Capt. Meelo

An infosec guy who's constantly seeking for knowledge.

Asset Enumeration: Expanding a Target's Attack Surface

02 Sep 2019 » bugbounty

Introduction

Whenever I'm doing bug hunting sessions with a wide range of scope (e.g. CIDRs, subdomains, all assets belonging to a company, etc.), I'm always overwhelmed with the amount of information that I have to gather to expand my attack surface. It's true that `a wider scope = a larger attack surface = more chances of pwning`. However, increasing the attack surface is always a challenge for me.

In this post, I'll describe the methodology that I'm using to expand the attack surface of my target domain or company.

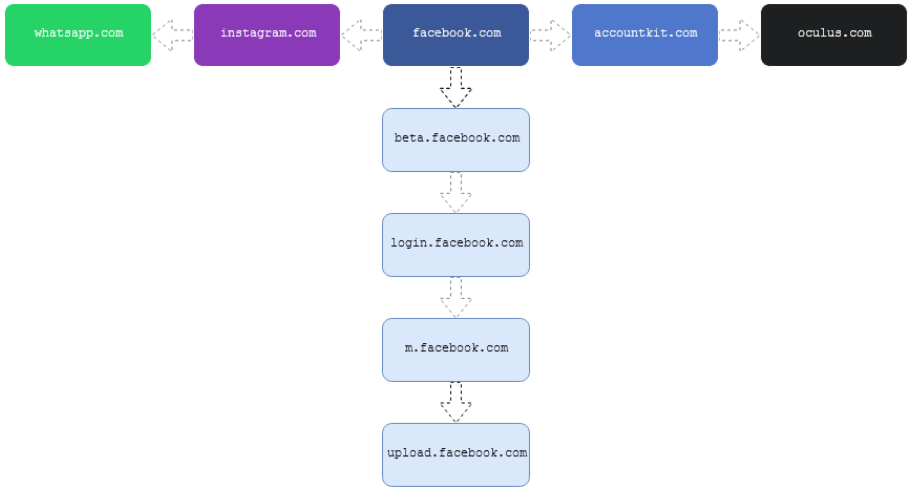
> *Please note that this methodology is limited only to enumerating subdomains and finding the domains associated with the target domain.*

Asset Discovery

Given a target domain, there are two ways to expand its attack surface:

1. Find as many domain names as possible that share the same base domain with the target. This is commonly called as **subdomain enumeration**.
2. Identify all domain names "associated" with the target domain.

The illustration below helps differentiate what the two methods do.



Now let's take a look at the methodology that I'm using to enumerate both the subdomains and the associated domains.

Subdomain Enumeration

The focus of this step is to identify as much subdomains as possible that are tied to the target domain. Ideally, we want to go as deep as we can. What do I mean by that is we don't want to limit our enumeration in finding only the subdomains of the target domain. We also want to identify the subdomains of the subdomains of the target domain. For example, we want to also identify `another.subdomain.domain.com`, and not just `subdomain.domain.com`.

There are many tools that will do subdomain enumeration but not all of them provide good results. Personally, I prefer to use a tool that combines results from various enumeration services or sources of

inputs, and has all the options and functionalities that I need such as recursive brute force and alteration of words.

For subdomain enumeration, I always start with **Amass** using its `-passive` option.

```
amass enum -passive -d <DOMAIN> -o <OUT_FILE>
```

Followed by a brute force attack on the target domain using either all.txt or commonspeak2.

```
amass enum -brute -w <WORDLIST> -d <DOMAIN> -o <OUT_FILE>
```

> *The success and efficiency of your brute force attack relies mostly on your wordlist; so better use a highly-reputed one.*

If you want to speed up the performance of **Amass**, you can use the options `-noalts`, `-norecursive`, and `-max-dns-queries`. Just don't be surprised if you got fewer results.

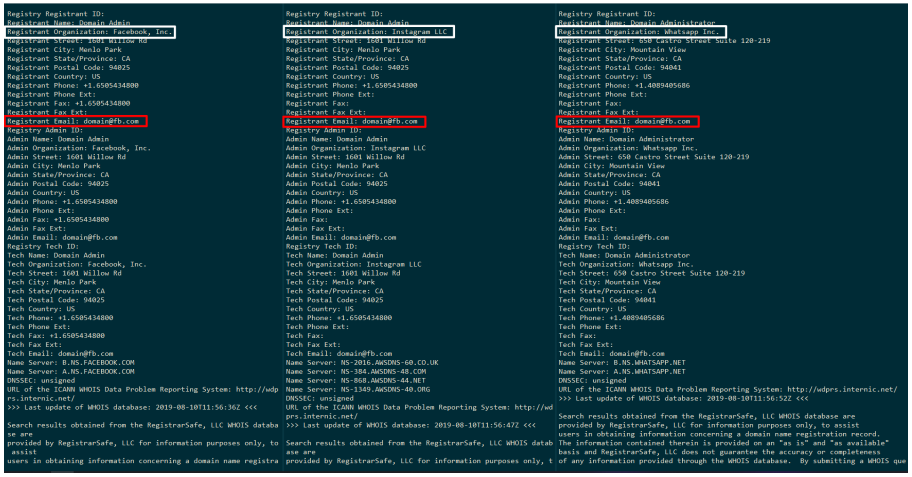
Not all subdomains gathered from the above commands will resolve to its corresponding IP addresses. To filter out only those that will be resolved, I prefer to use **Massdns**.

```
./bin/massdns -r lists/resolvers.txt -o S <LIST_OF_SUBDOMAINS> | grep -e ' A ' |  cut -d 'A' -f 1 | rev | cut -d "." -f1 --complement | rev | sort | uniq > <OUT_FILE>
```

Associated Domains Enumeration

Through acquisitions and merges, it is not only a company's business that grows but also their domains and associated domains. For example, when "Facebook, Inc." acquired Instragram and Whatsapp, the domains `instragram.com` and `whatsapp.com` became associated with `facebook.com`.

As we can see from the following `whois` queries below, the domains `facebook.com`, `instagram.com`, and `whatsapp.com` were all registered by the email address `domain@fb.com`.



To enumerate domains that are associated with the target domain, we could use the `Registrant Email` record taken from a WHOIS search result and perform a **Reverse WHOIS Lookup**. This can be done using

sites such as viewdns.info or whoisxmlapi.com.

ViewDNS.info

Tools | API | Research | Data

ViewDNS.info > Tools > **Reverse Whois Lookup**

This free tool will allow you to find domain names owned by an individual person or company. Simply enter the email address or name of the person or company to find other domains registered using those same details. FAQ.

Registrant Name or Email Address:

GO

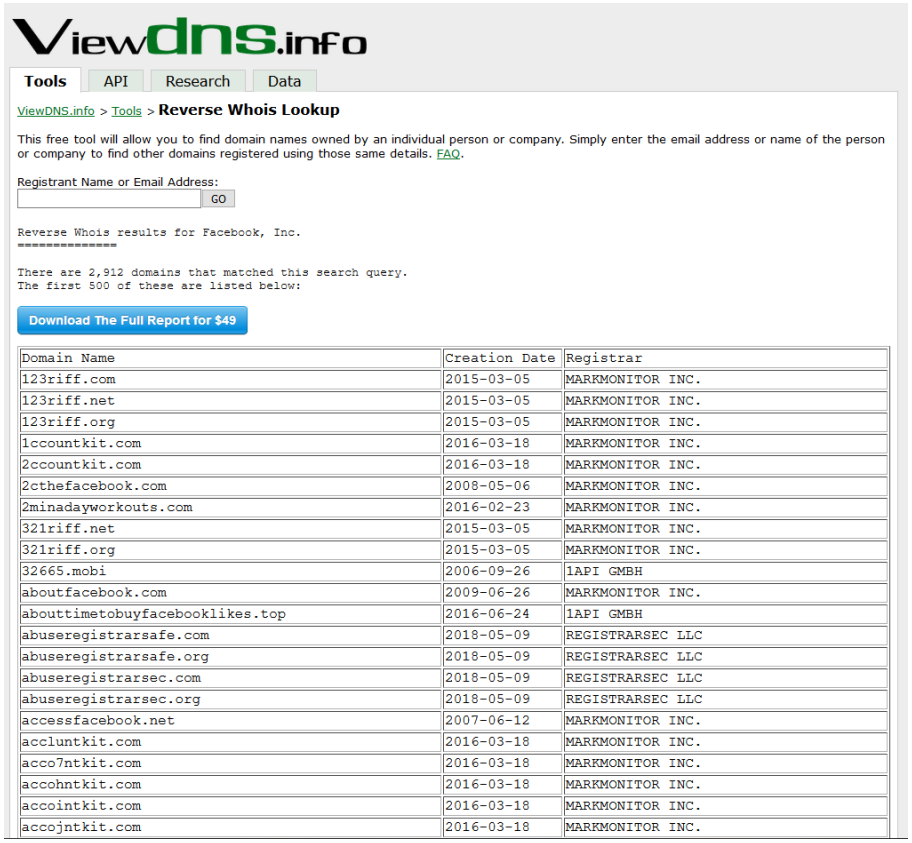
Reverse Whois results for domain@fb.com

There are 3,739 domains that matched this search query.
The first 500 of these are listed below:

Download The Full Report for $49

Domain Name	Creation Date	Registrar
0facebook.me	2010-04-09	REGISTRARSAFE, LLC
0fb.me	2010-04-09	REGISTRARSAFE, LLC
123riff.com	2015-03-05	MARKMONITOR INC.
123riff.net	2015-03-05	MARKMONITOR INC.
123riff.org	2015-03-05	MARKMONITOR INC.
1ccountkit.com	2016-03-18	MARKMONITOR INC.
2ccountkit.com	2016-03-18	MARKMONITOR INC.
2cthefacebook.com	2008-05-06	MARKMONITOR INC.
2minadayworkouts.com	2016-02-23	MARKMONITOR INC.
321riff.com	2015-03-05	REGISTRARSEC LLC
321riff.net	2015-03-05	MARKMONITOR INC.
321riff.org	2015-03-05	MARKMONITOR INC.
32665.mobi	2006-09-26	1API GMBH
360videofb.com	2016-02-29	REGISTRARSAFE, LLC
360videofb.net	2016-02-29	REGISTRARSAFE, LLC
360videofb.org	2016-02-29	REGISTRARSAFE, LLC
aboutfacebook.com	2009-06-26	MARKMONITOR INC.

You can also run a Reverse WHOIS Lookup based on the `Registrant Organization` record to get different results.



ViewDNS.info

Tools | API | Research | Data

ViewDNS.info > Tools > **Reverse Whois Lookup**

This free tool will allow you to find domain names owned by an individual person or company. Simply enter the email address or name of the person or company to find other domains registered using those same details. FAQ.

Registrant Name or Email Address:

GO

Reverse Whois results for Facebook, Inc.

There are 2,912 domains that matched this search query.
The first 500 of these are listed below:

Download The Full Report for $49

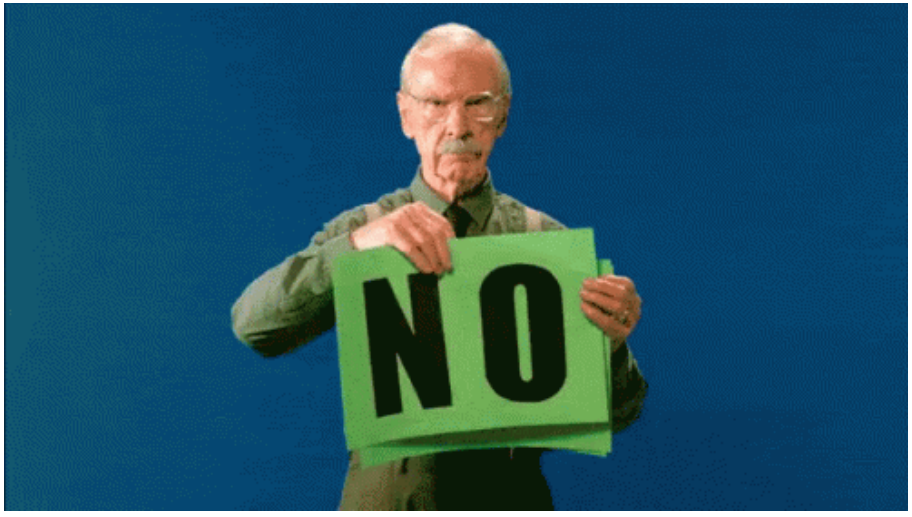
Domain Name	Creation Date	Registrar
123riff.com	2015-03-05	MARKMONITOR INC.
123riff.net	2015-03-05	MARKMONITOR INC.
123riff.org	2015-03-05	MARKMONITOR INC.
1ccountkit.com	2016-03-18	MARKMONITOR INC.
2ccountkit.com	2016-03-18	MARKMONITOR INC.
2cthefacebook.com	2008-05-06	MARKMONITOR INC.
2minadayworkouts.com	2016-02-23	MARKMONITOR INC.
321riff.net	2015-03-05	MARKMONITOR INC.
321riff.org	2015-03-05	MARKMONITOR INC.
32665.mobi	2006-09-26	1API GMBH
aboutfacebook.com	2009-06-26	MARKMONITOR INC.
abouttimetobuyfacebooklikes.top	2016-06-24	1API GMBH
abuseregistrarsafe.com	2018-05-09	REGISTRARSEC LLC
abuseregistrarsafe.org	2018-05-09	REGISTRARSEC LLC
abuseregistrarsec.com	2018-05-09	REGISTRARSEC LLC
abuseregistrarsec.org	2018-05-09	REGISTRARSEC LLC
accessfacebook.net	2007-06-12	MARKMONITOR INC.
accluntkit.com	2016-03-18	MARKMONITOR INC.
acco7ntkit.com	2016-03-18	MARKMONITOR INC.
accohntkit.com	2016-03-18	MARKMONITOR INC.
accointkit.com	2016-03-18	MARKMONITOR INC.
accojntkit.com	2016-03-18	MARKMONITOR INC.

It is normal to get overlapping results when performing Reverse WHOIS Lookup based on `Registrant Email` and `Registrant Organization` records, so make some post-processing and remove the duplicates.

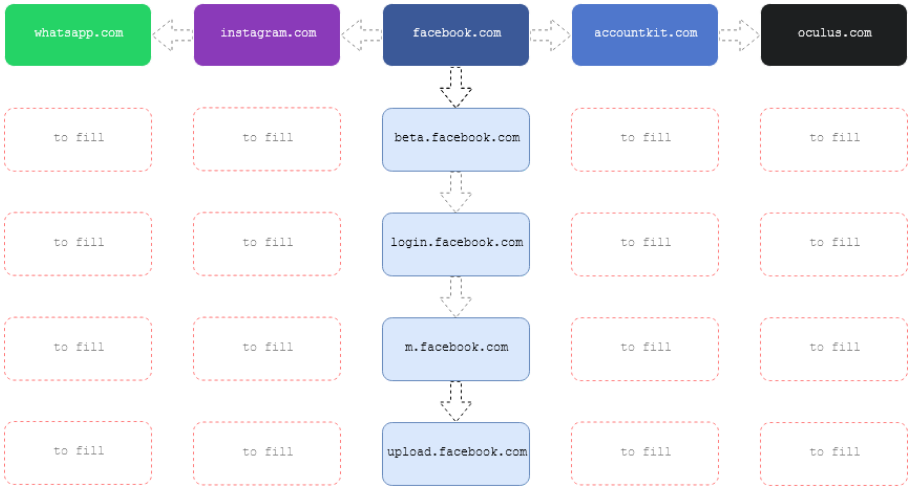
> *Be wary that most "Reverse WHOIS" services are freemium. If you want to get more results (or have the full report), they require you to spend some $$$.*

Filling Up the Empty Areas

Now that we've already enumerated the target domain "vertically" and "horizontally", should we stop and proceed with attacking these subdomains and associated domains? I'd say **NO!**.



Wouldn't it be nice if we expand more our target's attack surface by filling up the empty areas (*shown in red boxes*) below?



We can do this by doing subdomain enumeration on every associated domains that we've discovered from the previous step. Obviously, this will take way more time than doing a subdomain enumeration on the target domain alone. But who cares? Right? Remember, `a larger attack surface = more chances of pwning`.

But before that, filter first only those associated domains that will be resolved:

```
./bin/massdns -r lists/resolvers.txt -o S <LIST_OF_ASSOCIATED_DOMAINS> | grep -e ' A ' |  cut -d 'A' -f 1 | rev | cut -d "." -f1 --complement | rev | sort | uniq > <OUT_FILE>
```

Then run another subdomain enumeration via:

```
amass enum -passive -df <LIST_OF_RESOLVED_ASSOCIATED_DOMAINS> -o <OUT_FILE>
```

or through brute force attack:

```
amass enum -brute -w <WORDLIST> -df <LIST_OF_RESOLVED_ASSOC
IATED_DOMAINS> -o <OUT_FILE>
```

If you have a powerful machine, you can speed up this process by running multiple concurrent jobs using GNU Parallel or Xargs. For example:

```
cat <LIST_OF_RESOLVED_ASSOCIATED_DOMAINS> | parallel -j <NO
_OF_CONCURRENT_JOBS> "amass enum -passive -d {} -o {}.out"
```

> I prefer to use **GNU Parallel** so you won't see commands related to **xargs** here.

One you've enumerated the subdomains of all identified associated domains, filter them out again by running a DNS resolution using **Massdns**.

What's Next?

After doing the above steps, the last thing to do is to combine them and remove any duplicates.

Using all these enumerated data/hosts, you can do the following:

- Check for subdomain takeover
- Run a port scan and identify any running services
- Take screenshots for host/s that have web service/s running
- Run a directory brute force attack
- etc.

Conclusion

This methodology is what I'm doing when I'm trying to do bug bounties. I cannot guarantee that this will work for you as well, and there's no assurance that you will find any bug once you follow this methodology. This post was written to share my knowledge and to help bug hunters like me who are struggling with expanding a target's attack surface.

That's it for this post. I hope you will find this useful.

« Finding the Balance Between Speed & Accuracy During an Internet-wide Port Scanning