address. They have [plugins/add-ons](#).

- [Phonebook.cz](#) - Lists all domains, email addresses, or URLs for the given input domain. This requires you to login with a [intelx.io](#) account.

Phonebook.cz

[Logout](#)

Phonebook lists all domains, email addresses, or URLs for the given input domain. Wildcards such as *.gov.uk are allowed. You are searching 100 billion records.

humblebundle.com Submit

Try: [cia.gov](#), [cnn.com](#), [netflix.com](#), [*.ru](#), [*.gov.uk](#), [solarwinds.com](#)

☐ Domains
☒ Email Addresses
☐ URLs

[lizzie@humblebundle.com](#)
[christofer@humblebundle.com](#)
[angelica.warstler@humblebundle.com](#)
[jeffrey.rosen@humblebundle.com](#)
[terry.mack@humblebundle.com](#)

- [Clearbit.Connect](#) - Email discovery tool, only works in chrome.

Gathering Breached Credentials with Breach-Parse

[Breach-Parse](#) is a tool created by [Heath](#) and is a tool for parsing breached usernames and passwords that have been dumped online. This pulls from a file under `/opt/breach-parse/BreachCompilation/data` under the `symbols` file. He does provide a password list located [here](#). This is a magnet link so you will need some form of torrent software to download this. It is a larger

download(roughly 41GB).

```
breach-parse
Breach-Parse v2: A Breached Domain Parsing Tool by Heath Adams

Usage: ./breach-parse.sh <domain to search> <file to output> [breach data location]
Example: ./breach-parse.sh @gmail.com gmail.txt
Example: ./breach-parse.sh @gmail.com gmail.txt "~/Downloads/BreachCompilation/data"
You only need to specify [breach data location] if its not in the expected location (/opt/breach-parse/BreachCompilation/data)

For multiple domains: ./breach-parse.sh "<domain to search>|<domain to search>" <file to output>
Example: ./breach-parse.sh "@gmail.com|@yahoo.com" multiple.txt

breach-parse @humblebundle.com humblebundle.txt
Progress : [#####-----] 26%_

breach-parse @humblebundle.com humblebundle.txt
Progress : [#####] 100%
Extracting usernames...
Extracting passwords...

_
```

Once done it is broken in 3 files: `*-master.txt`, `*-passwords.txt`, and `*-users.txt`.

```
ls *.txt
humblebundle-master.txt  humblebundle-passwords.txt  humblebundle-users.txt

_
```

The master.txt file will show user:password. Unfortunately this pulled no information for me from HumbleBundle, but this is what teslas looks like from Heaths video:

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

marcos.camano@tesla.com.br:fago2k2k
melbogs@tesla.com.ph:etnegems
meneguín@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:sucesso1
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*
```

This is very helpful for credential stuffing by using the found users and passwords, but also by changing the passwords with upper/lower case strings, and appending numbers or symbols to the end.

We can also do a password spray by talking all the users and trying the same password against all the different users.

Hunting Breached Credentials with DeHashed

[DeHashed](#) "DeHashed is a public data search-engine created for Security Analysts, Journalists, Security Companies, and everyday people to help secure accounts and provide insight on breaches and account leaks. DeHashed can also be used for investigations & fraud prevention." This does cost money to use with an active subscription.

We can search: email, username, IP address, address, name, Phone number, VIN. At the time of these notes, I do not have an active subscription.

Hashes.org

Hunting Subdomains

Sublist3r

[Sublist3r](#) is a tool designed to enumerate subdomains of websites using OSINT.

```
< > /opt/Sublist3r > master python3 sublist3r.py -h
usage: sublist3r.py [-h] -d DOMAIN [-b [BRUTEFORCE]] [-p PORTS] [-v [VERBOSE]] [-t THREADS] [-e ENGINES]
                  [-o OUTPUT] [-n]

OPTIONS:
  -h, --help            show this help message and exit
  -d DOMAIN, --domain DOMAIN
                        Domain name to enumerate it's subdomains
  -b [BRUTEFORCE], --bruteforce [BRUTEFORCE]
                        Enable the subbrute bruteforce module
  -p PORTS, --ports PORTS
                        Scan the found subdomains against specified tcp ports
  -v [VERBOSE], --verbose [VERBOSE]
                        Enable Verbosity and display results in realtime
  -t THREADS, --threads THREADS
                        Number of threads to use for subbrute bruteforce
  -e ENGINES, --engines ENGINES
                        Specify a comma-separated list of search engines
  -o OUTPUT, --output OUTPUT
                        Save the results to text file
  -n, --no-color        Output without color

Example: python sublist3r.py -d google.com
< > /opt/Sublist3r > master _
```

Note: As of lately, it won't return results, [this issue on github](#) show if we edit the sublist3r.py file, go to line 158 and replace the user agent from:

```
'User-Agent': 'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/58.0.3029.110 Safari/537.36'
```

to:

```
'User-Agent': 'Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:105.0) Gecko/20100101
Firefox/105.0'
```

```
> /opt/Sublist3r > master !1 sudo python3 sublist3r.py -d humblebundle.com
```

```

  _ _ _ _ _
 / _ | _ _ | _ | ( ) _ _ | _ | _ _ / _ _
 \ _ \ | | | | ' _ \ | | / _ | _ | | \ _ |
  _ ) | | | | | ) | | \ _ \ | _ ) | |
 | _ _ / \ _ , | _ _ / | | | _ \ | _ _ / | |

```

```
# Coded By Ahmed Aboul-Ela - @aboul3la
```

```
[!] Enumerating subdomains now for humblebundle.com
[-] Searching now in Baidu..
[-] Searching now in Yahoo..
[-] Searching now in Google..
[-] Searching now in Bing..
[-] Searching now in Ask..
[-] Searching now in Netcraft..
[-] Searching now in DNSdumpster..
[-] Searching now in Virustotal..
[-] Searching now in ThreatCrowd..
[-] Searching now in SSL Certificates..
[-] Searching now in PassiveDNS..
[!] Error: Virustotal probably now is blocking our requests
[-] Total Unique Subdomains Found: 26
www.humblebundle.com
andytest.humblebundle.com
blog.humblebundle.com
www.blog.humblebundle.com
btg.humblebundle.com
checkout.humblebundle.com
cn.humblebundle.com
de.humblebundle.com
developer.humblebundle.com
dsar.humblebundle.com
es.humblebundle.com
fr.humblebundle.com
it.humblebundle.com
jenkins.humblebundle.com
www.jobs.humblebundle.com
mailer.humblebundle.com
d.mailer.humblebundle.com
e.mailer.humblebundle.com
i.mailer.humblebundle.com
t.mailer.humblebundle.com
reviews.humblebundle.com
ru.humblebundle.com
support.humblebundle.com
try.humblebundle.com
www.try.humblebundle.com
zh.humblebundle.com
```

```
> /opt/Sublist3r > master !1 _
```

Crt.sh is a tool to pull certificate fingerprints.

CriteriaType: IdentityMatch: ILIKESearch: 'humblebundle.com'									
Certificates	cert.sh ID	Logged At	Not Before	Not After	Common Name	Matching Identities		Issuer Name	
	10225574950	2023-08-20	2023-08-20	2023-11-18	www.try.humblebundle.com	www.try.humblebundle.com	C=US, O=Let's Encrypt, CN=R3		
	10082620370	2023-08-07	2023-08-07	2023-11-05	www.jobs.humblebundle.com	www.jobs.humblebundle.com	C=US, O=Let's Encrypt, CN=R3		
	10115450783	2023-08-07	2023-08-07	2023-11-05	www.jobs.humblebundle.com	www.jobs.humblebundle.com	C=US, O=Let's Encrypt, CN=R3		
	10000368272	2023-07-27	2023-07-27	2024-07-26	www.ziffdavis.com	btg.humblebundle.com	C=GB, ST=Greater Manchester, L=Salford, O=COMODO CA Limited, CN=COMODO RSA Organization Validation Secure Server CA		
	10000368295	2023-07-27	2023-07-27	2024-07-26	www.ziffdavis.com	btg.humblebundle.com	C=GB, ST=Greater Manchester, L=Salford, O=COMODO CA Limited, CN=COMODO RSA Organization Validation Secure Server CA		
	10000342564	2023-07-27	2023-07-27	2024-07-26	www.ziffdavis.com	btg.humblebundle.com	C=GB, ST=Greater Manchester, L=Salford, O=COMODO CA Limited, CN=COMODO RSA Organization Validation Secure Server CA		
	10000342608	2023-07-27	2023-07-27	2024-07-26	www.ziffdavis.com	btg.humblebundle.com	C=GB, ST=Greater Manchester, L=Salford, O=COMODO CA Limited, CN=COMODO RSA Organization Validation Secure Server CA		

OWASP Amass

OWASP Amass "The OWASP Amass Project performs network mapping of attack surfaces and external asset discovery using open source information gathering and active reconnaissance

techniques."

```
> /opt/amass ./amass

      .+++:.
      +W@@@@@8
      &@#+ .o@##.
      +@&      &@&
      8@        @@
      WW        &@o
      #@        :@W
      o@+       @&&
      WW        +@W@8.
      :@W:      o@# +Wo
      :W@W@W@W@8
      +o&&&&+.

      :
      &+W@#
      .@@@@o@W.o@@@@
      #@8 +@W@&8@+
      8@o 8@8 WW
      &@: o@+ o@+
      &@+ &@+ @8
      &@+ &@+ #@ &@.
      :& o@+ #@
      &@+ :W: +@W&o++o@W.
      +      :W@&@@@&
      &W .o#@@W&.
      :W@W@W@&
      +oooo.

      .+++
      +W@@@@@&#.
      .@#: .:oW+
      +@:
      .@W.
      +W@#+.
      oW@W+
      .+#@&
      &@:
      8@#o+&@W.
      .o#@@W&.
      :W@W@W@&
      +oooo.

      oW@@@@W#+
      .@#+++&#&
      .@8
      o@#:
      +W@8:
      oW@8
      o@W.
      :@o
      #@: o@+
      :W@W@W@&
      +oooo.

v4.1.0
OWASP Amass Project - @owaspamass
In-depth Attack Surface Mapping and Asset Discovery

Usage: amass intel|enum|db [options]

-h      Show the program usage message
-help
      Show the program usage message
-version
      Print the version number of this Amass binary

Subcommands:

      amass intel - Discover targets for enumerations
      amass enum  - Perform enumerations and network mapping
      amass db    - Manipulate the Amass graph database

The user's guide can be found here:
https://github.com/owasp-amass/amass/blob/master/doc/user_guide.md

An example configuration file can be found here:
https://github.com/owasp-amass/amass/blob/master/examples/config.yaml

The Amass tutorial can be found here:
https://github.com/owasp-amass/amass/blob/master/doc/tutorial.md

> /opt/amass _
```

A basic way to use amass is `amass enum -d example.com`, but as I am testing against HumbleBundle


```
🔍 > /opt/amass ➤ ./amass enum -d humblebundle.com
humblebundle.com (FQDN) --> ns_record --> mary.ns.cloudflare.com (FQDN)
humblebundle.com (FQDN) --> ns_record --> todd.ns.cloudflare.com (FQDN)
humblebundle.com (FQDN) --> mx_record --> alt4.aspmx.l.google.com (FQDN)
humblebundle.com (FQDN) --> mx_record --> alt2.aspmx.l.google.com (FQDN)
humblebundle.com (FQDN) --> mx_record --> alt1.aspmx.l.google.com (FQDN)
humblebundle.com (FQDN) --> mx_record --> aspmx.l.google.com (FQDN)
humblebundle.com (FQDN) --> mx_record --> alt3.aspmx.l.google.com (FQDN)
```

Identifying Website Technologies

Tools we can use:

[Builtwith](#)

We can see various technologies the site is running, any widgets, languages, etc. What frameworks it's running. CDN(Content Delivery Network), etc.

[Wappalyzer](#) - Ad-on/Extention for Chrome/Firefox and again see what technologies the website is using.

[WhatWeb](#) - Simply analyzes websites.

Information Gathering with Burp Suite

I have my own notes and documentation on Burpsuite [here](#).

[Burpsuite](#)

Google Fu

Simple enough, use [Google](#). Using and understanding [Google Search Syntax](#) will be helpful. understanding how to use `site:`, remove results using subtract (`-www`), `filetype:` to find documents or PDFs,

Utilizing Social Media

Use Social Media like [LinkedIn](#), [Facebook](#), [Twitter](#), etc.

Images hold a lot of information that have valuable OPSEC. Badge pictures, desks, etc. See the people and their email addresses and roles. Use their names for potential email addresses like

`first.last@domain.com`.