Recon For Red Teamers And Bug Hunters

Outline-ish

Setup

1. Questions
2. Note-taking

Wide recon

1. (P) IP enumeration (ASNs and Registrars)
2. (P) SHODAN
3. (P) Brand Enumeration (Acquisitions, RevWHOIS, Reverse tracker Analysis)
4. (A) Linked discovery
5. (P) Subdomain Enumeration (Scraping)
    a. (A) Bruteforcing,
    b. (A) permutation scanning
6. (P) Github
7. (P) Cloud Recon
8. (A) Screenshotting

Narrow recon

9. (A) Effective Port Scanning
10. (A) Version based vulnerability analysis
11. (A) Javascript analysis
12. (A++) Directory Bruteforcing / Content Discovery best practices
13. (A) Prioritizing target testing areas by technology and features

Social Recon

14. (P) Email enumeration
15. (P) Breach TI
16. (A-) Doc Enum

Questions

Welcome.

At any time during the workshop, if you don’t necessarily feel like asking in front of the class you may submit a question to me here:

jhaddix.twitter@gmail.com

Tools

Core:

https://portswigger.net/burp
https://github.com/OWASP/Amass
https://github.com/subfinder/subfinder
https://github.com/robertdavidgraham/masscan

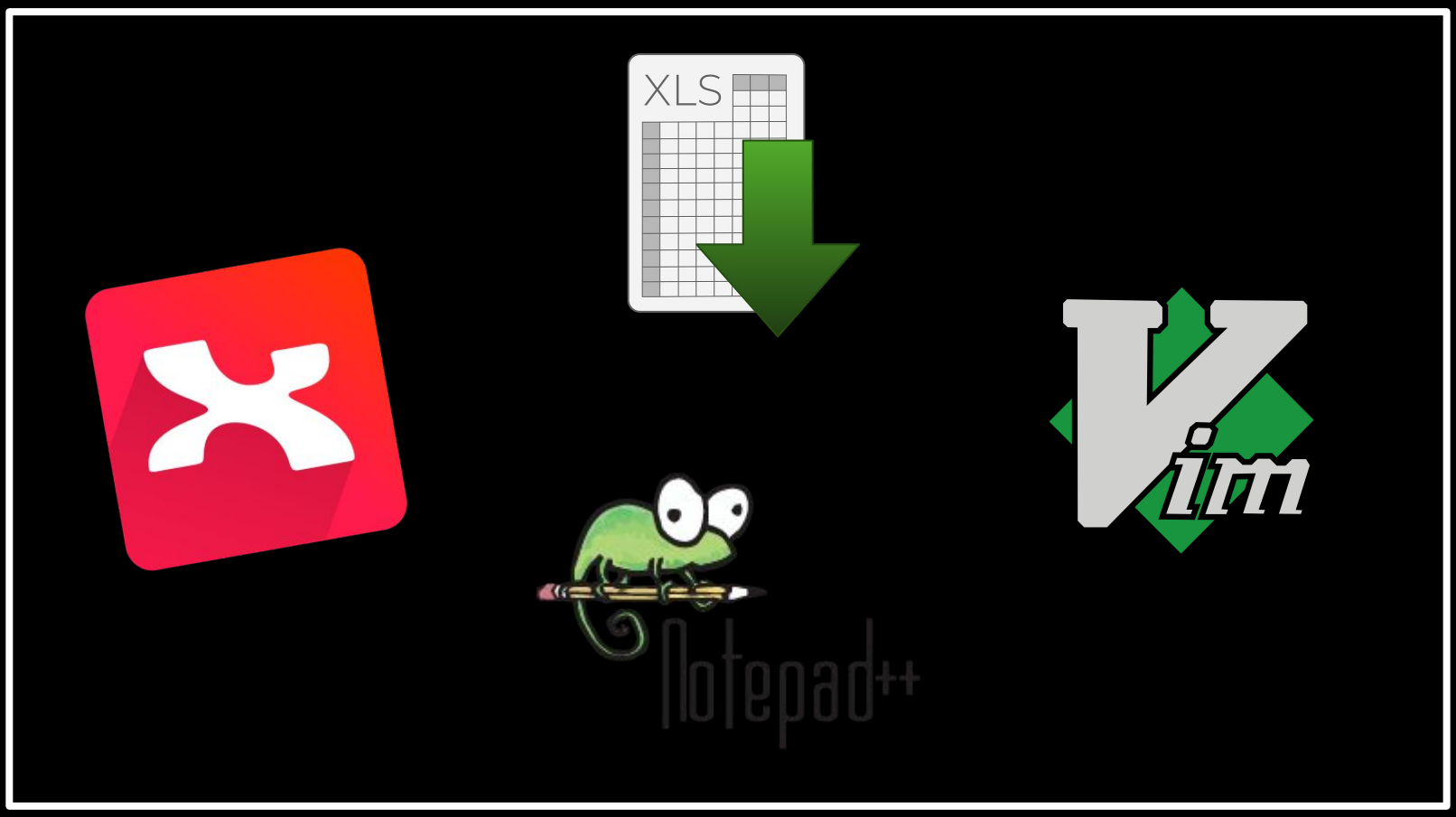
Optional:

https://www.xmind.net/

We will be working with many tools and discussing their merits in several sections of the training. Here are some I recommend downloading and installing now. Some of them will be available on Kali if you chose to bring that.

Please have docker and GO installed asap.

Note Taking

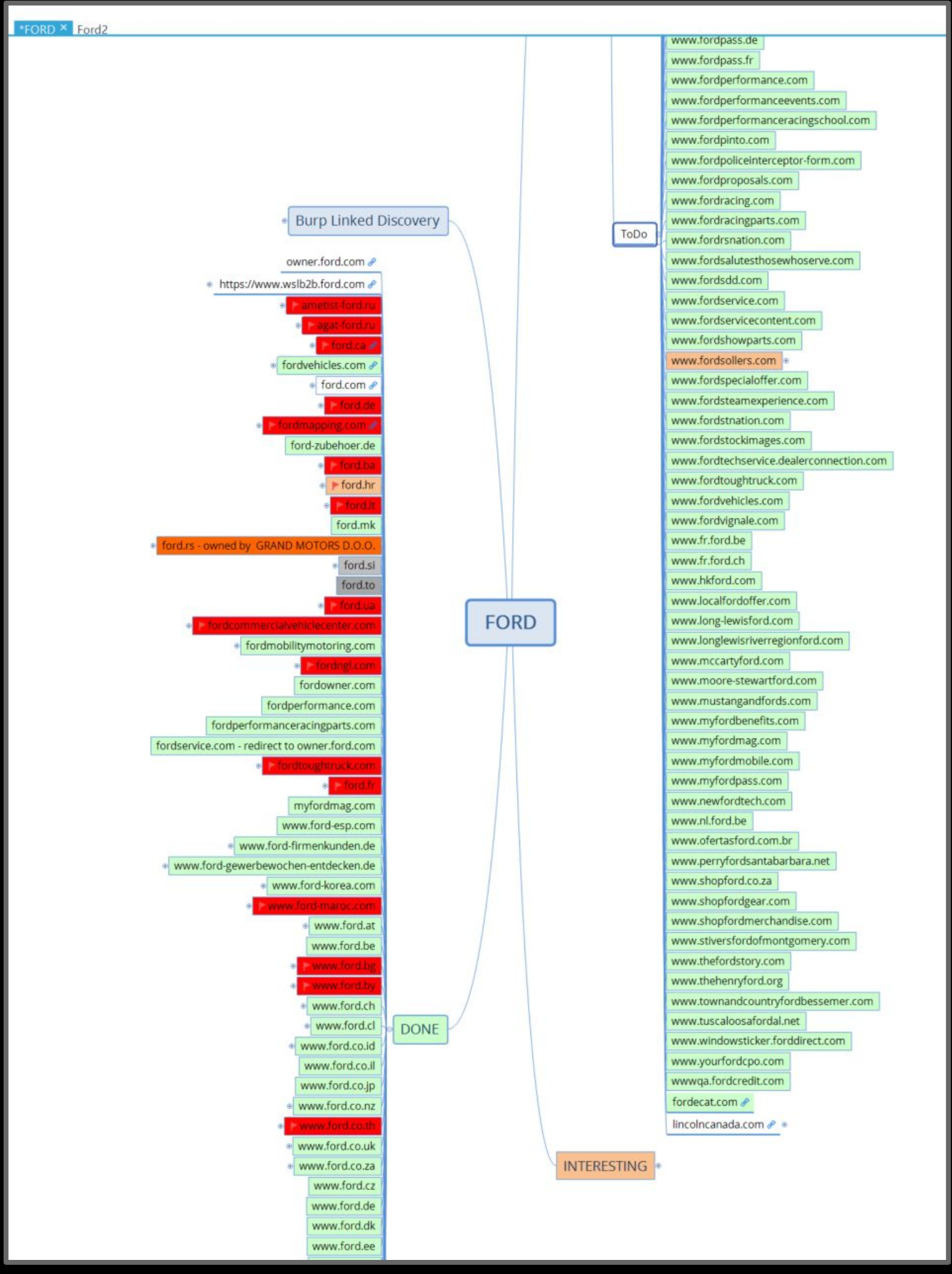


In many parts of the course we will need to keep track of site-hierarchy, tools output, interesting notes, etc.

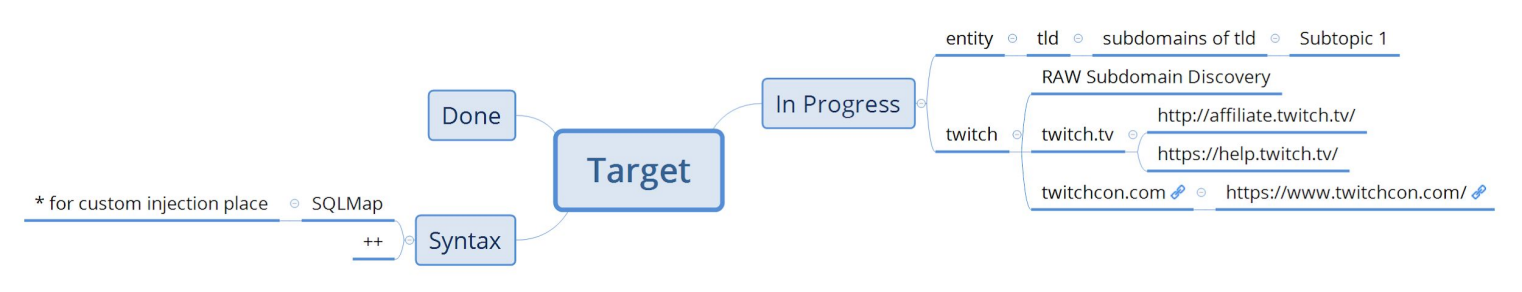
I use mindmaps with Xmind. I will use it throughout the course but the same effect can be achieved through a lot of different programs.

Mindmaps allow me to visualize large scope bug hunting targets and also allow me to break up methodology for in-depth bug hunting as well.

Here's and example:


*FORD
Ford2
FORD
Burp Linked Discovery
owner.ford.com
https://www.wslb2b.ford.com
ametist-ford.ru
agat-ford.ru
ford.ca
fordvehicles.com
ford.com
ford.de
fordmapping.com
ford-zubehoer.de
ford.ba
ford.hr
ford.lt
ford.mk
ford.rs - owned by GRAND MOTORS D.O.O.
ford.si
ford.to
ford.ua
fordcommercialvehiclecenter.com
fordmobilitymotoring.com
fordngl.com
fordowner.com
fordperformance.com
fordperformanceracingparts.com
fordservice.com - redirect to owner.ford.com
fordtoughtruck.com
ford.fr
myfordmag.com
www.ford-esp.com
www.ford-firmenkunden.de
www.ford-gewerbewochen-entdecken.de
www.ford-korea.com
www.ford-maroc.com
www.ford.at
www.ford.be
www.ford.bg
www.ford.by
www.ford.ch
www.ford.cl
www.ford.co.id
www.ford.co.il
www.ford.co.jp
www.ford.co.nz
www.ford.co.th
www.ford.co.uk
www.ford.co.za
www.ford.cz
www.ford.de
www.ford.dk
www.ford.ee
DONE
ToDo
www.fordpass.de
www.fordpass.fr
www.fordperformance.com
www.fordperformanceevents.com
www.fordperformanceracingschool.com
www.fordpinto.com
www.fordpoliceinterceptor-form.com
www.fordproposals.com
www.fordracing.com
www.fordracingparts.com
www.fordrsnation.com
www.fordsalutesthosewhoserve.com
www.fordsdd.com
www.fordservice.com
www.fordservicecontent.com
www.fordshowparts.com
www.fordsollers.com
www.fordspecialoffer.com
www.fordsteamexperience.com
www.fordstnation.com
www.fordstockimages.com
www.fordtechservice.dealerconnection.com
www.fordtoughtruck.com
www.fordvehicles.com
www.fordvignale.com
www.fr.ford.be
www.fr.ford.ch
www.hkford.com
www.localfordoffer.com
www.long-lewisford.com
www.longlewisriverregionford.com
www.mccartyford.com
www.moore-stewartford.com
www.mustangandfords.com
www.myfordbenefits.com
www.myfordmag.com
www.myfordmobile.com
www.myfordpass.com
www.newfordtech.com
www.nl.ford.be
www.ofertasford.com.br
www.perryfordsantabarbara.net
www.shopford.co.za
www.shopfordgear.com
www.shopfordmerchandise.com
www.stiversfordofmontgomery.com
www.thefordstory.com
www.thehenryford.org
www.townandcountryfordbessemer.com
www.tuscaloosafordal.net
www.windowsticker.forddirect.com
www.yourfordcpo.com
wwwqa.fordcredit.com
fordecat.com
lincolncanada.com
INTERESTING

Note Taking



That mindmap grew from a simple template like the one above.

Green means site / entity had no vulns

Orange means currently in progress of testing

Red means vulns found

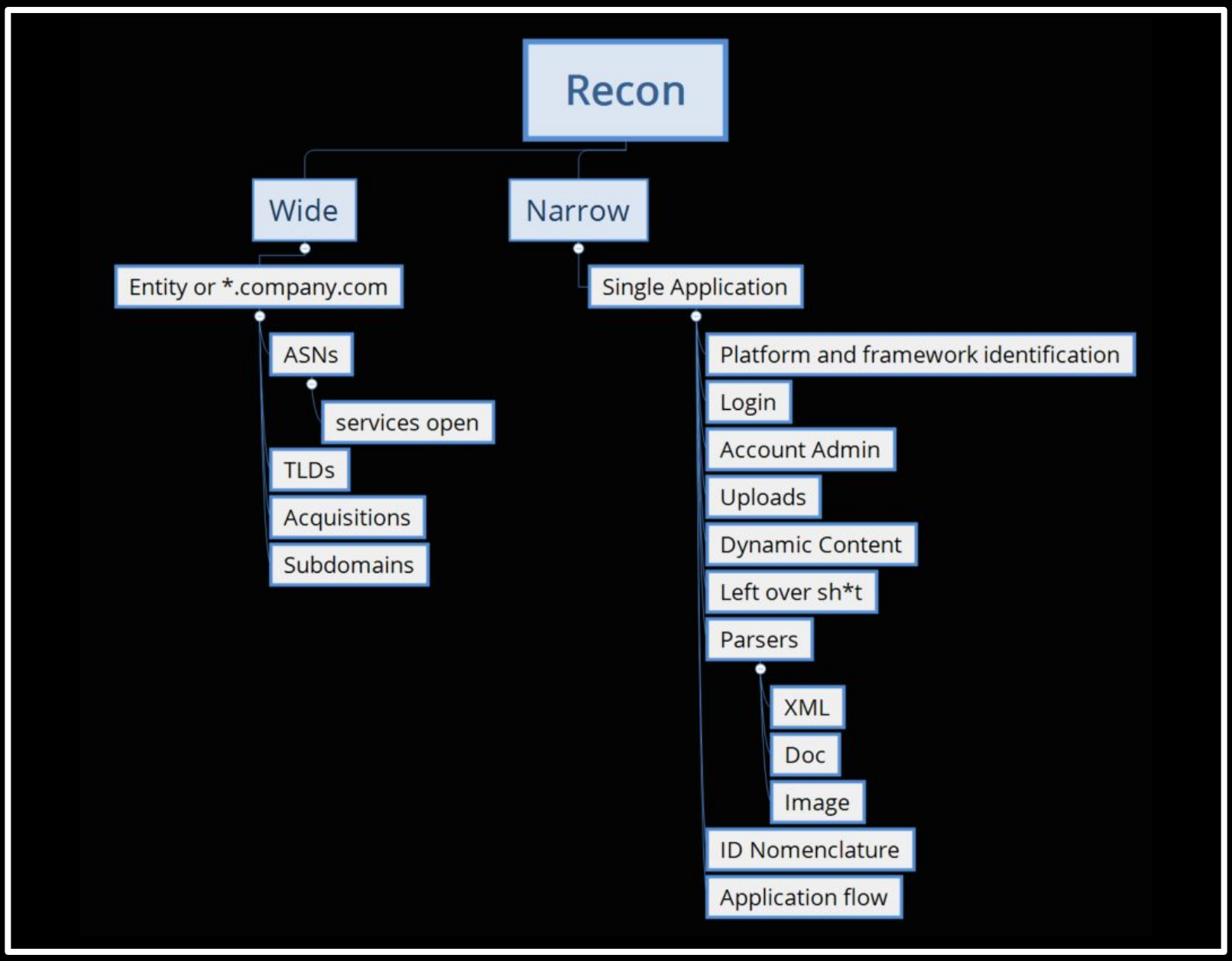
Checkmark icon means testing completed

You can keep track of targets, status/workflow, common tool syntax, and common reporting templates:

Recon

```
[05:06:24] [INFO] loading tamper script 'versionedmorekeywords'
[05:06:24] [WARNING] tamper script 'versionedmorekeywords' is only meant to be run against MySQL >= 5.1.13
[05:06:24] [INFO] loading tamper script 'xforwardedfor'
[05:06:24] [WARNING] using too many tamper scripts is usually not a good idea
custom injection marking character ('*') found in option '--headers/--user-agent/--referer/--cookie'. Do you want to process it? [Y/n/q] Y
[05:06:28] [INFO] testing connection to the target URL
[05:06:29] [WARNING] the web server responded with an HTTP error code (406) which could interfere with the results of the tests
[05:06:29] [INFO] testing if the target URL is stable
[05:06:29] [INFO] target URL is stable
[05:06:29] [INFO] testing if (custom) HEADER parameter 'Cookie #1*' is dynamic
[05:06:29] [WARNING] currently only couple of keywords are being processed ('UNION', 'SELECT', 'INSERT', 'UPDATE', 'FROM', 'WHERE'). You can set it manually according to your needs
[05:06:30] [WARNING] reflective value(s) found and filtering out
[05:06:30] [INFO] confirming that (custom) HEADER parameter 'Cookie #1*' is dynamic
[05:06:31] [INFO] (custom) HEADER parameter 'Cookie #1*' is dynamic
[05:06:31] [WARNING] heuristic (basic) test shows that (custom) HEADER parameter 'Cookie #1*' might not be injectable
[05:06:34] [INFO] testing for SQL injection on (custom) HEADER parameter 'Cookie #1*'
[05:06:34] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[05:06:39] [INFO] (custom) HEADER parameter 'Cookie #1*' seems to be 'AND boolean-based blind - WHERE or HAVING clause' injectable (with --string="\\xa0\\xa0\\xa0\\xa0\n\tat com.ibm.ws.http.channel.inbound.impl.HttpInboundLink.ready(HttpInboundLink.java:287)")
[05:06:48] [INFO] testing 'MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause'
[05:06:48] [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'
[05:06:49] [INFO] testing 'Microsoft SQL Server/Sybase AND error-based - WHERE or HAVING clause'
[05:06:49] [INFO] testing 'Oracle AND error-based - WHERE or HAVING clause (XMLType)'
[05:06:49] [INFO] testing 'MySQL >= 5.0 error-based - Parameter replace'
[05:06:49] [INFO] testing 'MySQL inline queries'
[05:06:50] [INFO] testing 'PostgreSQL inline queries'
[05:06:51] [INFO] testing 'Microsoft SQL Server/Sybase inline queries'
[05:06:52] [INFO] testing 'MySQL > 5.0.11 stacked queries (SELECT - comment)'
[05:06:52] [WARNING] time-based comparison requires larger statistical model, please wait. (done)
[05:06:53] [INFO] testing 'PostgreSQL > 8.1 stacked queries (comment)'
[05:06:53] [INFO] testing 'Microsoft SQL Server/Sybase stacked queries (comment)'
[05:06:54] [INFO] testing 'Oracle stacked queries (DBMS_PIPE.RECEIVE_MESSAGE - comment)'
[05:06:55] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (SELECT)'
[05:06:56] [INFO] testing 'PostgreSQL > 8.1 AND time-based blind'
[05:06:56] [INFO] testing 'Microsoft SQL Server/Sybase time-based blind'
[05:06:57] [INFO] testing 'Oracle AND time-based blind'
[05:06:58] [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns'
[05:06:58] [INFO] automatically extending ranges for UNION query injection technique tests as there is at least one other (potential) technique found
[05:07:12] [INFO] testing 'MySQL UNION query (NULL) - 1 to 20 columns'
[05:07:27] [INFO] checking if the injection point on (custom) HEADER parameter 'Cookie #1*' is a false positive
[05:07:29] [WARNING] false positive or unexploitable injection point detected
[05:07:29] [WARNING] (custom) HEADER parameter 'Cookie #1*' is not injectable
[05:07:29] [WARNING] HTTP error codes detected during run:
400 (Bad Request) - 10 times, 500 (Internal Server Error) - 72 times, 406 (Not Acceptable) - 3 times
        _
___ ___| |_____ ___ ___  {1.0.4.0#dev}
|_ -| . | |     | .'| . |
|___|_  |_|_|_|_|__,|  _|
      |_|           |_|   http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting at 05:08:44
```

Wide vs Narrow Recon



Depending on your security testing engagement (bounty, pentest, w/e) it's important to understand your scope and what kind of testing you prefer to do.

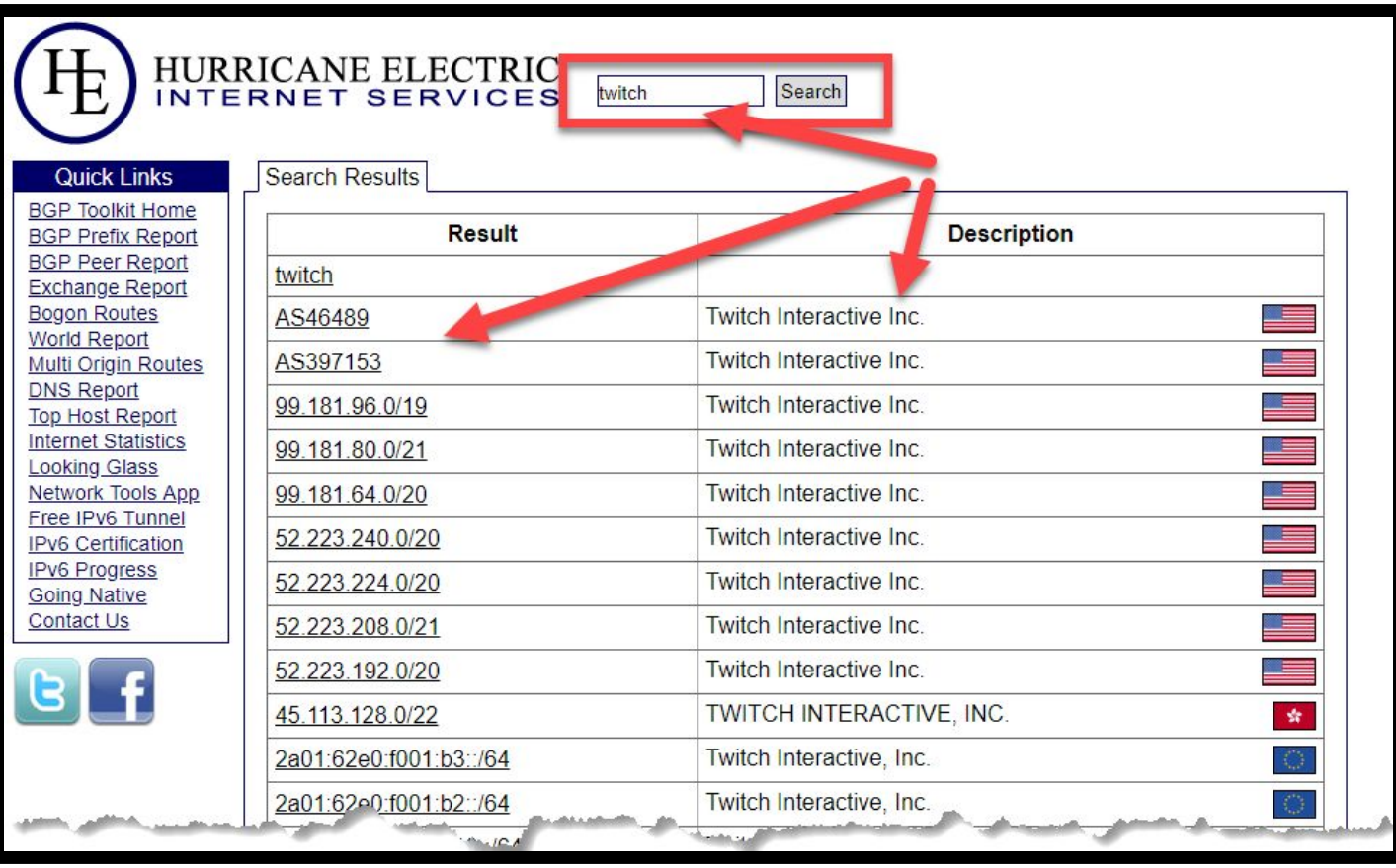
Going "wide" results in finding sites that have often been left less secure than a "flagship" application. Some of these wide scope sites may be worth less due to their threat profile to the entity but bugs are plentiful.

Focussing "narrow" is more involved and yields much higher payouts (normally) but requires you to invest a lot of time to understand the application.

Wide Recon

```
[05:06:24] [INFO] loading tamper script 'versionedmorekeywords'
[05:06:24] [WARNING] tamper script 'versionedmorekeywords' is only meant to be run against MySQL >=
5.1.13
[05:06:24] [INFO] loading tamper script 'xforwardedfor'
[05:06:24] [WARNING] using too many tamper scripts is usually not a good idea
custom injection marking character ('*') found in option '--headers/--user-agent/--referer/--cookie'
. Do you want to process it? [Y/n/q] Y
[05:06:28] [INFO] testing connection to the target URL
[05:06:29] [WARNING] the web server responded with an HTTP error code (406) which could interfere wi
th the results of the tests
[05:06:29] [INFO] testing if the target URL is stable
[05:06:29] [INFO] target URL is stable
[05:06:29] [INFO] testing if (custom) HEADER parameter 'Cookie #1*' is dynamic
[05:06:29] [WARNING] currently only couple of keywords are being processed ('UNION', 'SELECT', 'INSE
RT', 'UPDATE', 'FROM', 'WHERE'). You can set it manually according to your needs
[05:06:30] [WARNING] reflective value(s) found and filtering out
[05:06:30] [INFO] confirming that (custom) HEADER parameter 'Cookie #1*' is dynamic
[05:06:31] [INFO] (custom) HEADER parameter 'Cookie #1*' is dynamic
[05:06:31] [WARNING] heuristic (basic) test shows that (custom) HEADER parameter 'Cookie #1*' might
not be injectable
[05:06:34] [INFO] testing for SQL injection on (custom) HEADER parameter 'Cookie #1*'
[05:06:34] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[05:06:39] [INFO] (custom) HEADER parameter 'Cookie #1*' seems to be 'AND boolean-based blind - WHER
E or HAVING clause' injectable (with --string="\\xa0\\xa0\\xa0\\xa0\n\tat com.ibm.ws.http.channel.in
bound.impl.HttpInboundLink.ready(HttpInboundLink.java:287)")
[05:06:48] [INFO] testing 'MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause
'
[05:06:48] [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'
[05:06:49] [INFO] testing 'Microsoft SQL Server/Sybase AND error-based - WHERE or HAVING clause'
[05:06:49] [INFO] testing 'Oracle AND error-based - WHERE or HAVING clause (XMLType)'
[05:06:49] [INFO] testing 'MySQL >= 5.0 error-based - Parameter replace
[05:06:49] [INFO] testing 'MySQL inline queries'
[05:06:50] [INFO] testing 'PostgreSQL inline queries'
[05:06:51] [INFO] testing 'Microsoft SQL Server/Sybase inline queries'
[05:06:52] [INFO] testing 'MySQL > 5.0.11 stacked queries (SELECT - comment)'
[05:06:52] [WARNING] time-based comparison requires larger statistical model, please wait. (done)
[05:06:53] [INFO] testing 'PostgreSQL > 8.1 stacked queries (comment)'
[05:06:53] [INFO] testing 'Microsoft SQL Server/Sybase stacked queries (comment)'
[05:06:54] [INFO] testing 'Oracle stacked queries (DBMS_PIPE.RECEIVE_MESSAGE - comment)'
[05:06:55] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (SELECT)'
[05:06:56] [INFO] testing 'PostgreSQL > 8.1 AND time-based blind'
[05:06:56] [INFO] testing 'Microsoft SQL Server/Sybase time-based blind'
[05:06:57] [INFO] testing 'Oracle AND time-based blind'
[05:06:58] [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns'
[05:06:58] [INFO] automatically extending ranges for UNION query injection technique tests as there
is at least one other (potential) technique found
[05:07:12] [INFO] testing 'MySQL UNION query (NULL) - 1 to 20 columns'
[05:07:27] [INFO] checking if the injection point on (custom) HEADER parameter 'Cookie #1*' is a fal
se positive
[05:07:29] [WARNING] false positive or unexploitable injection point detected
[05:07:29] [WARNING] (custom) HEADER parameter 'Cookie #1*' is not injectable
[05:07:29] [WARNING] HTTP error codes detected during run:
400 (Bad Request) - 10 times, 500 (Internal Server Error) - 72 times, 406 (Not Acceptable) - 3 times

        _
___ ___| |_____ ___ ___  {1.0.4.0#dev}
|_ -| . | |     | .'| . |
|___|_  |_|_|_|_|__,|  _|
      |_|           |_|   http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal.
 It is the end user's responsibility to obey all applicable local, state and federal laws. Developer
s assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting at 05:08:44
```

IP Enumeration: ASNs (wide scope recon)

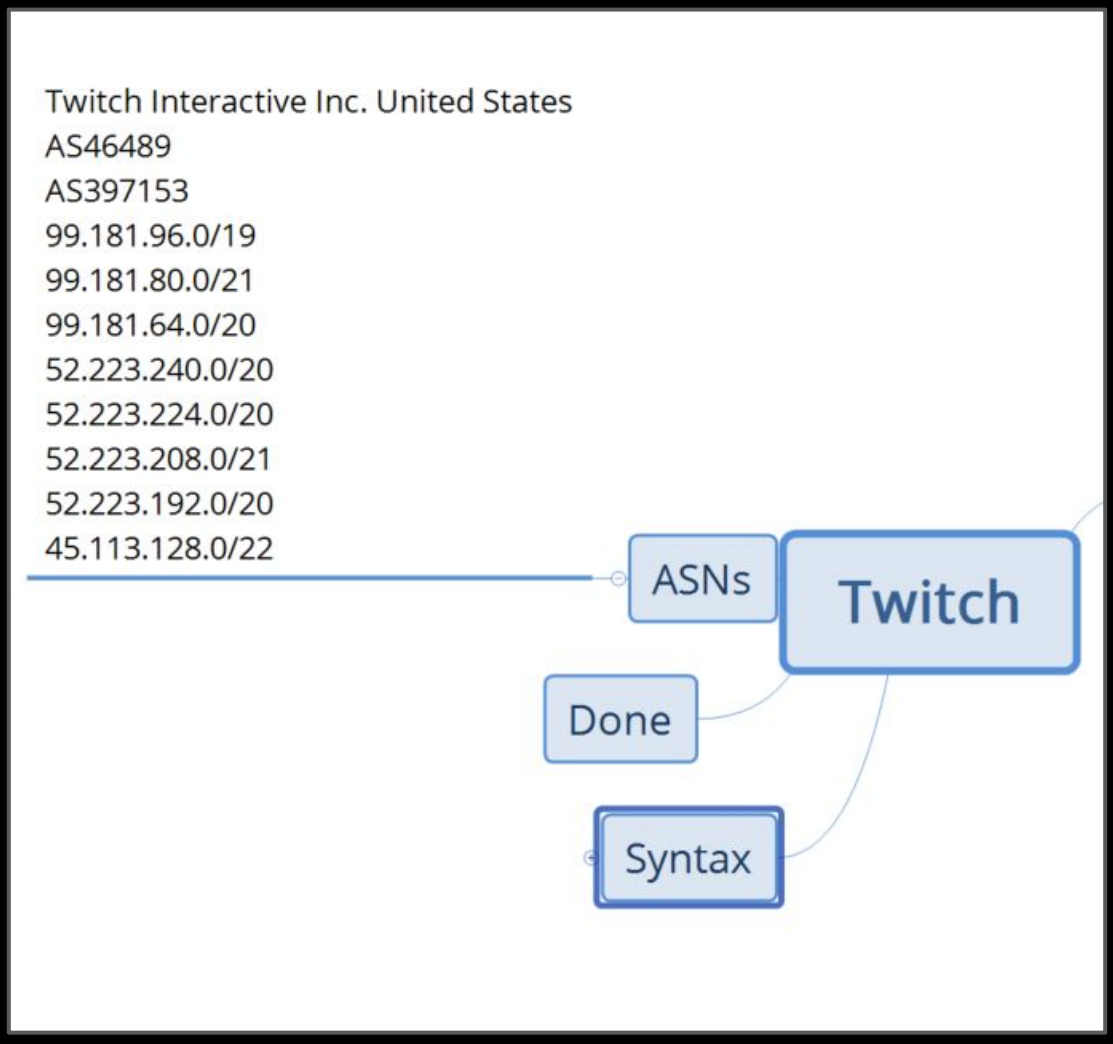


Autonomous System Numbers are given to large enough networks. These ASN's will help us track down some semblance of an entity's IT infrastructure. The most reliable way to get these is manually through Hurricane Electric's free-form search:

- http://bgp.he.net

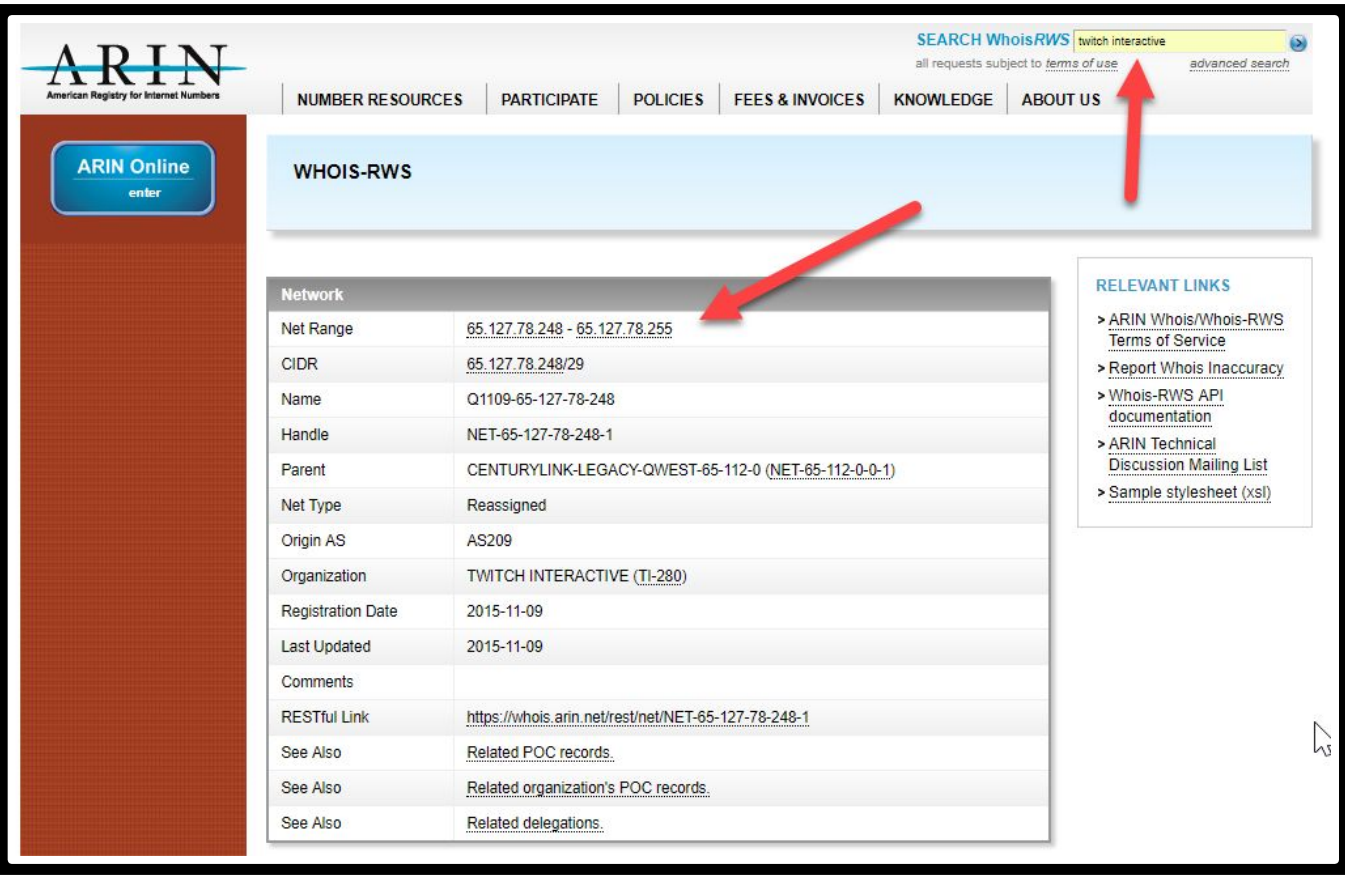
Because of the advent of cloud infrastructure, ASNs aren't always a complete picture of a network. Rogue assets could exist on cloud environments like AWS and Azure. Here we can see several IP ranges.

IP Enumeration: ASNs



Updates to our notes...

IP Enumeration: Registrars



ARIN and RIPE are Regional Registrars who allow full text searches for address space as well:

"As a Regional Internet Registry, we allocate and register blocks of Internet number resources to Internet service providers (ISPs) and other organisations in our geographical service region. These Internet number resources are mainly in the form of IPv4 and IPv6 address space and Autonomous System Numbers (ASNs)."

(US Region)

https://whois.arin.net/ui/query.do

(EU, Central Asia regions)
https://apps.db.ripe.net/db-web-ui/#/fulltextsearch

Other Infrastructure Enumeration

Shodan

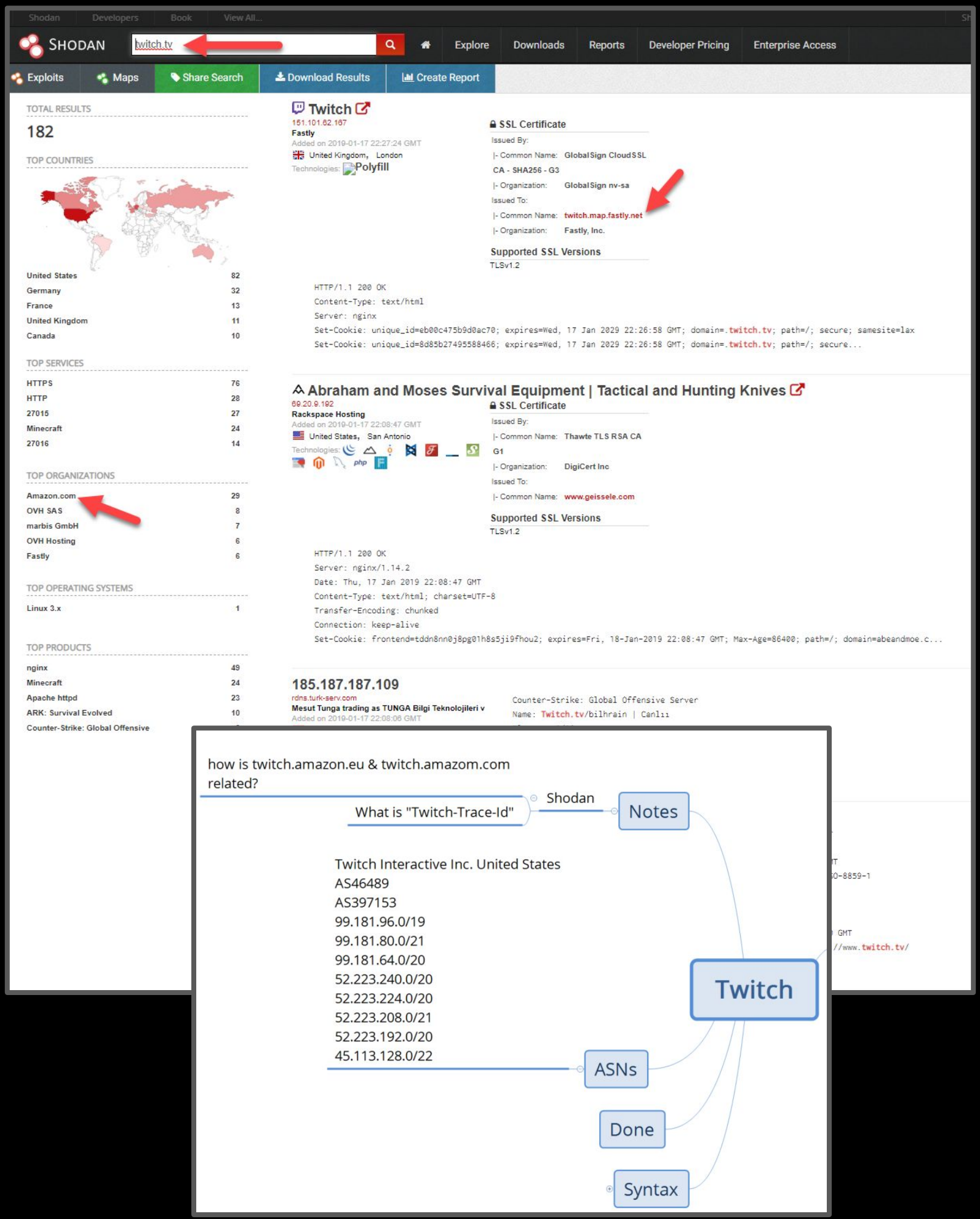
Shodan is a tool that continuously spiders infrastructure on the internet. It is much more verbose than regular spiders. It captures response data, cert data, stack profiling data, and more. It requires registration.

Example:

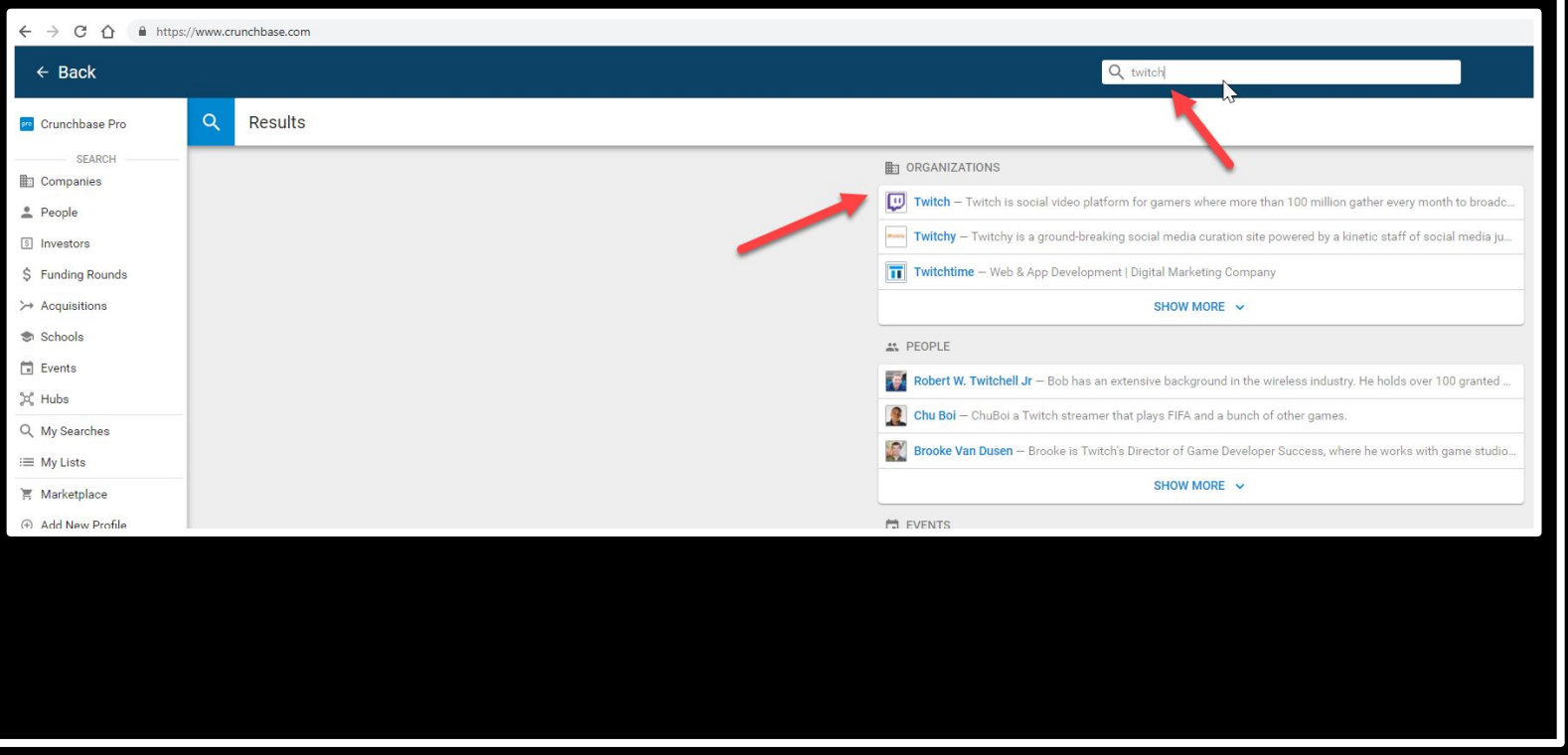
https://www.shodan.io/search?query=twitch.tv

....

Shodan


Shodan
Developers
Book
View All...
SHODAN
twitch.tv
Explore
Downloads
Reports
Developer Pricing
Enterprise Access
Exploits
Maps
Share Search
Download Results
Create Report
TOTAL RESULTS
182
TOP COUNTRIES
United States 82
Germany 32
France 13
United Kingdom 11
Canada 10
TOP SERVICES
HTTPS 76
HTTP 28
27015 27
Minecraft 24
27016 14
TOP ORGANIZATIONS
Amazon.com 29
OVH SAS 8
marbis GmbH 7
OVH Hosting 6
Fastly 6
TOP OPERATING SYSTEMS
Linux 3.x 1
TOP PRODUCTS
nginx 49
Minecraft 24
Apache httpd 23
ARK: Survival Evolved 10
Counter-Strike: Global Offensive
Twitch
151.101.62.167
Fastly
Added on 2019-01-17 22:27:24 GMT
United Kingdom, London
Technologies: Polyfill
SSL Certificate
Issued By:
|- Common Name: GlobalSign CloudSSL CA - SHA256 - G3
|- Organization: GlobalSign nv-sa
Issued To:
|- Common Name: twitch.map.fastly.net
|- Organization: Fastly, Inc.
Supported SSL Versions
TLSv1.2
HTTP/1.1 200 OK
Content-Type: text/html
Server: nginx
Set-Cookie: unique_id=eb00c475b9d0ac70; expires=Wed, 17 Jan 2029 22:26:58 GMT; domain=.twitch.tv; path=/; secure; samesite=lax
Set-Cookie: unique_id=8d85b27495588466; expires=Wed, 17 Jan 2029 22:26:58 GMT; domain=.twitch.tv; path=/; secure...
Abraham and Moses Survival Equipment | Tactical and Hunting Knives
69.20.9.192
Rackspace Hosting
Added on 2019-01-17 22:08:47 GMT
United States, San Antonio
Technologies:
SSL Certificate
Issued By:
|- Common Name: Thawte TLS RSA CA G1
|- Organization: DigiCert Inc
Issued To:
|- Common Name: www.geissele.com
Supported SSL Versions
TLSv1.2
HTTP/1.1 200 OK
Server: nginx/1.14.2
Date: Thu, 17 Jan 2019 22:08:47 GMT
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Connection: keep-alive
Set-Cookie: frontend=tddn8nn0j8pg01h8s5ji9fhou2; expires=Fri, 18-Jan-2019 22:08:47 GMT; Max-Age=86400; path=/; domain=abeandmoe.c...
185.187.187.109
rdns.turk-serv.com
Mesut Tunga trading as TUNGA Bilgi Teknolojileri v
Added on 2019-01-17 22:08:06 GMT
Counter-Strike: Global Offensive Server
Name: Twitch.tv/bilhrain | Canlıı
how is twitch.amazon.eu & twitch.amazom.com related?
What is "Twitch-Trace-Id"
Shodan
Notes
Twitch Interactive Inc. United States
AS46489
AS397153
99.181.96.0/19
99.181.80.0/21
99.181.64.0/20
52.223.240.0/20
52.223.224.0/20
52.223.208.0/21
52.223.192.0/20
45.113.128.0/22
ASNs
Twitch
Done
Syntax

Acquisitions: Crunchbase



So far we've been sniffing around an entity with one name. There are however entities that operate under many brand names or parent organizations who have acquired other brands. When these acquisitions happen, IT groups are folded together and infrastructure is decommissioned... or are they? When doing wide recon, it's important to look at this.

Cruncbase is a business analytics aggregator which tracks acquisitions.

We can use it to expand our scope:

- https://www.crunchbase.com/

It also offers a **wealth** of contextual data about our target.

Wikipedia can also have this data. We can also add ChatGPT!!

Overview

Acquired by **Amazon**

Number of Acquisitions **4**

Twitch

Twitch is social video platform for gamers where more than 100 million gather every month to broadcast, watch and talk about video games.

San Francisco, California, United States

Categories	Social Media, Video, Video Games, Video Streaming
Headquarters Regions	San Francisco Bay Area, West Coast, Western US
Sub-Organization of	Amazon
Founded Date	2007
Founders	Emmett Shear, Justin Kan, Kyle Vogt
Operating Status	Active
Funding Status	M&A
Last Funding Type	Series C
Number of Employees	251-500
Also Known As	twitch.tv, Twitch Interactive
Legal Name	Twitch Interactive, Inc.
IPO Status	Private
Company Type	For Profit
Website	www.twitch.tv/
Facebook	View on Facebook
LinkedIn	View on LinkedIn
Twitter	View on Twitter

Twitch is social video for gamers. It is a video platform and community for gamers where more than 100 million gather every month to broadcast, watch and talk about video games. Twitch's video platform is the backbone of both live and on-demand distribution for the entire video game ecosystem. This includes game developers, publishers, media...

Read More

Acquisitions

Number of Acquisitions **4**

Twitch has acquired 4 organizations. Their most recent acquisition was Revlo on Dec 13, 2018.

Which types of acquisition does this organization make most frequently? pro Show

Acquired Organization Name	Announced Date	Price	Transaction Name
Revlo	Dec 13, 2018	–	Revlo acquired by Twitch
ClipMine	Aug 17, 2017	–	ClipMine acquired by Twitch
Curse	Aug 16, 2016	–	Curse acquired by Twitch
GoodGame	Dec 9, 2014	–	GoodGame acquired by Twitch

Company Tech Stack by Siftery

Active Products

Twitch uses 61 technology products and services including Google Analytics, WordPress, and G Suite (formerly Google Apps for Work).

UNLOCK MORE TECHNOLOGIES DATA ›

Website Tech Stack by BuiltWith

Active Technology

Twitch is actively using 12 technologies for its website. These include SSL by Default, Content Delivery Network, and nginx.

UNLOCK WEBSITE TECHNOLOGIES DATA ›

Competitors & Revenue by Owler

Twitch has $30M in revenue annually. Twitch competes with Azubu, Dailymotion, and Plays.tv.

UNLOCK MORE COMPETITORS & REVENUE DATA ›

Current Team

Number of Current Team Members 40

Twitch has 40 current team members, including CEO and Founder Emmett Shear.

Emmett Shear
CEO and Founder

Justin Kan
Founder

Colin Carrier
Chief Strategy Officer

Sara Clemens
COO

Jonathan Simpson-Bint
CRO

Kym Nelson
SVP Sales

Khudor Annous
Head of Marketing & Partnerships

Mark Weiler
SVP, Head of Platform & Services

VIEW ALL ›

Past Team

Number of Past Team Members 7

Twitch has 7 past team members, including COO Kevin Lin.

Person Name	Title At Company	Start Date	End Date
Kevin Lin	COO	2008	2018
Christina Nguyen	UI/UX Designer	2016	2017
Anne Lin	4GAMERS(Twitch's agency) - Business Development Director	May 2015	Apr 2016
Andrew Shiffett	Internet Personality	Jan 2012	Feb 2014
Kyle Vogt	Co-Founder	Jun 2011	Oct 2013

VIEW ALL >

Events

Number of Events 17

Twitch has participated in 17 events. Recently, they attended TwitchCon 2018 on Oct 26, 2018.

TwitchCon 2018
Organizer
Oct 26, 2018

TwitchCon 2018
Sponsor
Oct 26, 2018

VidCon 2018
Sponsor
Jun 20, 2018

E3 2018
Sponsor
Jun 12, 2018

OMR Festival 2018
Exhibitor
Mar 22, 2018

Lesbians Who Tech London Summit 2017
Sponsor
Nov 10, 2017

TwitchCon 2017
Sponsor
Oct 20, 2017

TwitchCon 2017
Organizer
Oct 20, 2017

VIEW ALL >

Recent News & Activity

Number of Articles 4,189

Date	Activity
Dec 13, 2018	Twitch acquired Revlo for an undisclosed amount
Nov 12, 2018	Twitch: TechCrunch — What's next? The top media executives on the job market
Nov 6, 2018	Twitch: TechCrunch — How to stream U.S. elections coverage if you don't have TV
Oct 26, 2018	Twitch: TechCrunch — Twitch announces group streaming and a karaoke game for its 1M concurrent viewers
Oct 26, 2018	Twitch: TechCrunch — Snapchat's new Camera desktop camera app brings AR masks to Twitch, Skype...
Oct 25, 2018	Twitch: TechCrunch — YouTube is closing the gap with Twitch on live streaming, report finds
Oct 17, 2018	Twitch: TechCrunch — These are the most successful companies to emerge from Y Combinator
Sep 22, 2018	Twitch: Digital Trends — How to sign up for Amazon Prime
Sep 17, 2018	Twitch: GameSpot — Overwatch: Free Golden Loot Box For Amazon / Twitch Prime Members
Sep 14, 2018	Twitch: TechCrunch — Twitch updates security for its TwitchCon event following the Jacksonville esports shooting

VIEW ALL >

Twitter

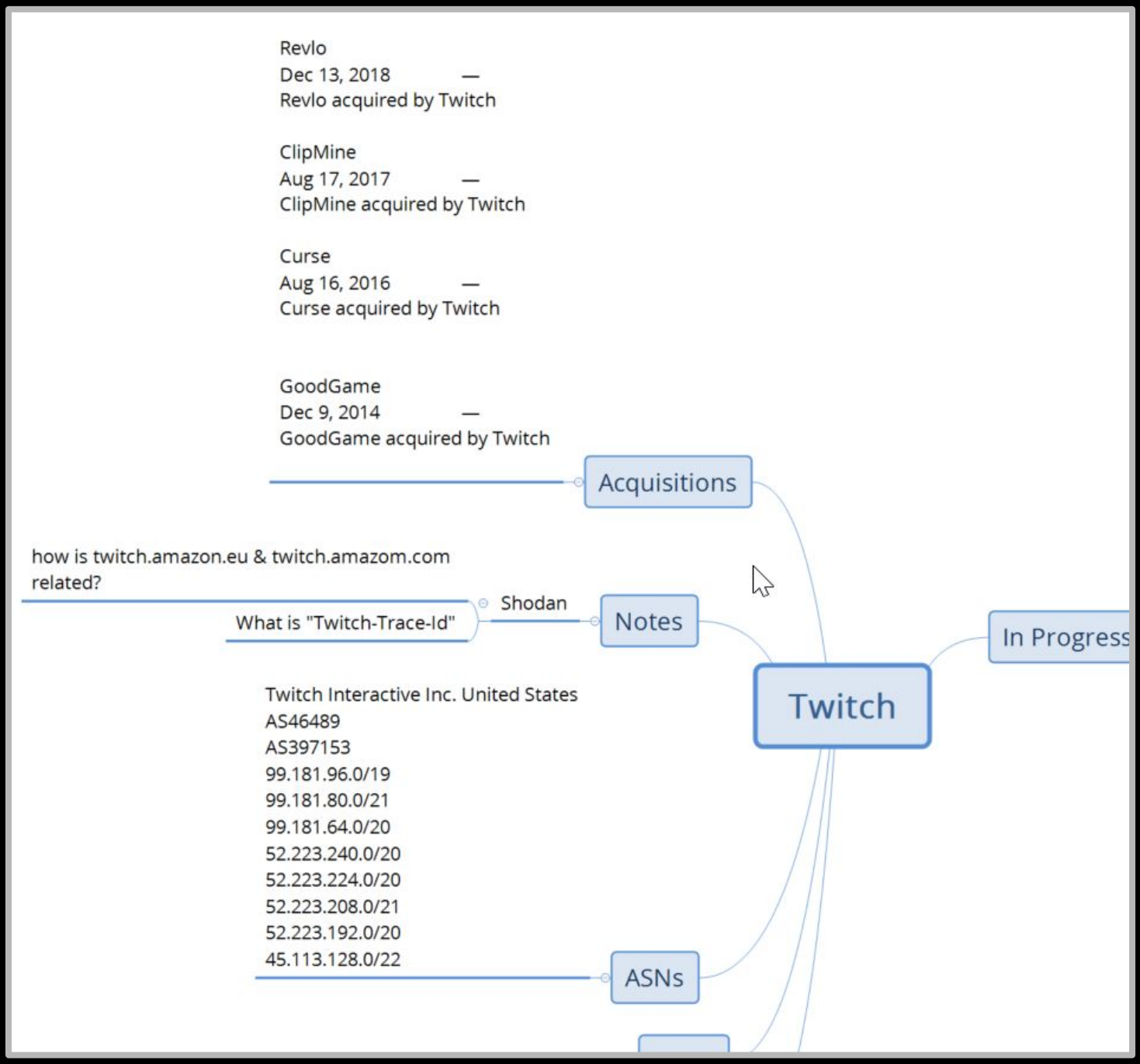
Jan 14, 2019

Twitch
@Twitch

The dark spirit thanked @Asmongold for the follow.

Asmongold Gets Jebaited

Acquisitions



Updates to our notes...

Linked Discovery (Burp Suite)

Another way to to widen our scope is to examine all the links of our main target. We can do this using Burp Suite.

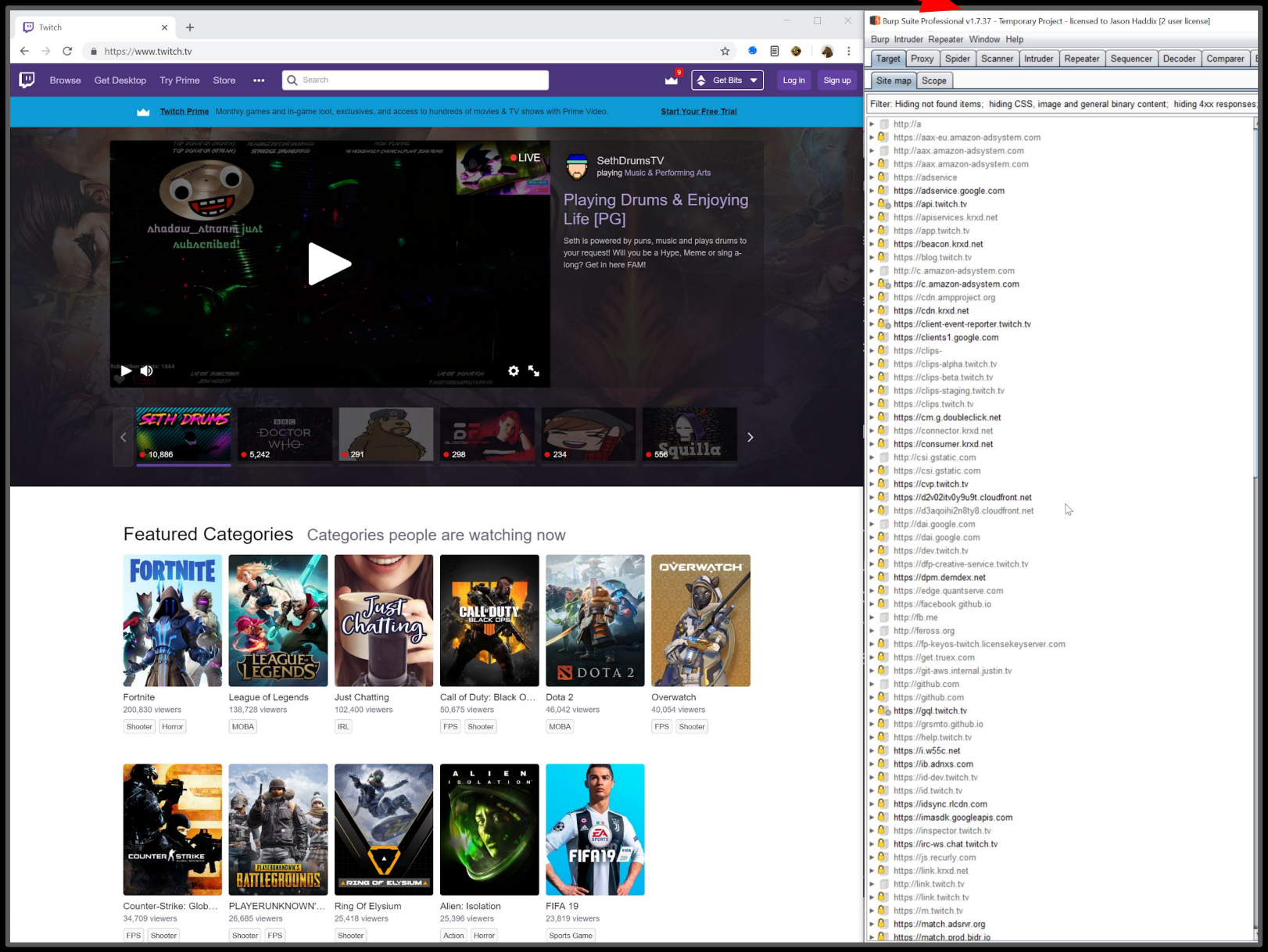
We can then recursively spider all those links for a term with regex, examining those links... and their links, and so on... until we have found all sites that could be in our scope.

Instructor DEMO

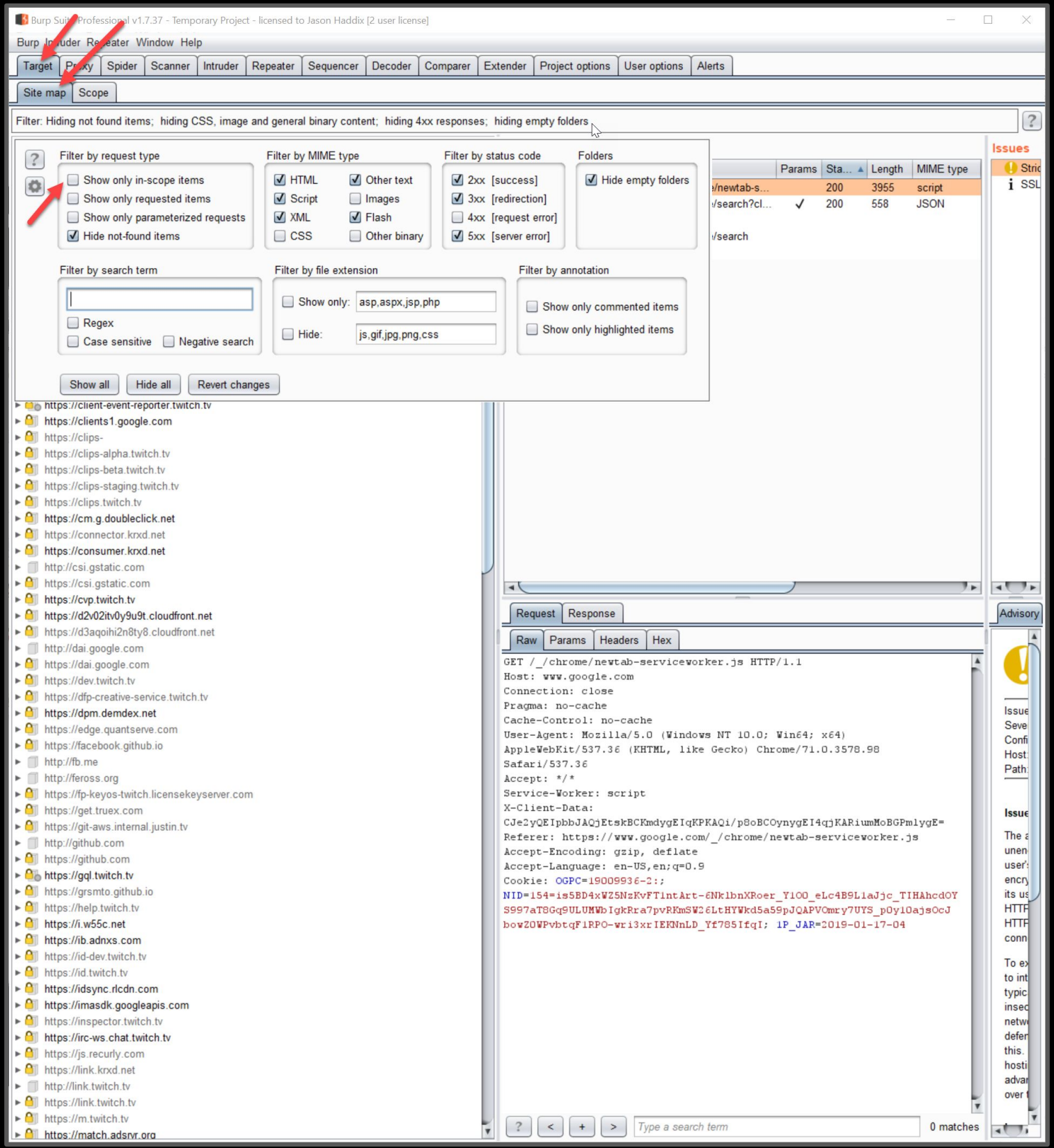
1) Turn off passive scanning
2) Set forms auto to submit (if you're feeling frisky)
3) Set scope to advanced control and use "keyword" of target name (not a normal FQDN)
4) Walk+browse main site, then spider all hosts recursively!
5) Profit

Linked Discovery (Burp Suite)

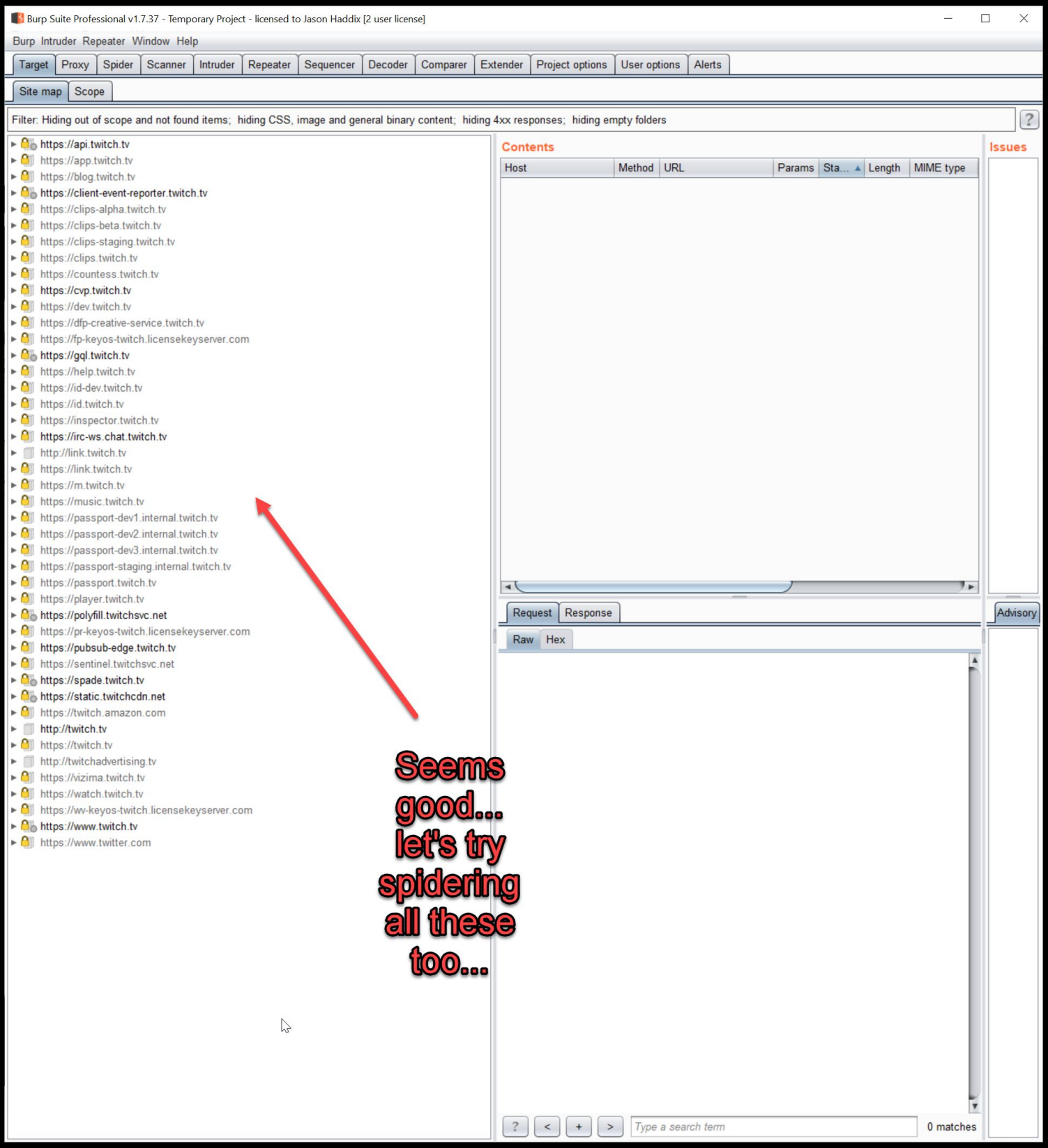
Burp Site Tree after 1 request...



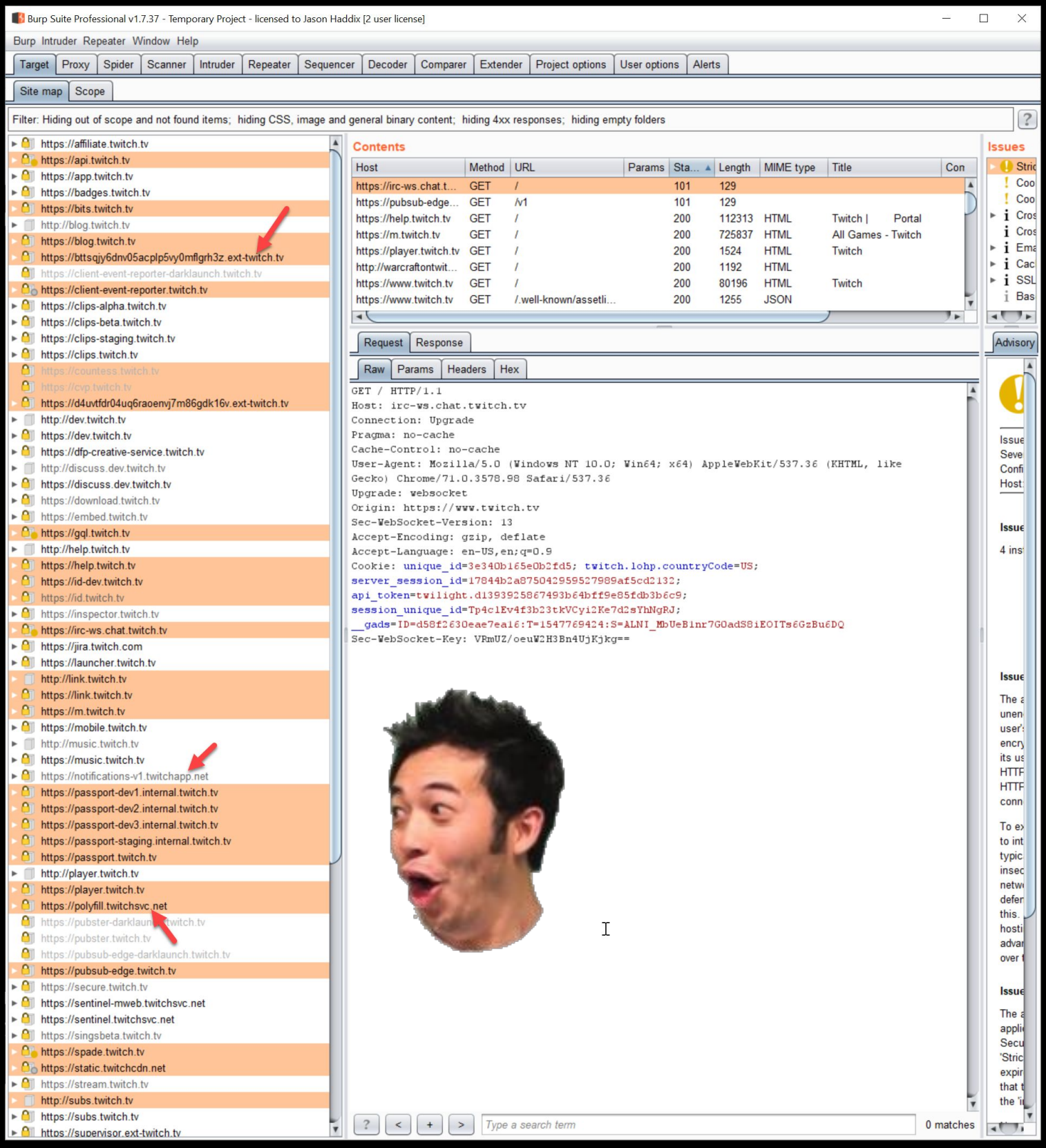
Linked Discovery (Burp Suite)



Linked Discovery (Burp Suite)



Linked Discovery (Burp Suite)



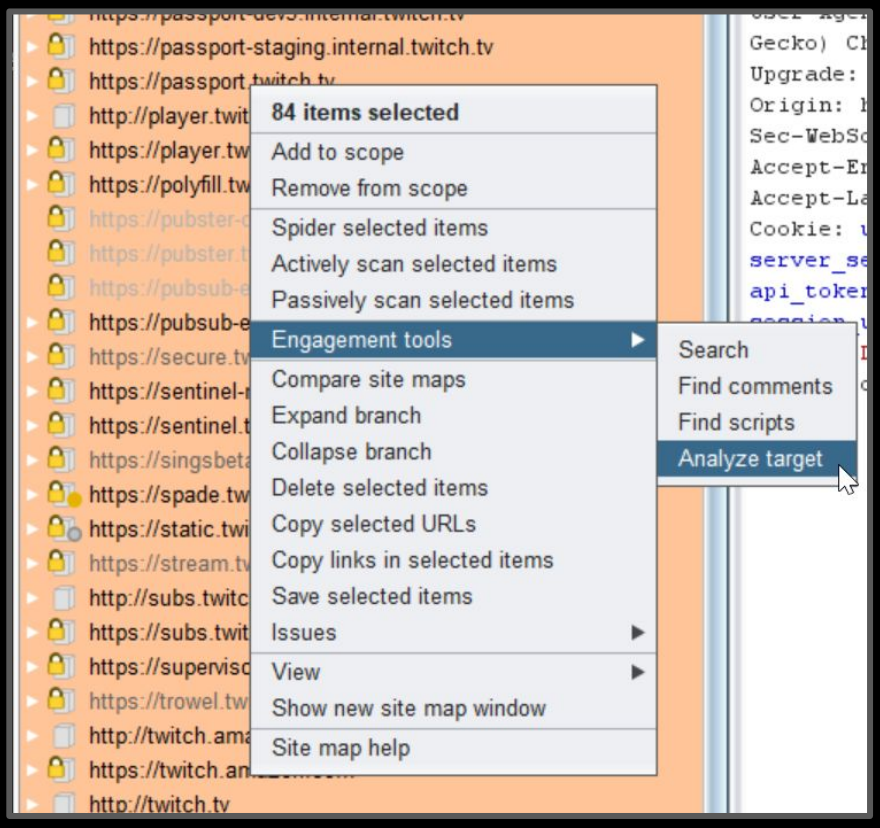
We've now discovered a TON of linked URLs that belong to our entity. Not only subdomains, but completely **NEW** domains we can do analysis later on in the subdomain discovery section. We have increased scope by over 9000! We can also now spider these hosts.

Linked Discovery (Burp Suite)

Now that we have this data, how do we export it?

Clumisily =(

1) Select all hosts in the site tree
2) In **PRO ONLY right click the selected hosts**
3) **Go to “Engagement Tools” -> “Analyze target”**
4) **Save report as an html file**
5) **Copy the hosts from the “Target” section**

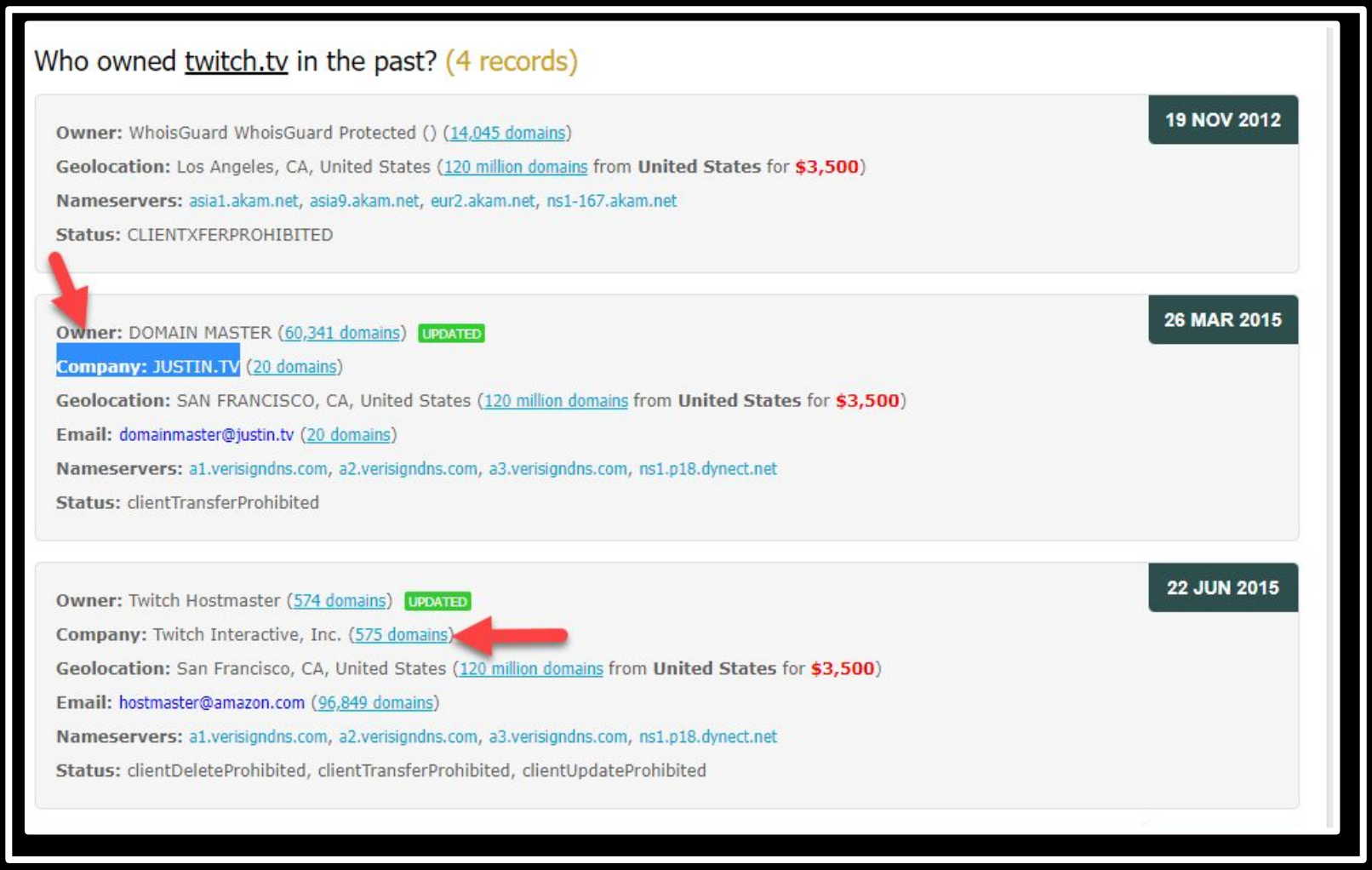


Target analysis

Targets

- https://affiliate.twitch.tv/
- https://api.twitch.tv/
- https://app.twitch.tv/
- https://badges.twitch.tv/
- https://bits.twitch.tv/
- http://blog.twitch.tv/
- https://blog.twitch.tv/
- https://bttsqjy6dnv05acplp5vy0mflgrh3z.ext-twitch.tv/
- https://client-event-reporter-darklaunch.twitch.tv/
- https://client-event-reporter.twitch.tv/
- https://clips-alpha.twitch.tv/
- https://clips-beta.twitch.tv/
- https://clips-staging.twitch.tv/
- https://clips.twitch.tv/
- https://countess.twitch.tv/
- https://cvp.twitch.tv/
- https://d4uvtfdr04uq6raoenvj7m86gdk16v.ext-twitch.tv/
- http://dev.twitch.tv/
- https://dev.twitch.tv/
- https://dfp-creative-service.twitch.tv/
- http://discuss.dev.twitch.tv/
- https://discuss.dev.twitch.tv/
- https://download.twitch.tv/
- https://embed.twitch.tv/
- https://gql.twitch.tv/
- http://help.twitch.tv/
- https://help.twitch.tv/
- https://id-dev.twitch.tv/
- https://id.twitch.tv/
- https://inspector.twitch.tv/
- https://irc-ws.chat.twitch.tv/
- https://jira.twitch.com/
- https://launcher.twitch.tv/
- http://link.twitch.tv/
- https://link.twitch.tv/
- https://m.twitch.tv/
- https://mobile.twitch.tv/
- http://music.twitch.tv/
- https://music.twitch.tv/
- https://notifications-v1.twitchapp.net/
- https://passport-dev1.internal.twitch.tv/
- https://passport-dev2.internal.twitch.tv/
- https://passport-dev3.internal.twitch.tv/
- https://passport-staging.internal.twitch.tv/
- https://passport.twitch.tv/
- http://player.twitch.tv/

Reverse WHOIS analysis: WHOXY.com

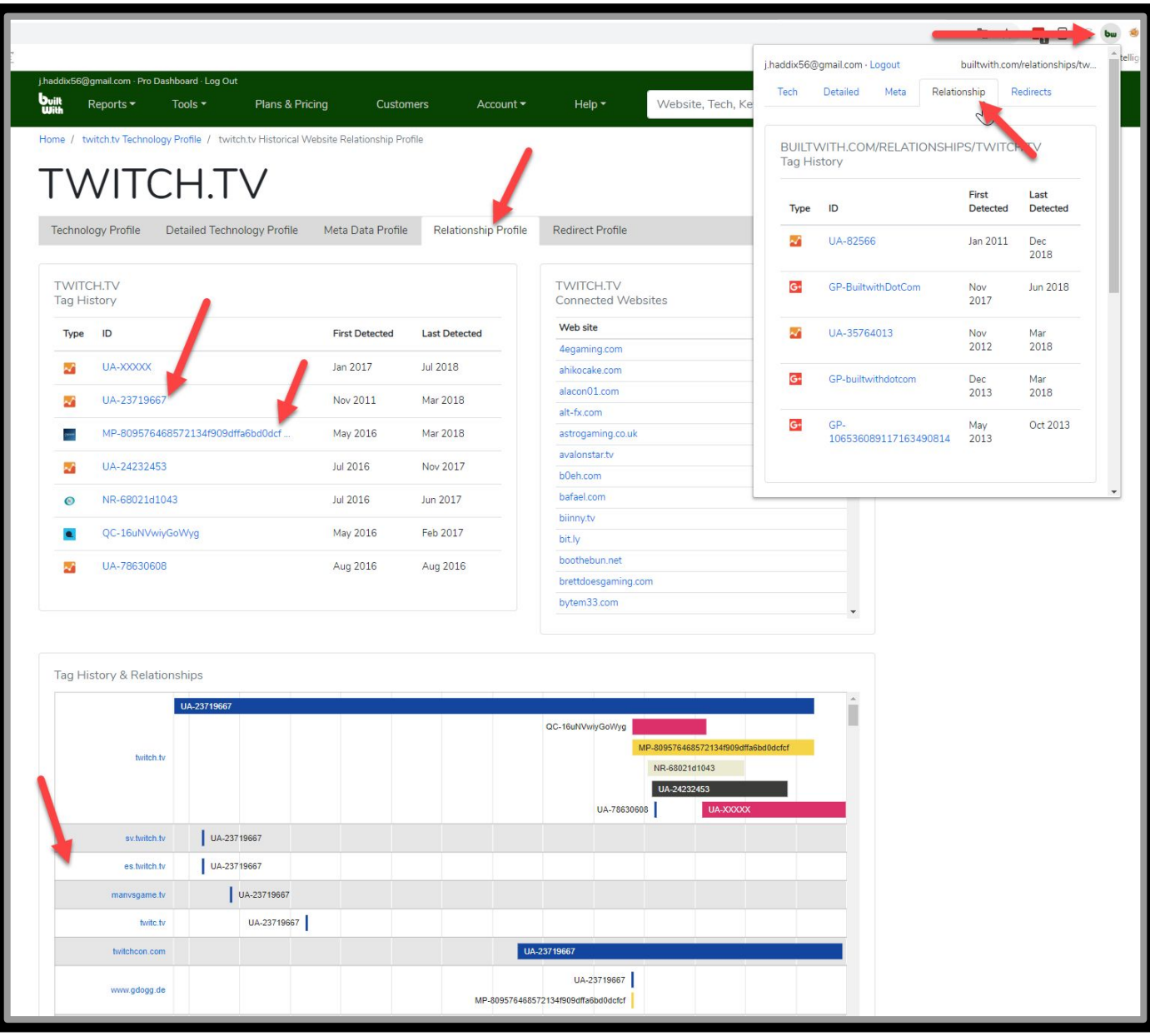


Every registered website has some registration info on file with the registrars. Two key pieces of data we can use are Organization name and any emails in the WHOIS data. To do this you need access to a large WHOIS database. WHOXY.com is one such database.

DOMLink is a tool written by Vincent Yiu (@vysecurity) which will recursively query the WHOXY WHOIS API. It will start by querying our targets WHOIS record, then analyze all the data and look for other records which comaint the organization name or any emails in the record. It does this until it finds no more records of match. You can also just use whoxy.com in this fashion, after you register and your free API key:

- http://api.whoxy.com/?key=APIkeyHERE&reverse=whois&name=Twitch+Hostmaster

Ad/Analytics Relationships



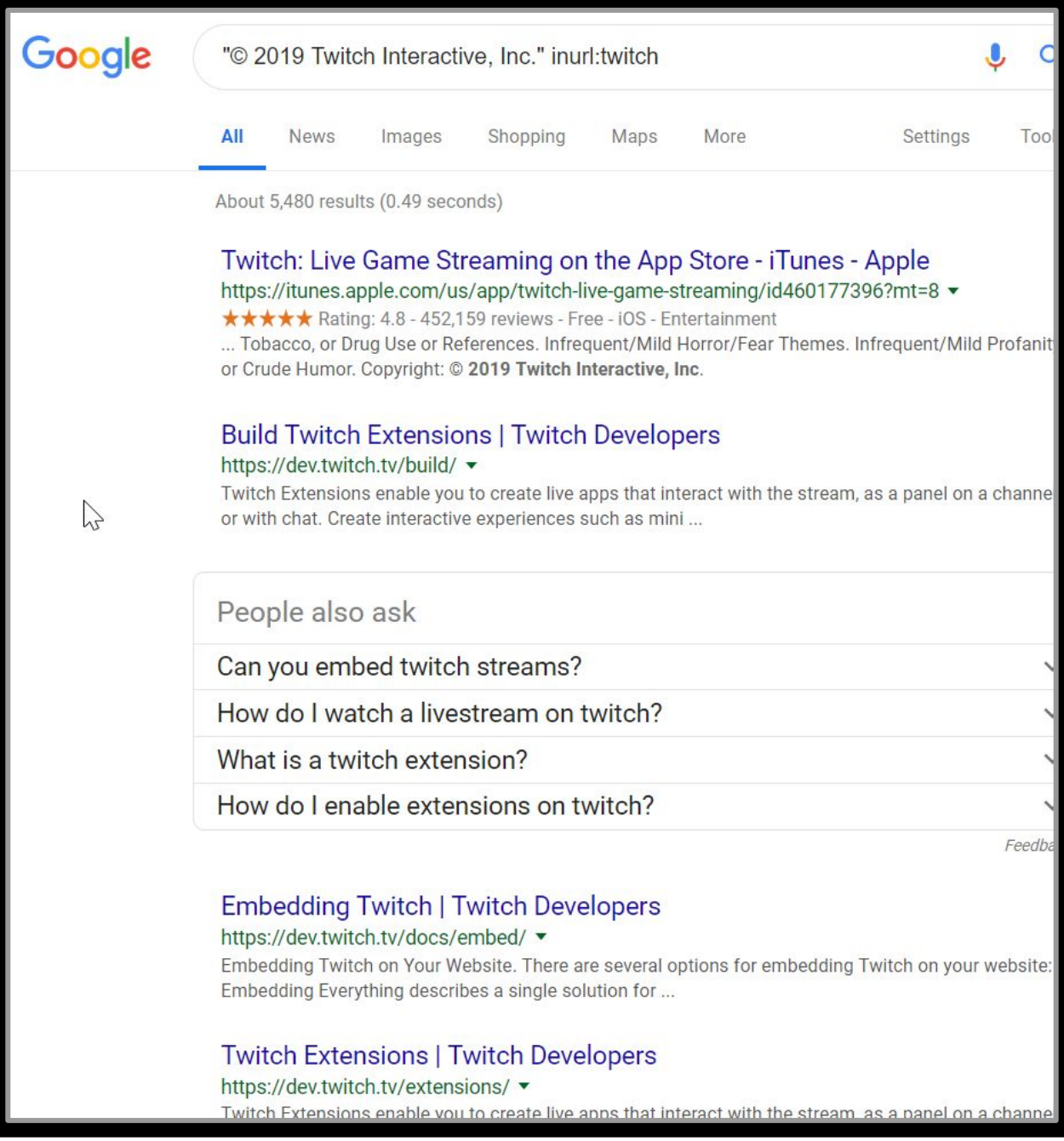
You can also glean related domains and subdomains by looking at a target's ad/analytics tracker codes. Many sites use the same codes across all their domains. Google analytics and New Relic codes are the most common. We can look at these "relationships" via a site called BuiltWith. Builtwith also has a Chrome and Firefox extension to do this on the fly.

- https://builtwith.com/relationships/twitch.tv

BuiltWith is also a tool we'll use to profile the technology stack of a target in later slides.

Google-fu Trademark and Others

"© 2019 Twitch Interactive, Inc." inurl:twitch
"© 2018 Twitch Interactive, Inc." inurl:twitch



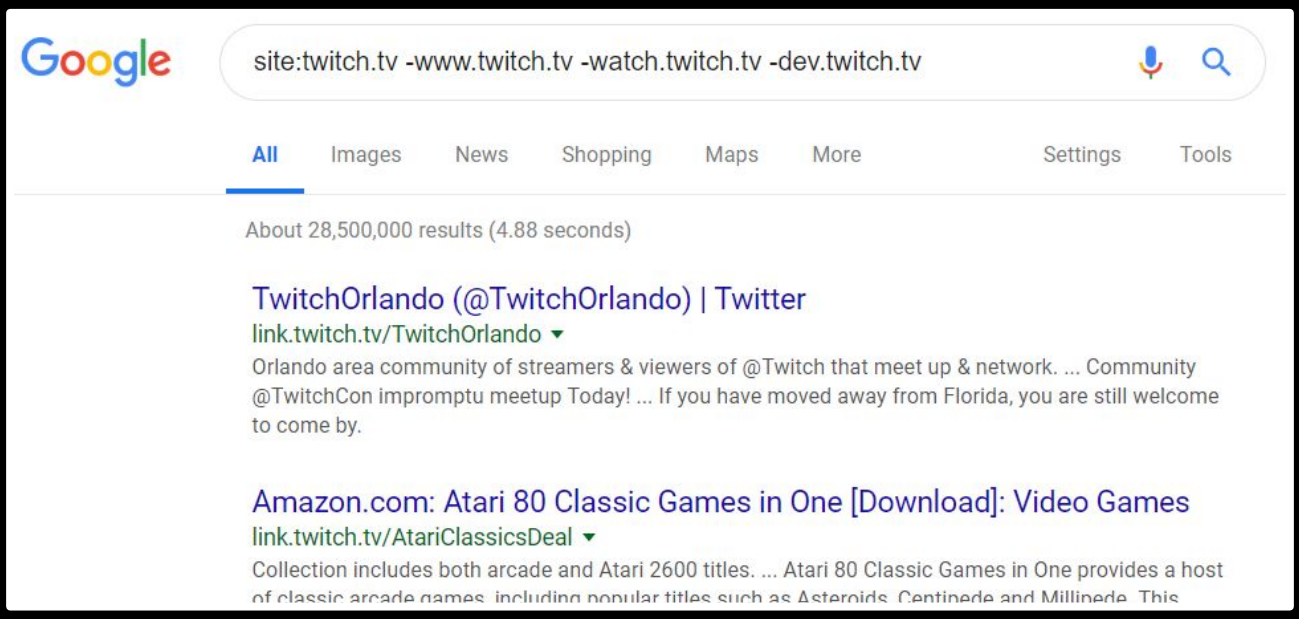
You can Google the copyright and terms of service text from a main target to glean related hosts.

Subdomain Enumeration

(still wide recon)

Web scraping for subdomains

1. site:twitch.tv -www.twitch.tv
2. site:twitch.tv -www.twitch.tv -watch.twitch.tv
3. site:twitch.tv -www.twitch.tv -watch.twitch.tv -dev.twitch.tv
4. ...



Domain and URL information gets used across the internet for a multitude of reasons. There are all sorts of data projects that expose databases of URLs or domains they store. Some examples:

- Google, Yahoo!, Bing, Baidu, Dogpile, Ask ++ are search engines and cache all urls their bots spider. This is used to make their search engines work.
- Robtex/itools, DNSDB, dnsdumpster, PTRarchve, and netcraft are search engines for DNS or infrastructure information.
- Censys.io, hackertarget, SecurityTrails, ThreatCrowd, ThreatMiner are all sites dedicated to monitoring internet assets with a focus on security.
- Sslmate CertSpotter, CertDB, and crt.sh are all projects related to SSL certificate gathering and transparency.
- VirusTotal parses out domain information for every submitted file, and archive.org takes snapshots of website and URL history over time.

In a wide scope project, we want to query these sites and API to discover what subdomains they might know about related to our domain. Luckily we don't have to do this manually. There are tools for this.

A manual example of using Google for this is shown in the image.

Amass

```
root@Test2:~/tools/amass# amass -d twitch.tv
twitch.tv
passport-external.aws.twitch.tv
gql.twitch.tv
pubsub-edge.twitch.tv
pubsub-edge.chat.twitch.tv
passport.twitch.tv
www.twitch.tv
m.twitch.tv
irc-ws-edge.chat.twitch.tv
irc-ws.chat.twitch.tv
app.twitch.tv
download.twitch.tv
discuss.dev.twitch.tv
invite.twitch.tv
join.twitch.tv
edgecast.hls.twitch.tv has a static DNS wildcard
blog.twitch.tv
polls.twitch.tv
th.blog.twitch.tv
link.twitch.tv
servers.twitch.tv
cis.blog.twitch.tv
graphql.prod.us-west2.twitch.tv
it.blog.twitch.tv
rc.twitch.tv
de.blog.twitch.tv
tr.blog.twitch.tv
release.twitch.tv
nl.blog.twitch.tv
canary.twitch.tv
ccu.event-engineering.twitch.tv
pong.prod.us-west2.twitch.tv
jp.blog.twitch.tv
event-panel.event-engineering.twitch.tv
assets.help.twitch.tv
uploads-regional.twitch.tv
websub-test-proxy.twitch.tv
```

For scraping subdomain data there are two industry leading tools at the moment, Amass and Subfinder. They parse all the “sources” referenced in the previous slide, and more. Amass has extensible output, bruteforcing, and more.

Amass is written by Jeff Foley

Amass

Amass also corresponds these scraped domains to ASNs and lists what network ranges they appeared in.

Useful.

```
Average DNS names processed: 49/sec

OWASP Amass v2.9.0                                https://github.com/OWASP/Amass
--------------------------------------------------------------------------------
439 names discovered - alt: 77, dns: 3, cert: 105, archive: 24, scrape: 218, api: 12
--------------------------------------------------------------------------------
ASN: 54113 - FASTLY - Fastly, US
        151.101.64.0/22         2    Subdomain Name(s)
        151.101.0.0/22          2    Subdomain Name(s)
        151.101.188.0/22        52   Subdomain Name(s)
        151.101.40.0/22         1    Subdomain Name(s)
        151.101.244.0/22        1    Subdomain Name(s)
        151.101.192.0/22        2    Subdomain Name(s)
        151.101.128.0/22        2    Subdomain Name(s)
ASN: 16509 - AMAZON-02 - Amazon.com, Inc., US
        52.24.0.0/14            35   Subdomain Name(s)
        52.84.48.0/23           4    Subdomain Name(s)
        13.35.8.0/23            4    Subdomain Name(s)
        34.208.0.0/12           15   Subdomain Name(s)
        35.160.0.0/13           16   Subdomain Name(s)
        54.254.128.0/17         1    Subdomain Name(s)
        54.186.0.0/15           9    Subdomain Name(s)
        52.18.0.0/15            2    Subdomain Name(s)
        52.36.0.0/14            6    Subdomain Name(s)
        35.178.0.0/15           1    Subdomain Name(s)
        2600:9000:202d::/48     54   Subdomain Name(s)
        52.40.0.0/14            5    Subdomain Name(s)
        54.192.144.0/22         12   Subdomain Name(s)
        52.220.0.0/15           1    Subdomain Name(s)
        54.171.0.0/16           2    Subdomain Name(s)
        3.8.0.0/14              1    Subdomain Name(s)
        54.192.12.0/22          55   Subdomain Name(s)
        2600:9000:2001::/48     26   Subdomain Name(s)
        54.148.0.0/15           10   Subdomain Name(s)
        2600:9000:204b::/48     8    Subdomain Name(s)
        52.10.0.0/15            3    Subdomain Name(s)
        2600:9000:201d::/48     8    Subdomain Name(s)
        52.84.44.0/22           1    Subdomain Name(s)
        52.88.0.0/15            7    Subdomain Name(s)
        2600:9000:2145::/48     8    Subdomain Name(s)
        184.169.128.0/17        1    Subdomain Name(s)
        52.208.0.0/13           3    Subdomain Name(s)
        52.32.0.0/14            11   Subdomain Name(s)
        13.35.124.0/22          44   Subdomain Name(s)
        50.112.128.0/19         3    Subdomain Name(s)
        52.52.0.0/15            3    Subdomain Name(s)
        54.215.128.0/18         2    Subdomain Name(s)
        54.68.0.0/15            2    Subdomain Name(s)
        54.191.0.0/16           3    Subdomain Name(s)
        54.70.0.0/15            2    Subdomain Name(s)
        54.200.0.0/15           6    Subdomain Name(s)
        50.18.0.0/18            1    Subdomain Name(s)
ASN: 0 - Private Networks
        10.0.0.0/8              66   Subdomain Name(s)
ASN: 46489 - JUSTINTV - Twitch Interactive Inc., US
        52.223.240.0/20         21   Subdomain Name(s)
        192.16.64.0/21          32   Subdomain Name(s)
        99.181.64.0/20          2    Subdomain Name(s)
        52.223.224.0/20         27   Subdomain Name(s)
        45.113.128.0/22         10   Subdomain Name(s)
        192.108.239.0/24        4    Subdomain Name(s)
        52.223.208.0/21         28   Subdomain Name(s)
        199.9.248.0/21          164  Subdomain Name(s)
        185.42.204.0/22         15   Subdomain Name(s)
        23.160.0.0/24           8    Subdomain Name(s)
        52.223.192.0/20         11   Subdomain Name(s)
ASN: 40341 - Q9-AS-CAL2 - Q9 Networks Inc., CA
        162.219.8.0/21          1    Subdomain Name(s)
ASN: 14618 - AMAZON-AES - Amazon.com, Inc., US
        54.84.0.0/15            1    Subdomain Name(s)
        52.72.0.0/15            1    Subdomain Name(s)
        52.0.0.0/15             71   Subdomain Name(s)
        2600:1f18::/33          1    Subdomain Name(s)
        52.2.0.0/15             2    Subdomain Name(s)
        18.204.0.0/14           3    Subdomain Name(s)
        52.200.0.0/13           4    Subdomain Name(s)
        52.4.0.0/14             137  Subdomain Name(s)
        52.20.0.0/14            1    Subdomain Name(s)
ASN: 395224 - BITLY-AS - Bitly Inc, US
        67.199.248.0/24         4    Subdomain Name(s)
ASN: 15169 - GOOGLE - Google LLC, US
        35.185.0.0/19           1    Subdomain Name(s)
ASN: 22606 - EXACT-7 - ExactTarget, Inc., US
        13.111.18.0/24          5    Subdomain Name(s)
        13.111.19.0/24          1    Subdomain Name(s)
        13.111.20.0/24          1    Subdomain Name(s)
        13.111.97.0/24          1    Subdomain Name(s)
ASN: 11377 - SENDGRID - SendGrid, Inc., US
        167.89.64.0/19          1    Subdomain Name(s)
ASN: 38895 - AMAZON-AS-AP Amazon.com Tech Telecom, JP
        2600:9000:20c7::/48     8    Subdomain Name(s)
```

Subfinder

Subfinder

I use both tools and concatenate and uniq the outputs.

```
~/recon/SNI # subfinder -d nasa.gov
[INF] Detected old /root/.config/subfinder/confi
nder/provider-config.yaml

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

                projectdiscovery.io

[INF] Current subfinder version v2.5.7 (latest)
[INF] Loading provider config from the default l
[INF] Enumerating subdomains for nasa.gov
bh363bl-30092762.ndc.nasa.gov
arsla16020872.ndc.nasa.gov
guest-portal.nasa.gov
ecc.earthdata.nasa.gov
lm000647901.ndc.nasa.gov
arlmhelp1.ndc.nasa.gov
arsvhphv02.ndc.nasa.gov
wicfd01.ndc.nasa.gov
arlmmacstg1.ndc.nasa.gov
sbir.nasa.gov
risk.eva.staging.appdat.jsc.nasa.gov
quark1.ndc.nasa.gov
wicchost1.ndc.nasa.gov
ndmaisc02.ndc.nasa.gov
csi.ndclab.nasa.gov
cantabria.mdscc.nasa.gov
hpjscres01.ndc.nasa.gov
mcsdexter11.ndc.nasa.gov
```

Brute force for Subdomains

```
root@Test2:~# host thistotallydoesntexist.twitch.com
Host thistotallydoesntexist.twitch.com not found: 3(NXDOMAIN)
```

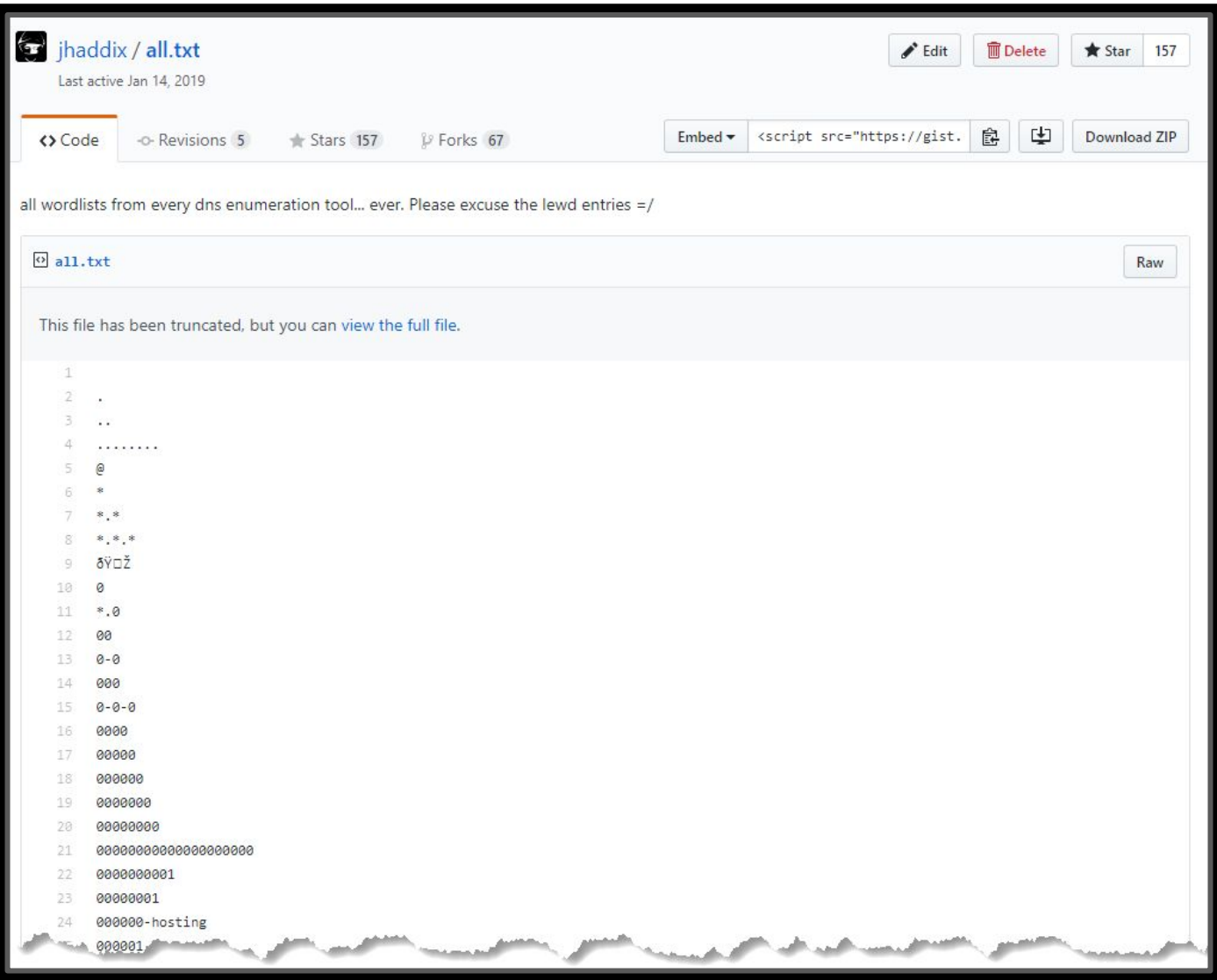
The other option to discover subdomains is bruteforce.

If we try and resolve thistotallydoesntexist.company.com we will *usually* not get a record.

So we can use a large list of common subdomain names and just try and resolve them analyzing if they succeed.

Both Amass and Subfinder have options built in to bruteforce subdomains. The problem in this method is that only using one DNS server to do this will take forever. Some tools have come out that are both threaded and use multiple DNS resolvers simultaneously. This speeds up this process significantly. massdns pioneered this idea. Amass (8 revolvers by default) does this with the -rf flag, allowing you to specify a list of resolvers. Subfinder (8 resolvers by default) with the -rL flag.

Sources for Bruteforce Lists



With multi resolver tools we no longer need to wait weeks to bruteforce a comprehensive list of possible subdomains. I went out and gathered every subdomain tool in existence and combined them. Over a million entries can be run in 2-5min.

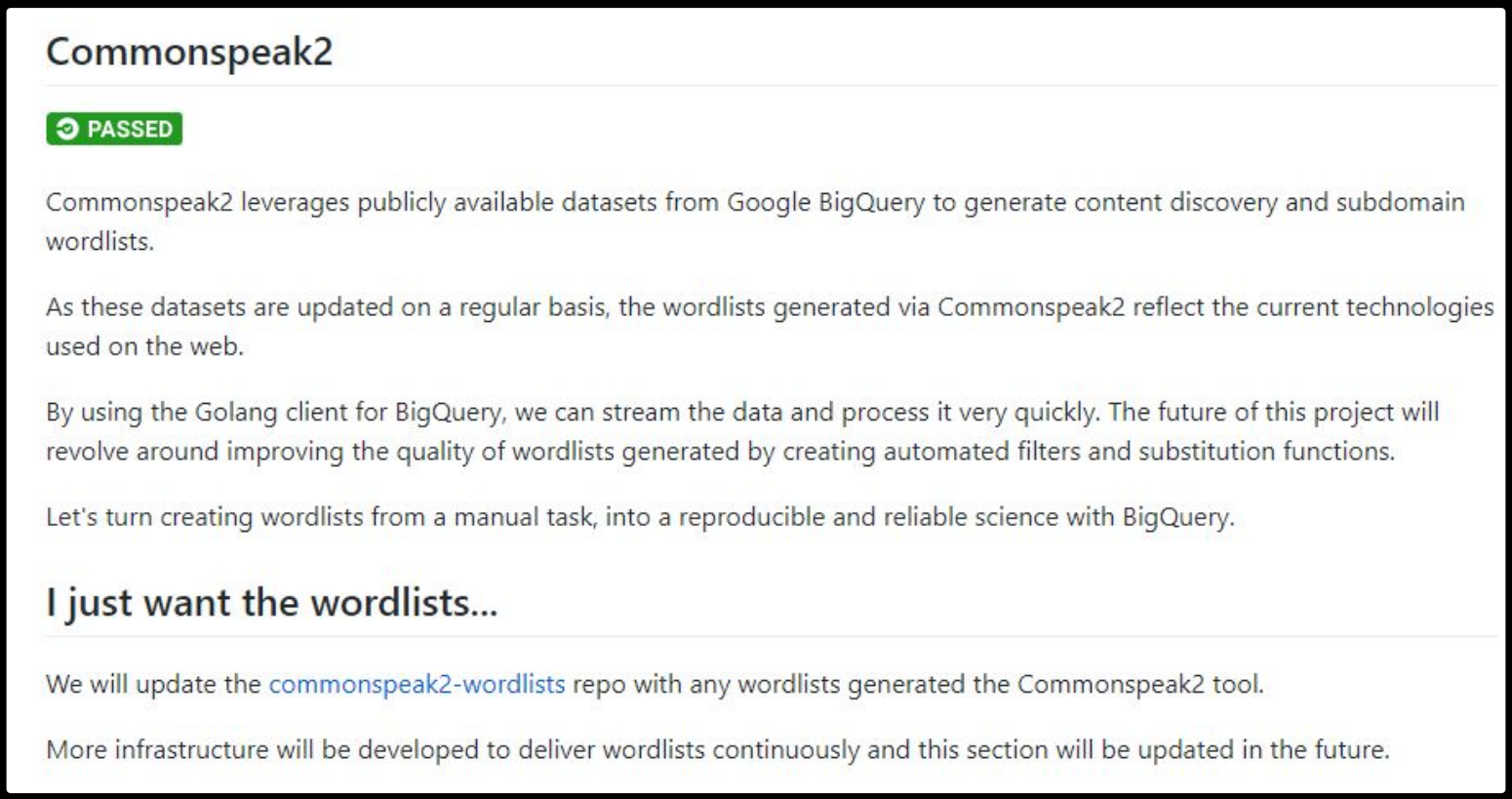
- https://gist.github.com/jhaddix/86a06c5dc309d08580a018c66354a056

But wait... there's more!

- https://gist.github.com/orangetw/c10324f68f200fbdc365ec17fa5c18c7

2,062,248 lines...

Sources for Bruteforce Lists



Commonspeak2

PASSED

Commonspeak2 leverages publicly available datasets from Google BigQuery to generate content discovery and subdomain wordlists.

As these datasets are updated on a regular basis, the wordlists generated via Commonspeak2 reflect the current technologies used on the web.

By using the Golang client for BigQuery, we can stream the data and process it very quickly. The future of this project will revolve around improving the quality of wordlists generated by creating automated filters and substitution functions.

Let's turn creating wordlists from a manual task, into a reproducible and reliable science with BigQuery.

I just want the wordlists...

We will update the commonspeak2-wordlists repo with any wordlists generated the Commonspeak2 tool.

More infrastructure will be developed to deliver wordlists continuously and this section will be updated in the future.

New lists for subdomain bruteforce are relatively the same nowadays, but the 1st team to really iterate on this is the AssetNote team. The all.txt file includes commonspeak v1 data but there is also a second version of commonspeak data out:

- https://github.com/assetnote/commonspeak2

Permutation scanning

dev.company.com

dev1.company.com

dev2.company.com

dev-1.company.com

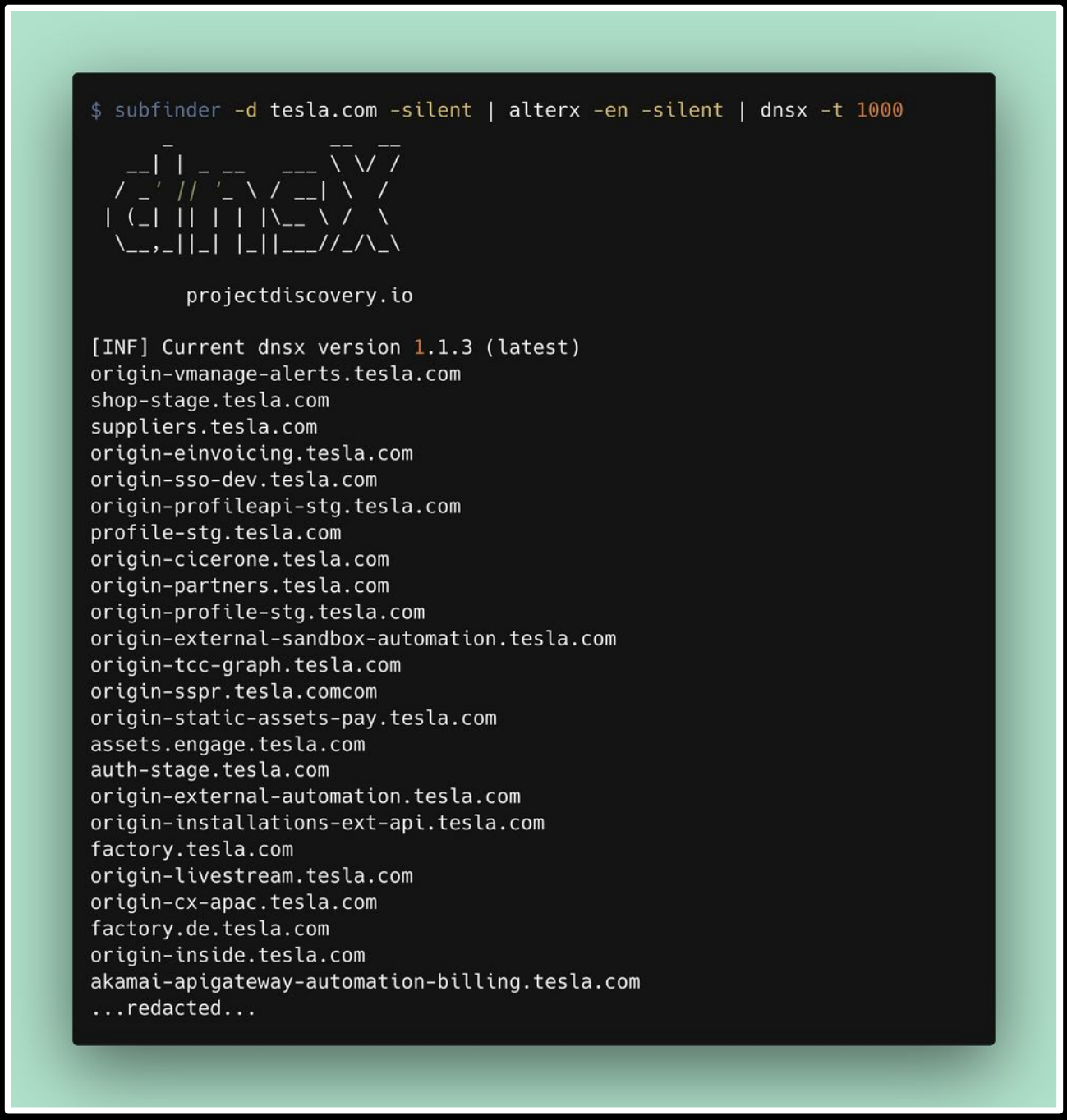
dev-2.company.com

When bruteforcing or gathering subdomains via scraping you may come across a naming pattern in these subdomains. Even though you may not have found it yet, there may be other targets that conform to those naming conventions. In addition, sometimes targets are not explicitly protected across naming conventions. The first tool to attempt to recognize these patterns and bruteforce for some of them was altdns written by Nathaniel Wakelam and Shubs.

Now amass contains logic to check for these “permutations”. Amass includes this analysis in a default run

Some personal experience cited on the next page.

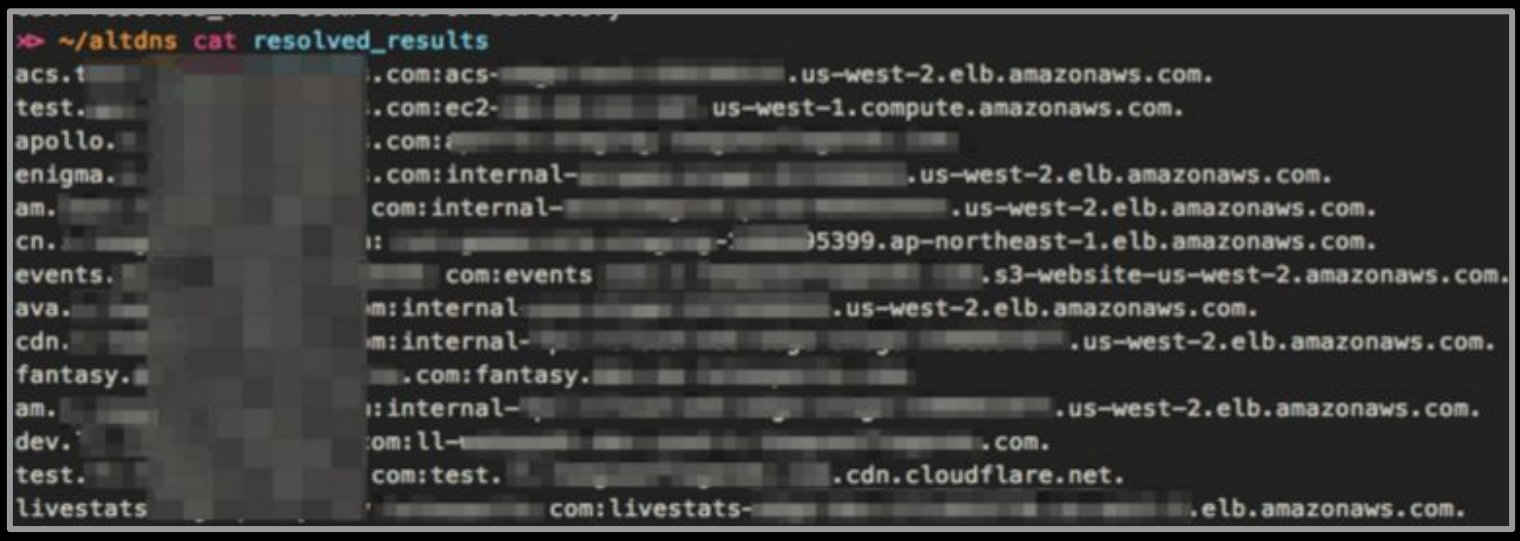
Alterx



```
$ subfinder -d tesla.com -silent | alterx -en -silent | dnsx -t 1000

      _              __  __
   __| | _ __   ___ \ \/ /
  / _` || '_ \ / __| \  /
 | (_| || | | |\__ \ /  \
  \__,_||_| |_||___//_/\_\

        projectdiscovery.io

[INF] Current dnsx version 1.1.3 (latest)
origin-vmanage-alerts.tesla.com
shop-stage.tesla.com
suppliers.tesla.com
origin-einvoicing.tesla.com
origin-sso-dev.tesla.com
origin-profileapi-stg.tesla.com
profile-stg.tesla.com
origin-cicerone.tesla.com
origin-partners.tesla.com
origin-profile-stg.tesla.com
origin-external-sandbox-automation.tesla.com
origin-tcc-graph.tesla.com
origin-sspr.tesla.comcom
origin-static-assets-pay.tesla.com
assets.engage.tesla.com
auth-stage.tesla.com
origin-external-automation.tesla.com
origin-installations-ext-api.tesla.com
factory.tesla.com
origin-livestream.tesla.com
origin-cx-apac.tesla.com
factory.de.tesla.com
origin-inside.tesla.com
akamai-apigateway-automation-billing.tesla.com
...redacted...
```

Alterx is the newest tool for generating both patterned and common permutations.

https://github.com/projectdiscovery/alterx



Jason Haddix
@Jhaddix

Security testing against Akamai? look for origin-sub.domain.com or origin.sub.domain.com , bypass the filtering by going to the source.

12:06 PM - 13 Sep 2017

43 Retweets **95** Likes

2 43 95

Jason Haddix
@Jhaddix

WAF had me down on www.$target.com ='(

too bad they missed ww2.$target.com !

sqli in progress...
#OMGSOMANYTABLESToEXFIL

8:14 PM - 16 Feb 2018

6 Retweets **104** Likes

Other Wide Recon

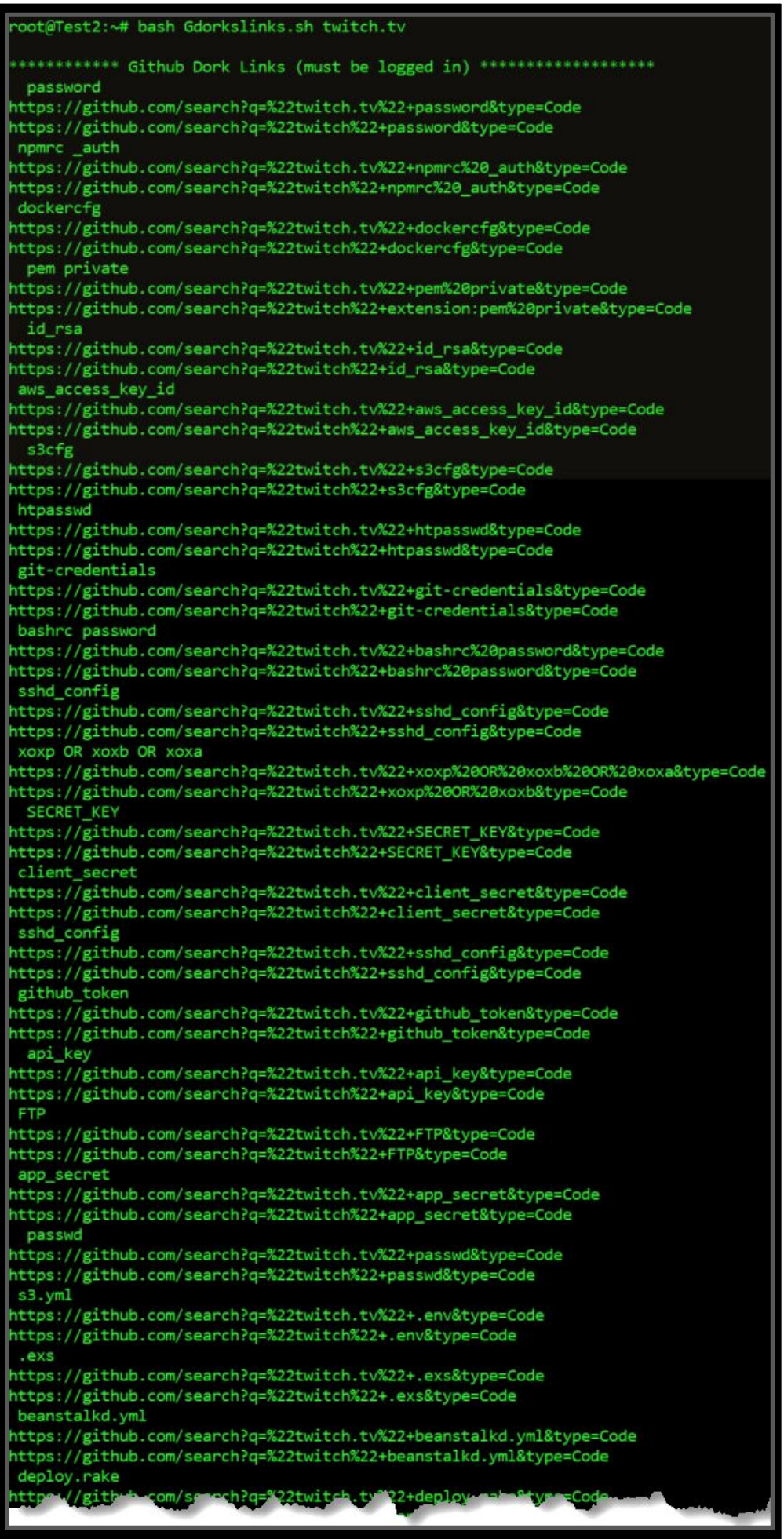
```
[05:06:24] [INFO] loading tamper script 'versionedmorekeywords'
[05:06:24] [WARNING] tamper script 'versionedmorekeywords' is only meant to be run against MySQL >= 5.1.13
[05:06:24] [INFO] loading tamper script 'xforwardedfor'
[05:06:24] [WARNING] using too many tamper scripts is usually not a good idea
custom injection marking character ('*') found in option '--headers/--user-agent/--referer/--cookie'. Do you want to process it? [Y/n/q] Y
[05:06:28] [INFO] testing connection to the target URL
[05:06:29] [WARNING] the web server responded with an HTTP error code (406) which could interfere with the results of the tests
[05:06:29] [INFO] testing if the target URL is stable
[05:06:29] [INFO] target URL is stable
[05:06:29] [INFO] testing if (custom) HEADER parameter 'Cookie #1*' is dynamic
[05:06:29] [WARNING] currently only couple of keywords are being processed ('UNION', 'SELECT', 'INSERT', 'UPDATE', 'FROM', 'WHERE'). You can set it manually according to your needs
[05:06:30] [WARNING] reflective value(s) found and filtering out
[05:06:30] [INFO] confirming that (custom) HEADER parameter 'Cookie #1*' is dynamic
[05:06:31] [INFO] (custom) HEADER parameter 'Cookie #1*' is dynamic
[05:06:31] [WARNING] heuristic (basic) test shows that (custom) HEADER parameter 'Cookie #1*' might not be injectable
[05:06:34] [INFO] testing for SQL injection on (custom) HEADER parameter 'Cookie #1*'
[05:06:34] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[05:06:39] [INFO] (custom) HEADER parameter 'Cookie #1*' seems to be 'AND boolean-based blind - WHERE or HAVING clause' injectable (with --string="\\xa0\\xa0\\xa0\\xa0\n\tat com.ibm.ws.http.channel.inbound.impl.HttpInboundLink.ready(HttpInboundLink.java:287)")
[05:06:48] [INFO] testing 'MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause'
[05:06:48] [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'
[05:06:49] [INFO] testing 'Microsoft SQL Server/Sybase AND error-based - WHERE or HAVING clause'
[05:06:49] [INFO] testing 'Oracle AND error-based - WHERE or HAVING clause (XMLType)'
[05:06:49] [INFO] testing 'Generic inline queries'
[05:06:49] [INFO] testing 'PostgreSQL inline queries'
[05:06:50] [INFO] testing 'Microsoft SQL Server/Sybase inline queries'
[05:06:51] [INFO] testing 'MySQL > 5.0.11 stacked queries (SELECT - comment)'
[05:06:52] [WARNING] time-based comparison requires larger statistical model, please wait. (done)
[05:06:52] [INFO] testing 'PostgreSQL > 8.1 stacked queries (comment)'
[05:06:53] [INFO] testing 'Microsoft SQL Server/Sybase stacked queries (comment)'
[05:06:54] [INFO] testing 'Oracle stacked queries (DBMS_PIPE.RECEIVE_MESSAGE - comment)'
[05:06:55] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (SELECT)'
[05:06:56] [INFO] testing 'PostgreSQL > 8.1 AND time-based blind'
[05:06:56] [INFO] testing 'Microsoft SQL Server/Sybase time-based blind'
[05:06:57] [INFO] testing 'Oracle AND time-based blind'
[05:06:58] [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns'
[05:06:58] [INFO] automatically extending ranges for UNION query injection technique tests as there is at least one other (potential) technique found
[05:07:12] [INFO] testing 'MySQL UNION query (NULL) - 1 to 20 columns'
[05:07:27] [INFO] checking if the injection point on (custom) HEADER parameter 'Cookie #1*' is a false positive
[05:07:29] [WARNING] false positive or unexploitable injection point detected
[05:07:29] [WARNING] (custom) HEADER parameter 'Cookie #1*' is not injectable
[05:07:29] [WARNING] HTTP error codes detected during run:
400 (Bad Request) - 10 times, 500 (Internal Server Error) - 72 times, 406 (Not Acceptable) - 3 times
        _
___ ___| |_____ ___ ___  {1.0.4.0#dev}
|_ -| . | |     | .'| . |
|___|_  |_|_|_|_|__,|  _|
      |_|           |_|   http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting at 05:08:44
```

Github Recon

Many organizations quickly grow in their engineering teams. Sooner or later a new developer, intern, contractor, or other staff will leak source code online, usually through a public Github repo that they mistakenly thought they had set private.

Enjoy my github dork collection... They win... a lot.

*** Helps if your console supports clickable hyperlinks*



https://gist.github.com/jhaddix/1fb7ab2409ab579178d2a79959909b33

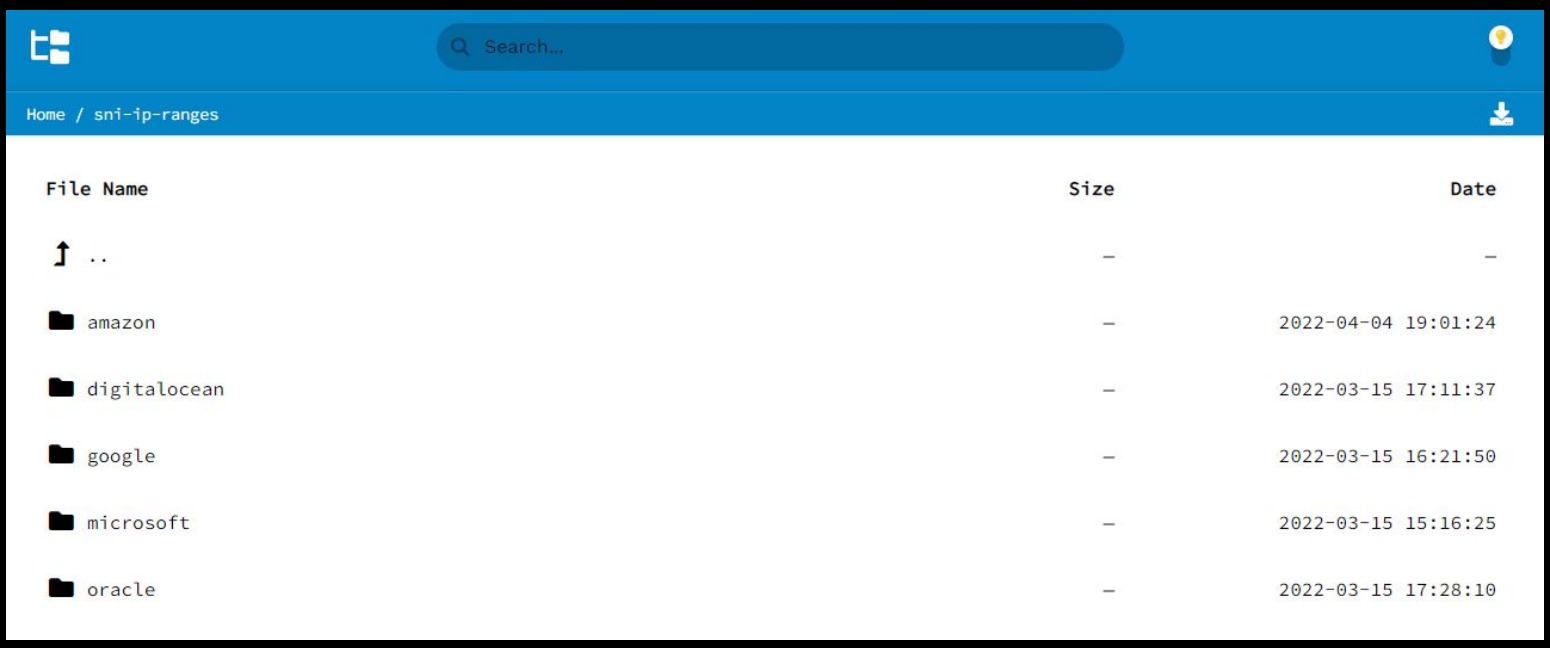
Github Recon cont.

The last list and script was designed to find sensitive source code, most often exposure of sensitive credentials (very common). You can also learn a lot about:

- New hosts and TLDs
  - Be sure to keep a lookout for cloud storage (AWS and Azure)
    - amazonaws.com
- Host naming patterns
  - bender.company.com, fry.company.com, zoidberg.company.com, ...
- Technology stacks
  - DBMS type for injection
  - OSS components for CVE’s

More on Github in the Github sections!

Cloud Recon



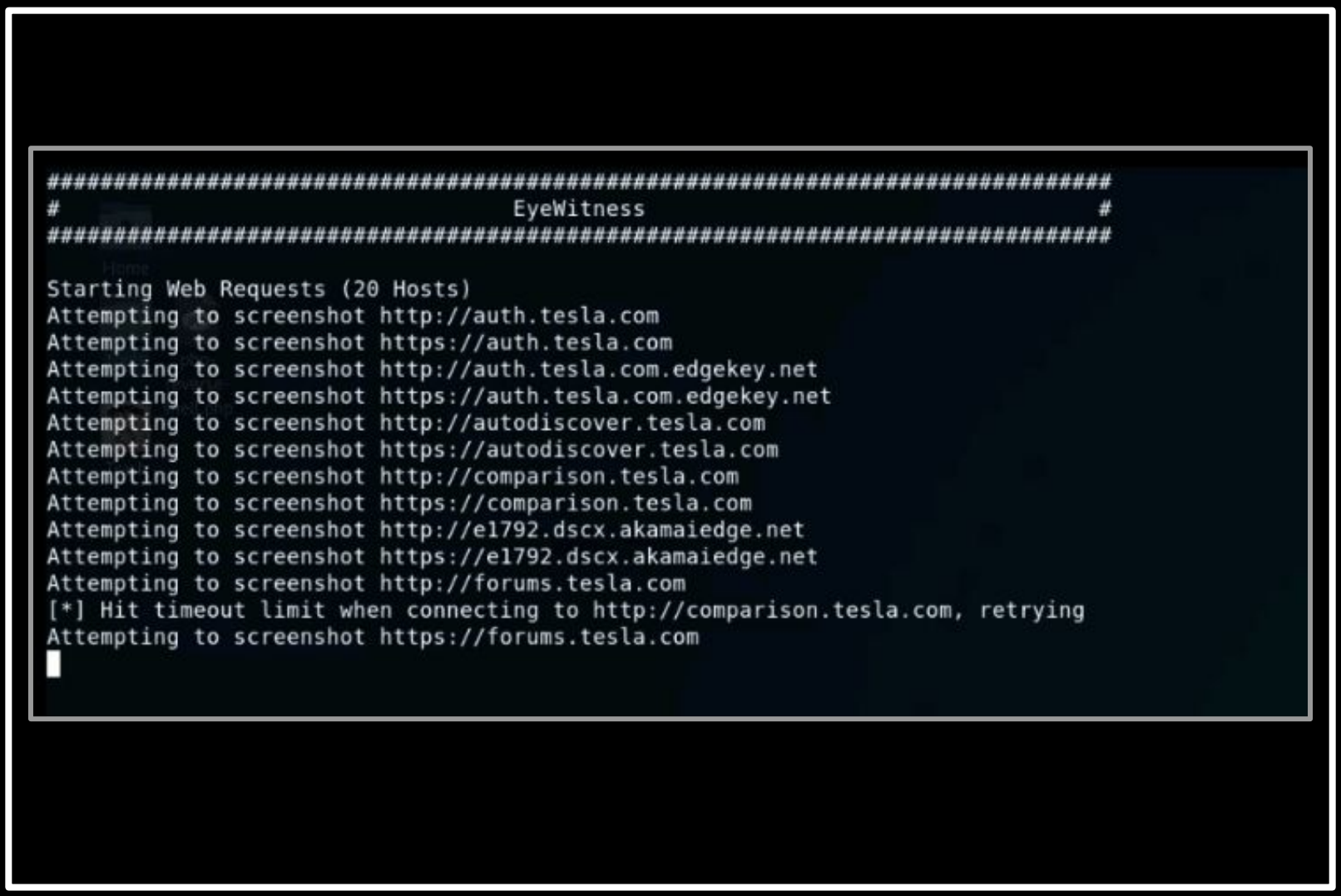
Another way we can glean domains or subdomains is from scanning cloud IP ranges. We scan the ranges for port 443 and attempt to parse the certificate data. A project exists for this:

- http://kaeferjaeger.gay/?dir=sni-ip-ranges

We can then query these files on the command line:

Cat *.txt | grep {TARGET}

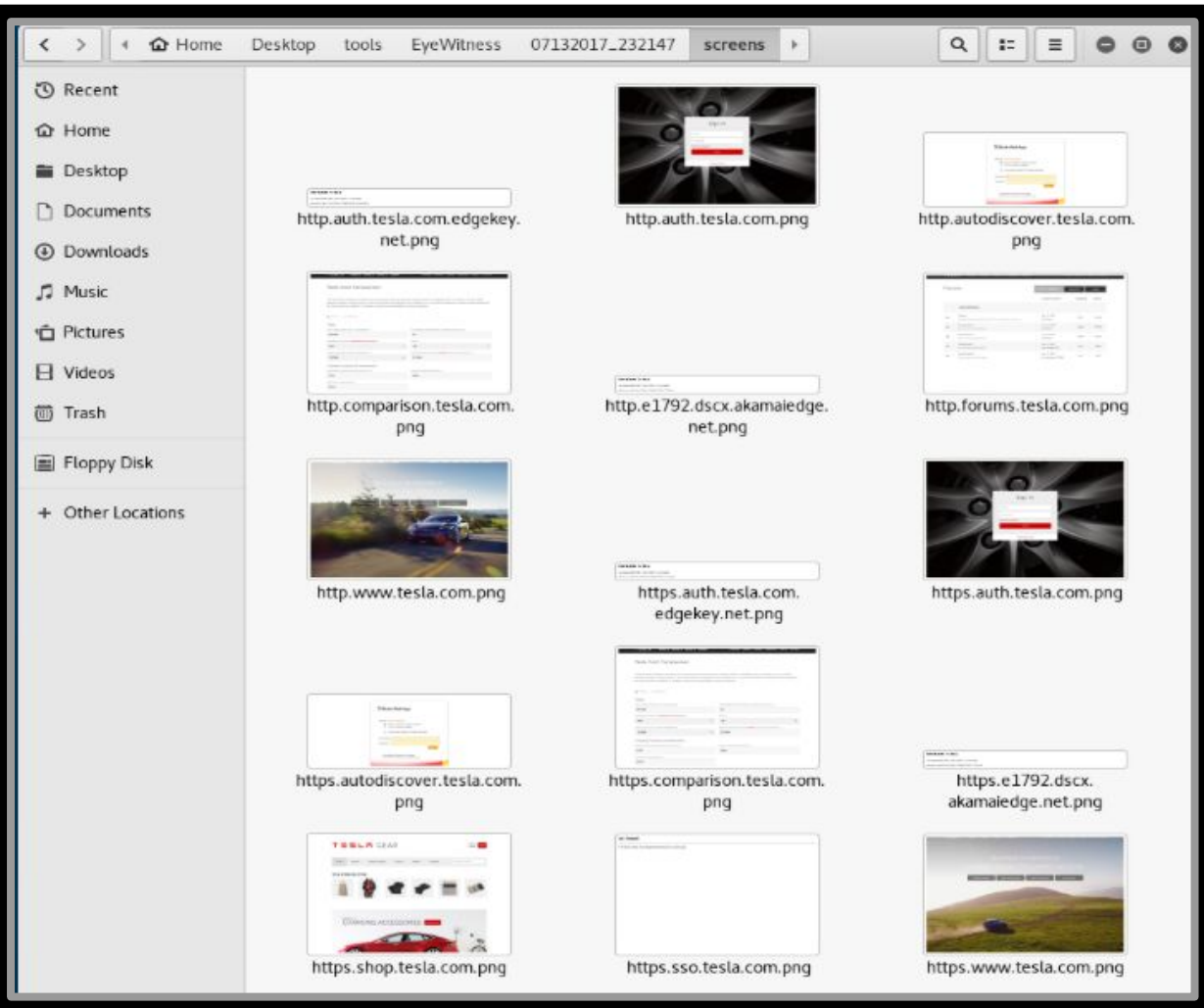
Screen-shotting for prioritization



At this point we have a lot of attack surface. We can feed possible sites to a tool and attempt to screenshot the results. This will allow us to “eye-ball” things that might be interesting.

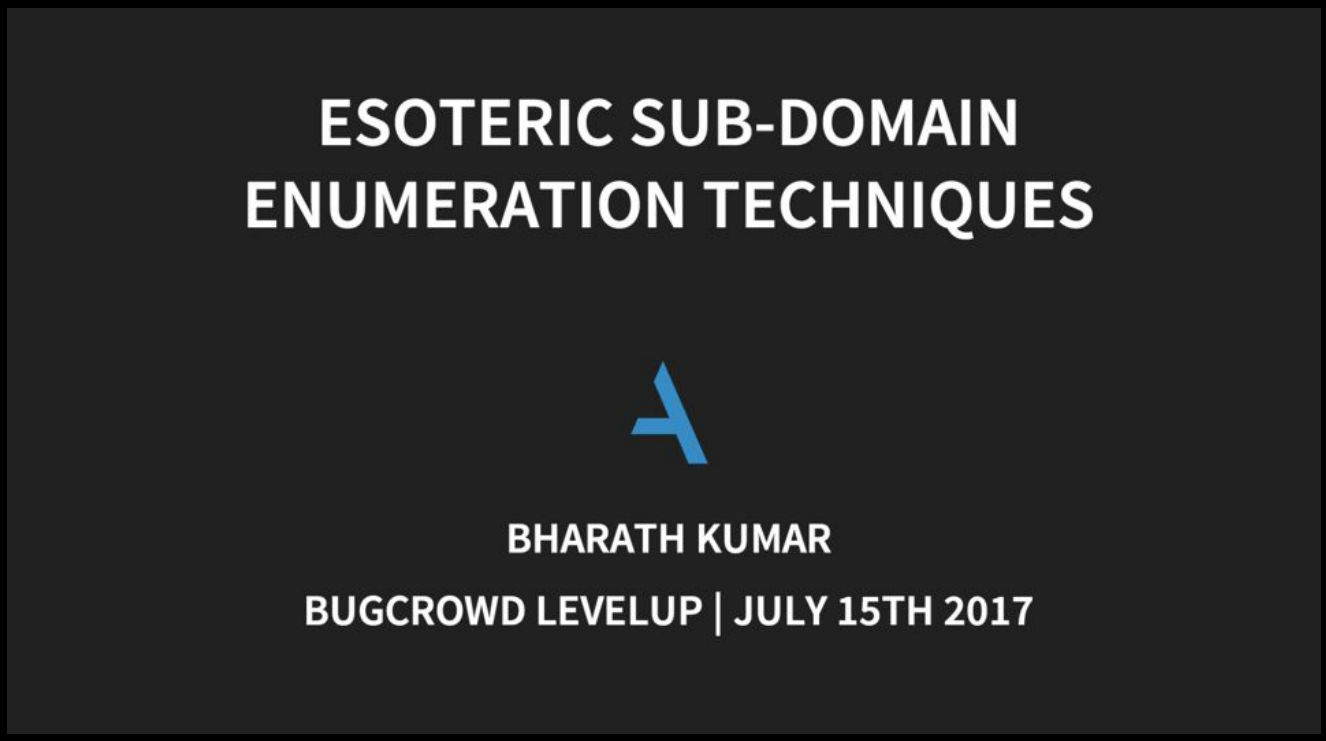
There are many tools for this. Aquatone is a wider recon framework that does this, HTTPscreenshot, and Eyewitness. I use Eyewitness because it will prepend both the http and https protocol for each domain we have observed. I’m not highly tied to this tool though, find one that works for you.

Screen-shotting for prioritization



*** talk about aquatone and recon-ng merits and advancements

DNSSEC Scanning (new zone transfer)

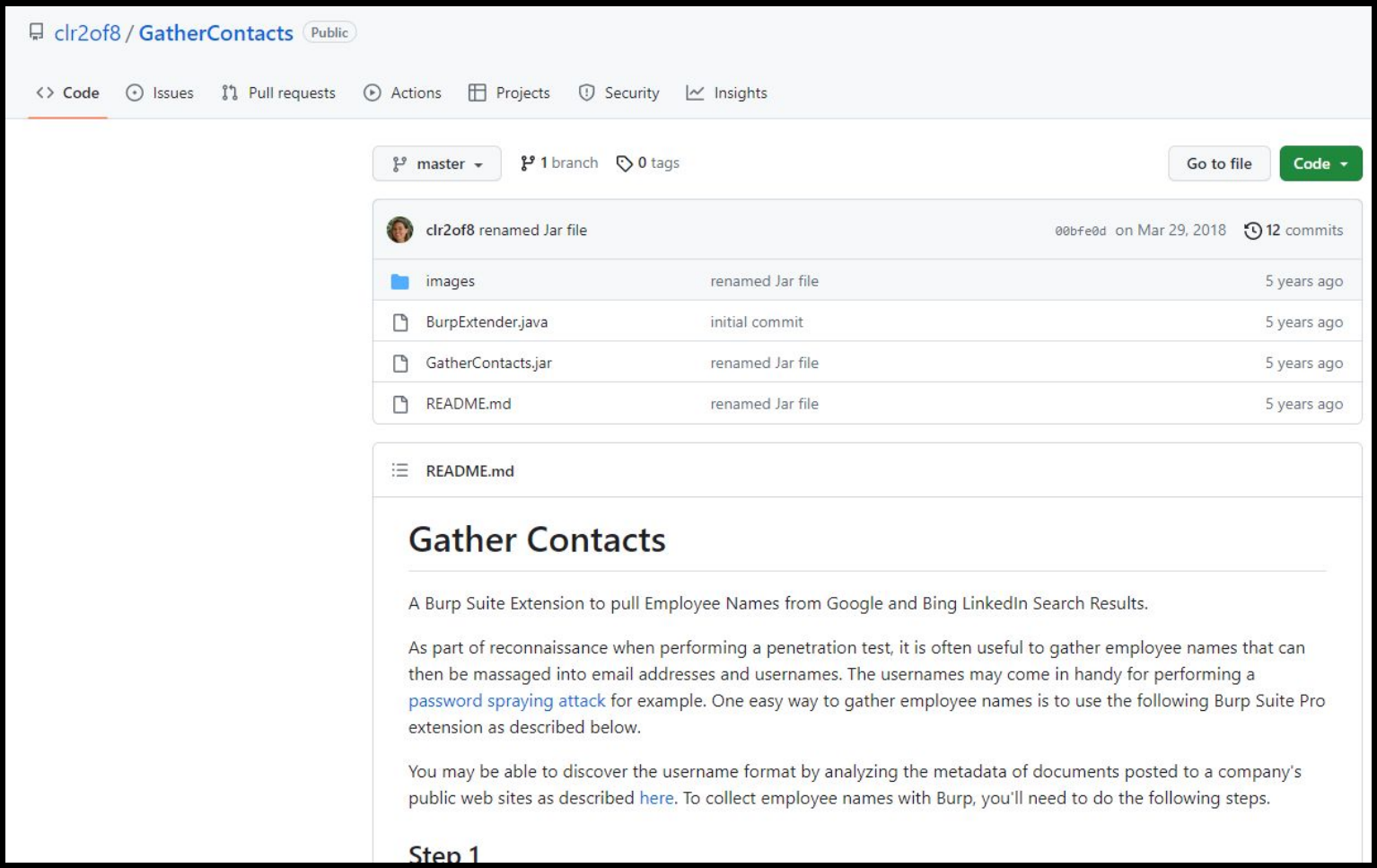


There also exists some methods to do DNSSEC “Walking” , emulating a zone transfer. You can find more research on this at the below URLs:

- https://github.com/appsecco/bugcrowd-levelup-subdomain-enumeration/blob/master/esoteric_subdomain_enumeration_techniques.pdf

- https://www.youtube.com/watch?v=1Kg0_53ZEg8

Email Enum

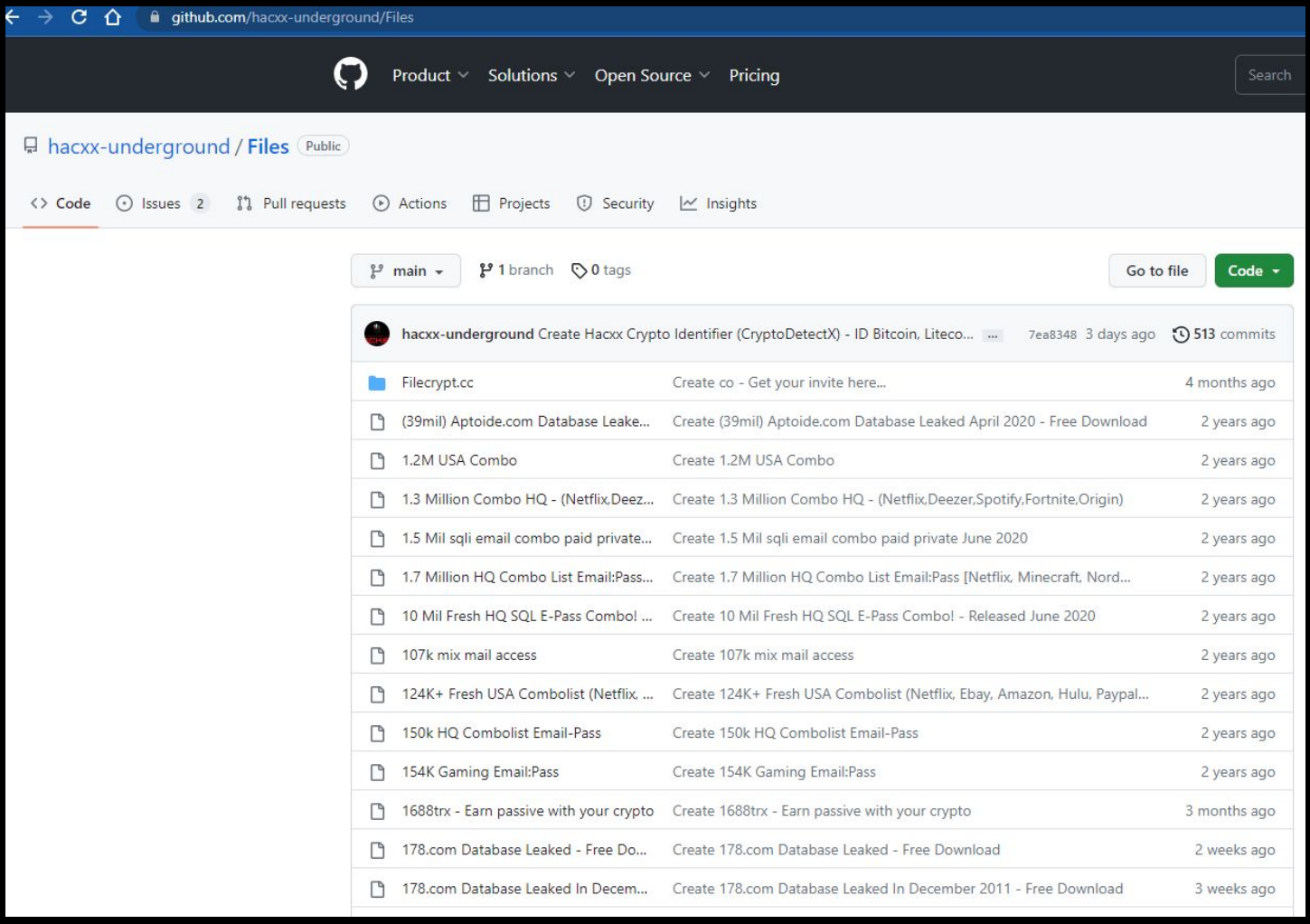


One of the biggest asks we have for red teaming is to enumerate email addresses. We can do this a few ways.

1) Hunter.io
2) Dehashed
3) Then Gather Contacts

https://github.com/clr2of8/GatherContacts

Build your own!



Download torrents and build your own TI mass data:

https://github.com/hacxx-underground/Files

Wayback / Archive.org

We can gather more website endpoints using Waymore, it gives us:

- Wayback Machine (web.archive.org)
- Common Crawl (index.commoncrawl.org)
- Alien Vault OTX (otx.alienvault.com)
- URLScan (urlscan.io)

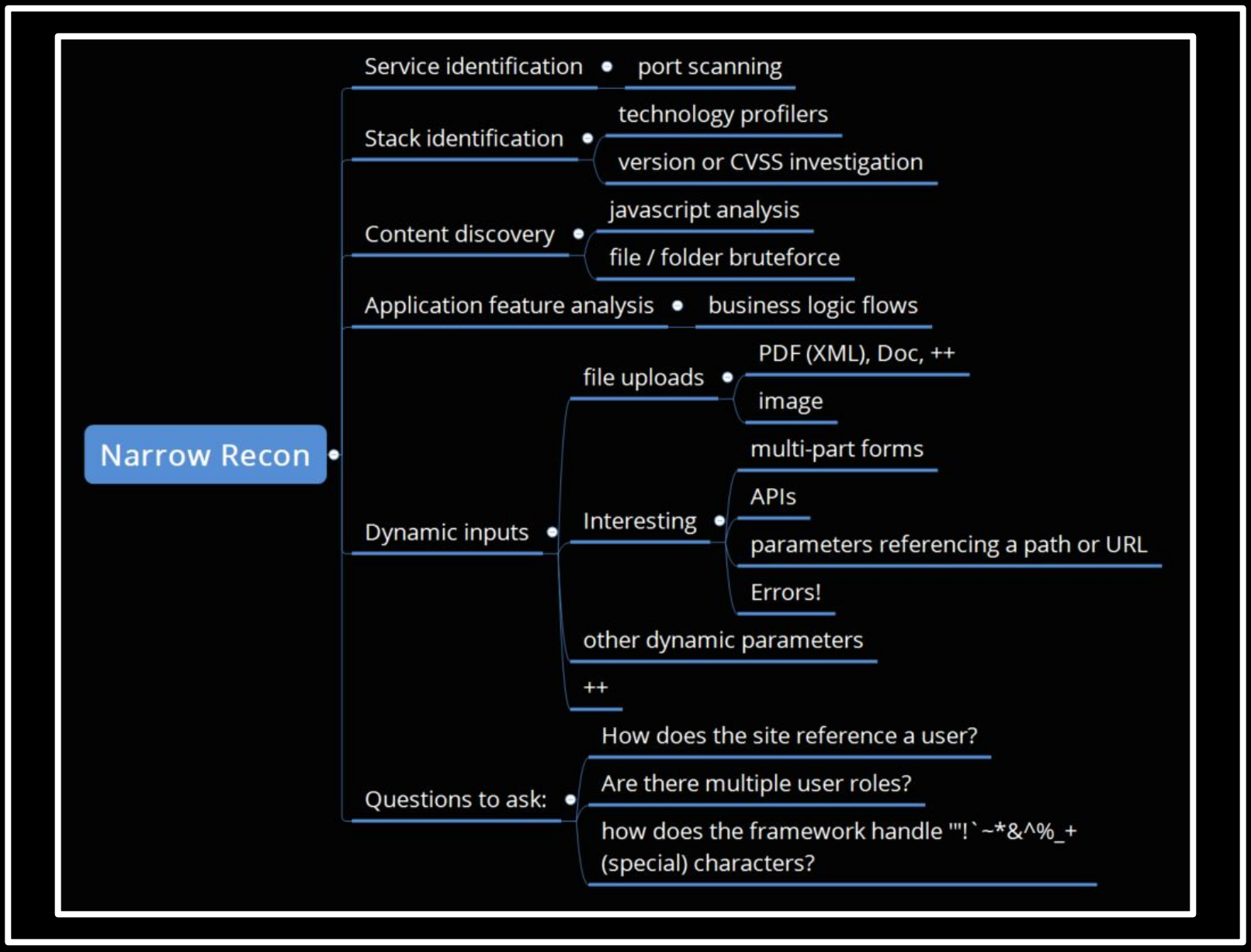
Some other tools for wayback enumeration:

1. WaybackURLs: https://gist.github.com/mhmdiaa/adf6bff70142e5091792841d4b372050
2. WaybackRobots: https://gist.github.com/mhmdiaa/2742c5e147d49a804b408bfed3d32d07
3. https://github.com/tomnomnom/waybackurls
4. https://github.com/daudmalik06/ReconCat

```
[05:06:24] [WARNING] tamper script 'versionedkeywords' is only meant to be run against MySQL
[05:06:24] [INFO] loading tamper script 'versionedmorekeywords'
[05:06:24] [WARNING] tamper script 'versionedmorekeywords' is only meant to be run against MySQL >=
5.1.13
[05:06:24] [INFO] loading tamper script 'xforwardedfor'
[05:06:24] [WARNING] using too many tamper scripts is usually not a good idea
custom injection marking character ('*') found in option '--headers/--user-agent/--referer/--cookie'
. Do you want to process it? [Y/n/q] Y
[05:06:28] [INFO] testing connection to the target URL
[05:06:29] [WARNING] the web server responded with an HTTP error code (406) which could interfere wi
th the results of the tests
[05:06:29] [INFO] testing if the target URL is stable
[05:06:29] [INFO] target URL is stable
[05:06:29] [INFO] testing if (custom) HEADER parameter 'Cookie #1*' is dynamic
[05:06:29] [WARNING] currently only couple of keywords are being processed ('UNION', 'SELECT', 'INSE
RT', 'UPDATE', 'FROM', 'WHERE'). You can set it manually according to your needs
[05:06:30] [WARNING] reflective value(s) found and filtering out
[05:06:30] [INFO] confirming that (custom) HEADER parameter 'Cookie #1*' is dynamic
[05:06:31] [INFO] (custom) HEADER parameter 'Cookie #1*' is dynamic
[05:06:31] [WARNING] heuristic (basic) test shows that (custom) HEADER parameter 'Cookie #1*' might
not be injectable
[05:06:34] [INFO] testing for SQL injection on (custom) HEADER parameter 'Cookie #1*'
[05:06:34] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[05:06:39] [INFO] (custom) HEADER parameter 'Cookie #1*' seems to be 'AND boolean-based blind - WHER
E or HAVING clause' injectable (with --string="\\xa0\\xa0\\xa0\\xa0\n\tat com.ibm.ws.http.channel.in
bound.impl.HttpInboundLink.ready(HttpInboundLink.java:287)")
[05:06:48] [INFO] testing 'MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause
'
[05:06:48] [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'
[05:06:49] [INFO] testing 'Microsoft SQL Server/Sybase AND error-based - WHERE or HAVING clause'
[05:06:49] [INFO] testing 'Oracle AND error-based - WHERE or HAVING clause (XMLType)'
[05:06:49] [INFO] testing 'MySQL >= 5.0 error-based - Parameter replace'
[05:06:49] [INFO] testing 'MySQL inline queries'
[05:06:50] [INFO] testing 'PostgreSQL inline queries'
[05:06:51] [INFO] testing 'Microsoft SQL Server/Sybase inline queries'
[05:06:52] [INFO] testing 'MySQL > 5.0.11 stacked queries (SELECT - comment)'
[05:06:52] [WARNING] time-based comparison requires larger statistical model, please wait. (done)
[05:06:53] [INFO] testing 'PostgreSQL > 8.1 stacked queries (comment)'
[05:06:53] [INFO] testing 'Microsoft SQL Server/Sybase stacked queries (comment)'
[05:06:54] [INFO] testing 'Oracle stacked queries (DBMS_PIPE.RECEIVE_MESSAGE - comment)'
[05:06:55] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (SELECT)'
[05:06:56] [INFO] testing 'PostgreSQL > 8.1 AND time-based blind'
[05:06:56] [INFO] testing 'Microsoft SQL Server/Sybase time-based blind'
[05:06:57] [INFO] testing 'Oracle AND time-based blind'
[05:06:58] [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns'
[05:06:58] [INFO] automatically extending ranges for UNION query injection technique tests as there
is at least one other (potential) technique found
[05:07:12] [INFO] testing 'MySQL UNION query (NULL) - 1 to 20 columns'
[05:07:27] [INFO] checking if the injection point on (custom) HEADER parameter 'Cookie #1*' is a fal
se positive
[05:07:29] [WARNING] false positive or unexploitable injection point detected
[05:07:29] [WARNING] (custom) HEADER parameter 'Cookie #1*' is not injectable
[05:07:29] [WARNING] HTTP error codes detected during run:
400 (Bad Request) - 10 times, 500 (Internal Server Error) - 72 times, 406 (Not Acceptable) - 3 times
        _
 ___ ___| |_____ ___ ___  {1.0.4.0#dev}
|_ -| . | |     | .'| . |
|___|_  |_|_|_|_|__,|  _|
      |_|           |_|   http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal.
 It is the end user's responsibility to obey all applicable local, state and federal laws. Developer
s assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting at 05:08:44
```

Narrow Recon

Narrow Recon



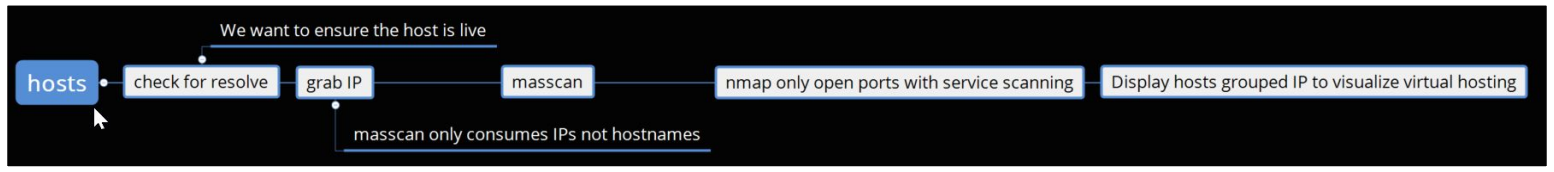
Now that we have a plethora of hosts to investigate (or just one) we need a prioritized methodology to investigate the application. This whole section is predicated on spending a lot of time using the app as a USER.

Each host should have at least a day dedicated to it in wide-scope projects. In narrow scope projects I can spend a week on just getting comfortable in the application.

Let’s get started.

Service Scanning IP Space

```
root@Test2:~# host twitch.tv
twitch.tv has address 151.101.194.167
twitch.tv has address 151.101.2.167
twitch.tv has address 151.101.130.167
twitch.tv has address 151.101.66.167
twitch.tv mail is handled by 30 alt2.aspmx.l.google.com.
twitch.tv mail is handled by 50 aspmx3.googlemail.com.
twitch.tv mail is handled by 10 aspmx.l.google.com.
twitch.tv mail is handled by 40 aspmx2.googlemail.com.
twitch.tv mail is handled by 20 alt1.aspmx.l.google.com.
root@Test2:~# masscan -p1-65535 151.101.194.167 --max-rate 1800

Starting masscan 1.0.3 (http://bit.ly/14GZzcT) at 2019-01-18 06:34:05 GMT
 -- forced options: -sS -Pn -n --randomize-hosts -v --send-eth
Initiating SYN Stealth Scan
Scanning 1 hosts [65535 ports/host]
Discovered open port 80/tcp on 151.101.194.167
Discovered open port 443/tcp on 151.101.194.167
```



We need to identify open services on all the boxes now. This allows us to verify an application is present, and sometimes “hidden” on obscure ports. Hacker education would have you use nmap here, but masscan by Robert Graham is much faster for general “finding-open-ports-on-TCP”. Chaining masscan’s output to then be nmap’ed can save a lot of time.

Masscan achieves this speed with a re-written TCP/IP stack, true multi-threading, and is written in C.

Sample syntax for scanning a list of IPs:

- masscan -p1-65535 -iL $hostFile --max-rate 1800 -oG $outPutFile.log

A full syntax guide of masscan (authored by Daniel Miessler) can be found here: https://danielmiessler.com/study/masscan/

Service Scanning IP Space

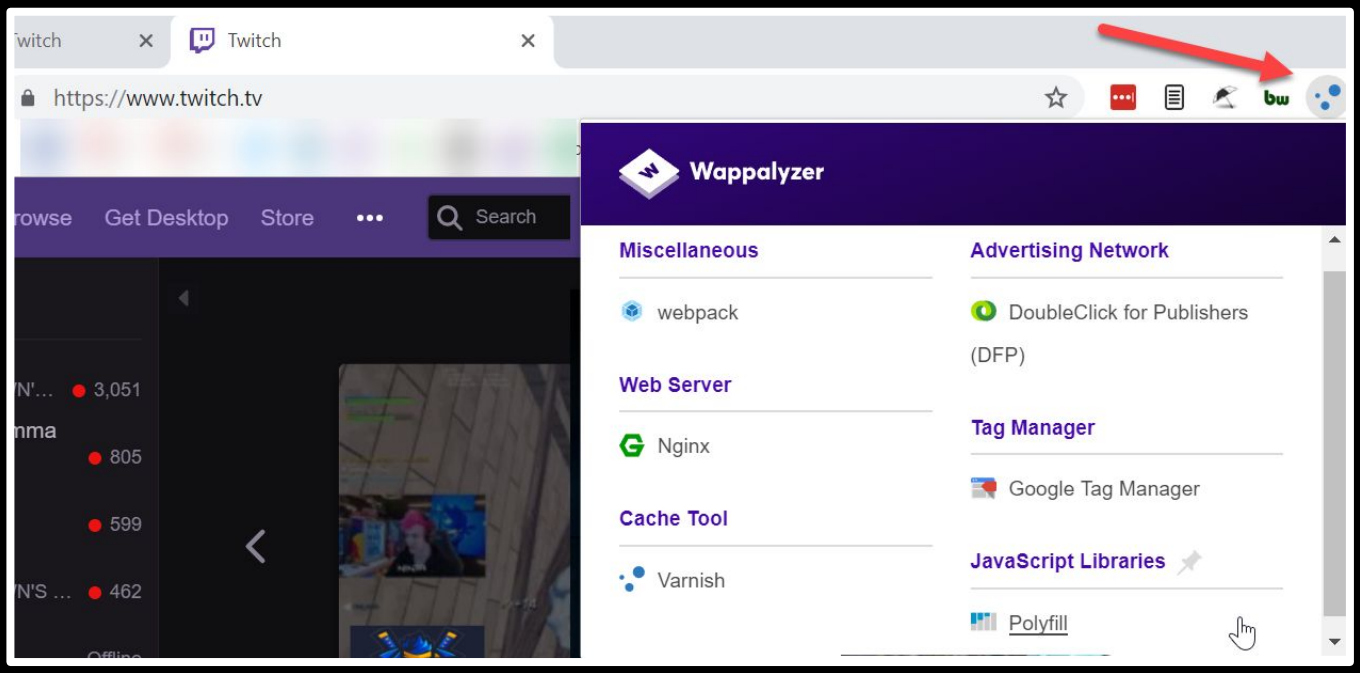
```
#!/bin/bash
strip=$(echo $1|sed 's/https\?:\/\///')
echo ""
echo "################################################"
host $strip
echo "################################################"
echo ""
masscan -p1-65535 $(dig +short $strip|grep -oE
"\b([0-9]{1,3}\.){3}[0-9]{1,3}\b"|head -1) --max-rate 1000 |& tee $strip_scan
```

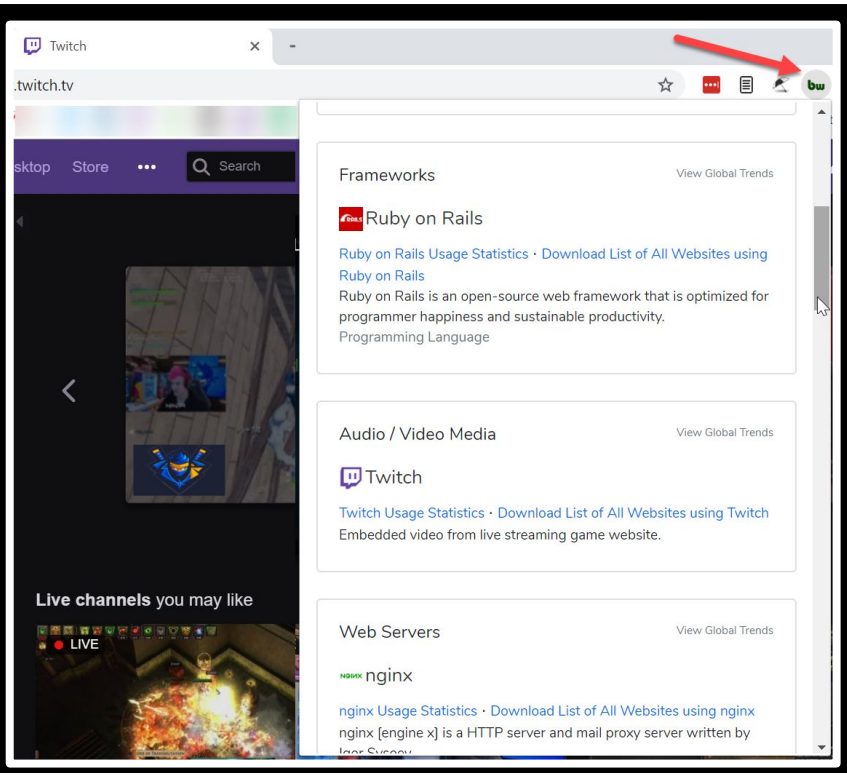
```
root@Test2:~# bash massAlink.sh https://twitch.tv

################################################
twitch.tv has address 151.101.194.167
twitch.tv has address 151.101.2.167
twitch.tv has address 151.101.130.167
twitch.tv has address 151.101.66.167
twitch.tv mail is handled by 40 aspmx2.googlemail.com.
twitch.tv mail is handled by 30 alt2.aspmx.l.google.com.
twitch.tv mail is handled by 20 alt1.aspmx.l.google.com.
twitch.tv mail is handled by 50 aspmx3.googlemail.com.
twitch.tv mail is handled by 10 aspmx.l.google.com.
################################################


Starting masscan 1.0.3 (http://bit.ly/14GZzcT) at 2019-01-18 07:35:03 GMT
 -- forced options: -sS -Pn -n --randomize-hosts -v --send-eth
Initiating SYN Stealth Scan
Scanning 1 hosts [65535 ports/host]
Discovered open port 443/tcp on 151.101.194.167
Discovered open port 80/tcp on 151.101.194.167
```

This script will:

1. Take a http or https link and strip the protocol
2. Resolve the domain to IP and pull the 1st server listed
3. Masscan all ports on that IP
4. Create a log file of the scan

Stack Analysis



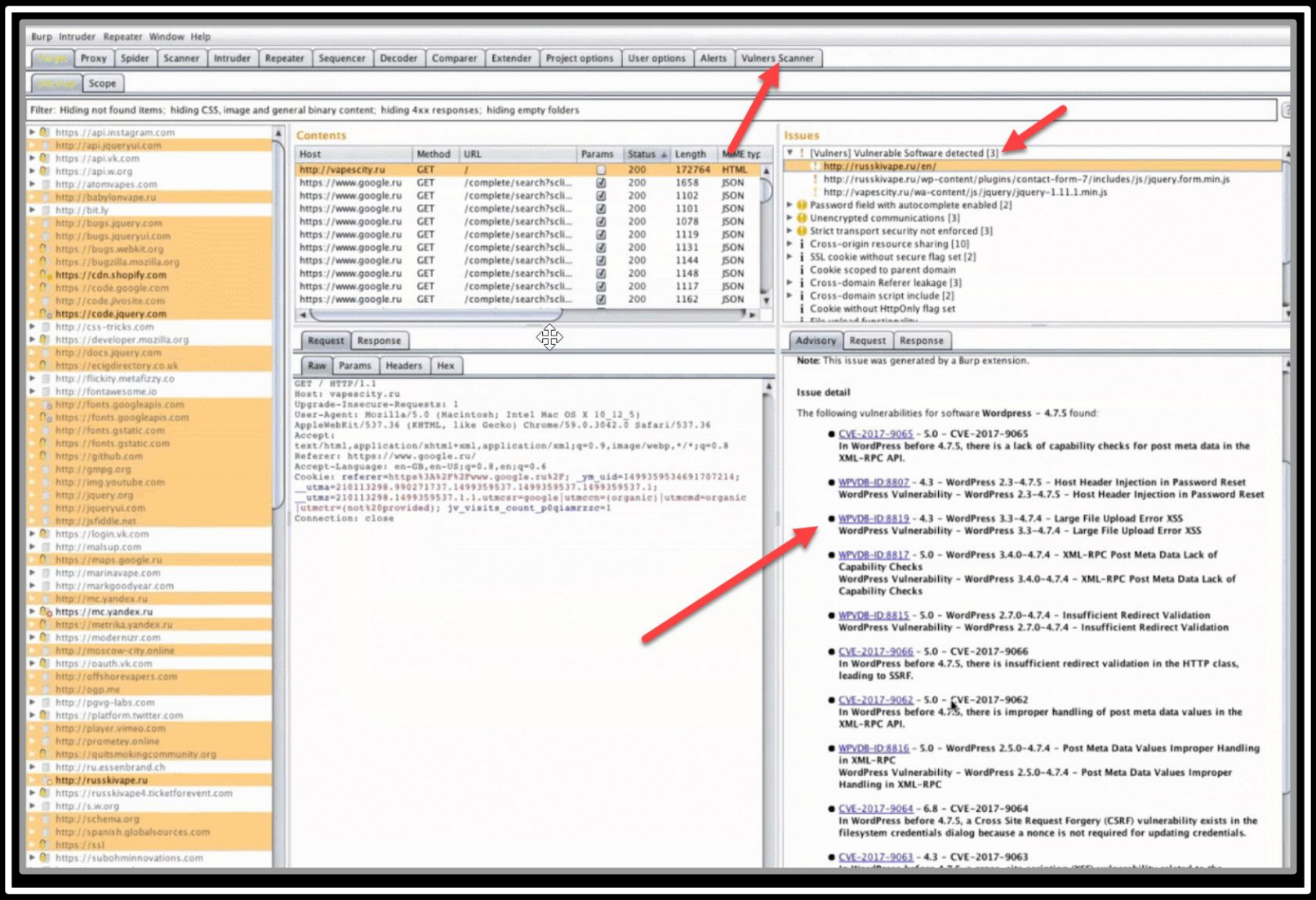


Now that we know all the targets and services running on those targets we have to identify what they are using server-side.

We can use a few projects to do this. My favorites (due to accuracy and usability) are BuiltWith and Wappalyzer. Both have browser extensions to check a site you are currently visiting in the browser.

Wappalyzer has a command line version.

Version based vulnerability analysis



When we have this stack information, it sometimes comes with version information as well (usually from headers, banners, comments, and default files).

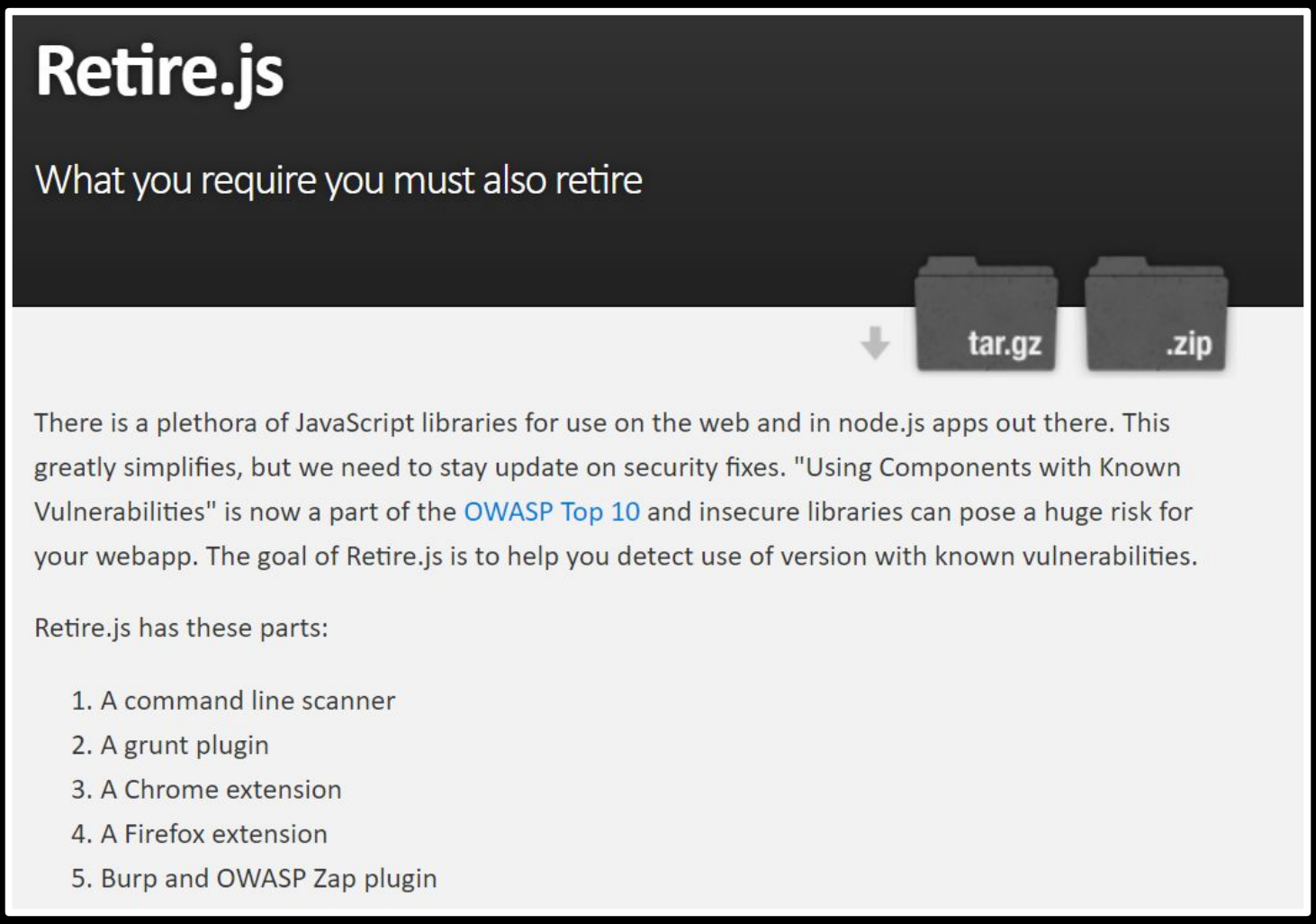
There are very standard vulnerability assessment tools that look at this version info and see if the software is out of date, and alert us to any security flaws that have come out since that version. Most notably Nessus.

In the context of web applications (without buying a commercial scanner to do this) I prefer to use Nikto, Arachni, and Vulners Burp extension. Complete usage-guides of these tools are out of the scope of this training but they all offer different strengths.

In this slide i focus on the one that has no “active” scanning components (vulners) and only does version analysis and inference.

- We can also chat about active scanners now... I’ll give you my opinions =P
- https://github.com/We5ter/Scanners-Box

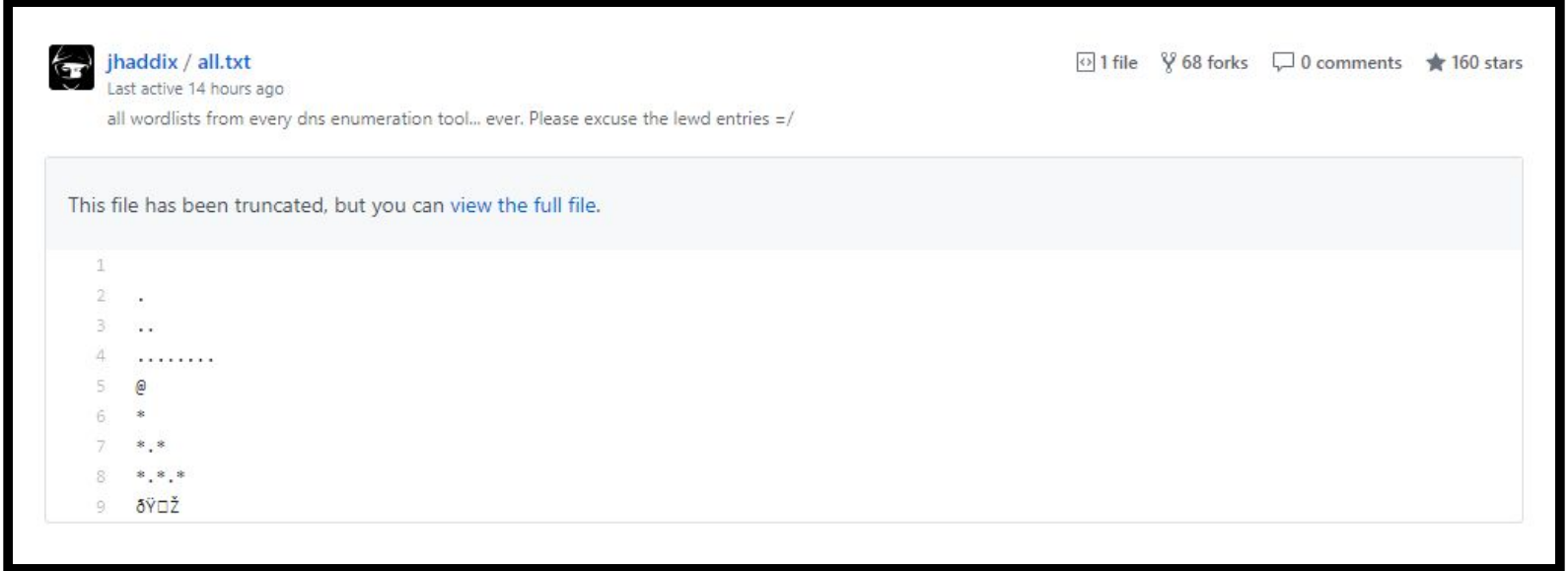
Version based vulnerability analysis



The vulners extension covers most COTS software and web servers, but there are also many more tools to do this for very specific portions of the server software:

- WPScan: assesses Wordpress and it's plugin ecosys
- Retire.js: assesses JavaScript libraries
- CMSmap and CMSScan : WordPress, Joomla, Drupal, vBulletin, and Moodle scanners

Content discovery



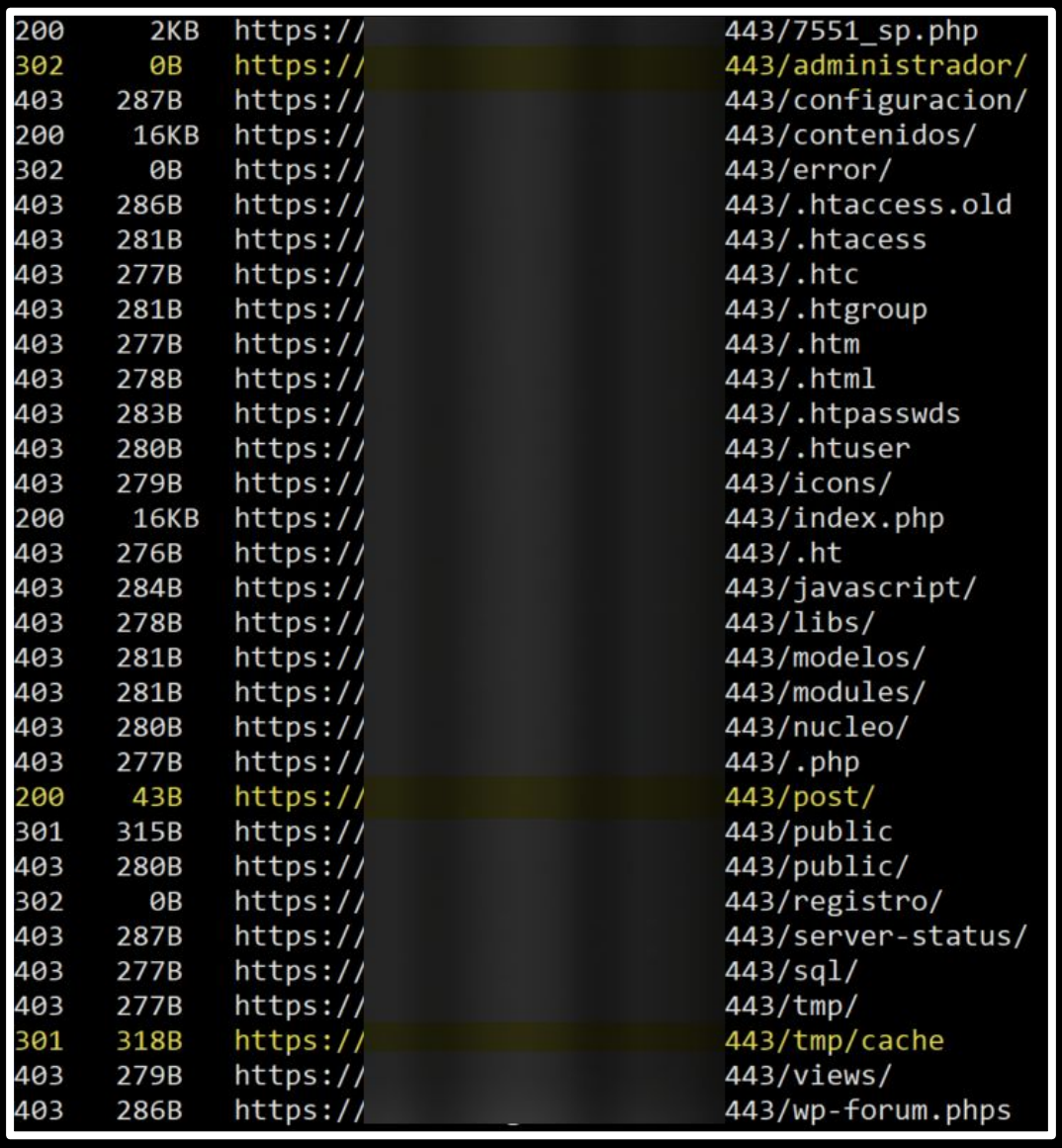
Content discovery, also known as file/folder bruteforce, “Dirbusting”, etc, is an integral part of any bug hunter’s flow.

My preferred two tools in this area are Gobuster and Dirsearch (there are many though).

I support these tools by using a directory/file list that i made from all tools i’ve ever seen on this topic. My all.txt file.

I also use lists from Seclists, commonspeak2, and robotsdisallowed.

Content discovery



```
200     2KB   https://                    443/7551_sp.php
302     0B    https://                    443/administrador/
403   287B    https://                    443/configuracion/
200    16KB   https://                    443/contenidos/
302     0B    https://                    443/error/
403   286B    https://                    443/.htaccess.old
403   281B    https://                    443/.htacess
403   277B    https://                    443/.htc
403   281B    https://                    443/.htgroup
403   277B    https://                    443/.htm
403   278B    https://                    443/.html
403   283B    https://                    443/.htpasswds
403   280B    https://                    443/.htuser
403   279B    https://                    443/icons/
200    16KB   https://                    443/index.php
403   276B    https://                    443/.ht
403   284B    https://                    443/javascript/
403   278B    https://                    443/libs/
403   281B    https://                    443/modelos/
403   281B    https://                    443/modules/
403   280B    https://                    443/nucleo/
403   277B    https://                    443/.php
200    43B    https://                    443/post/
301   315B    https://                    443/public
403   280B    https://                    443/public/
302     0B    https://                    443/registro/
403   287B    https://                    443/server-status/
403   277B    https://                    443/sql/
403   277B    https://                    443/tmp/
301   318B    https://                    443/tmp/cache
403   279B    https://                    443/views/
403   286B    https://                    443/wp-forum.phps
```

Some notes on content discovery:

- While lists may be long and traffic high, directory bruteforce leads to more exploits than any other technique. Either aiding in discovery of new places to exploit with a technical flaw, or due to information disclosure or authorization bypass.
- 200/301 responses are 1st priority if the words look sensitive
- Recursively brute forcing is a must on 403 responses.
- When using command line tools filter by response codes or return bytes.