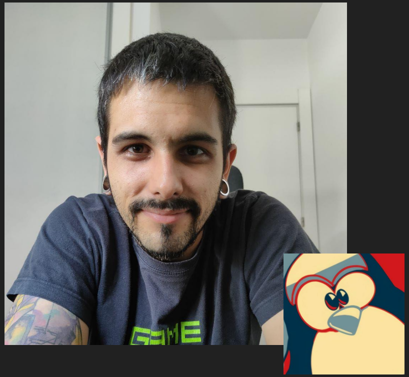
Recon tips & tricks

Improving your hunting for fun

```
> whoami

    six2dez

> neofetch
```



```
Name Alexis Fernández
Location Madrid, Spain
Position Visma Red Team
Others Cobalt Core & Synack Red Team
Website six2dez.com
Projects reconFTW & OneListForAll
GitHub/Twitter six2dez/@six2dez1
```

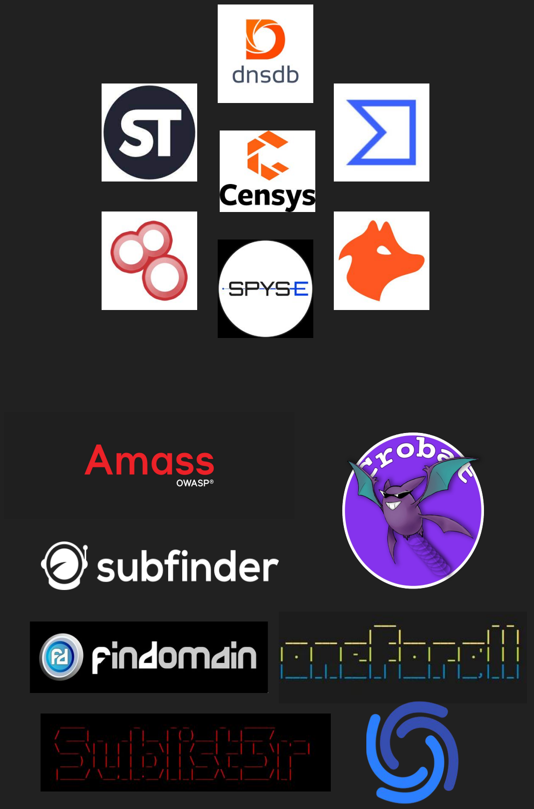
```
> startx
```

Index

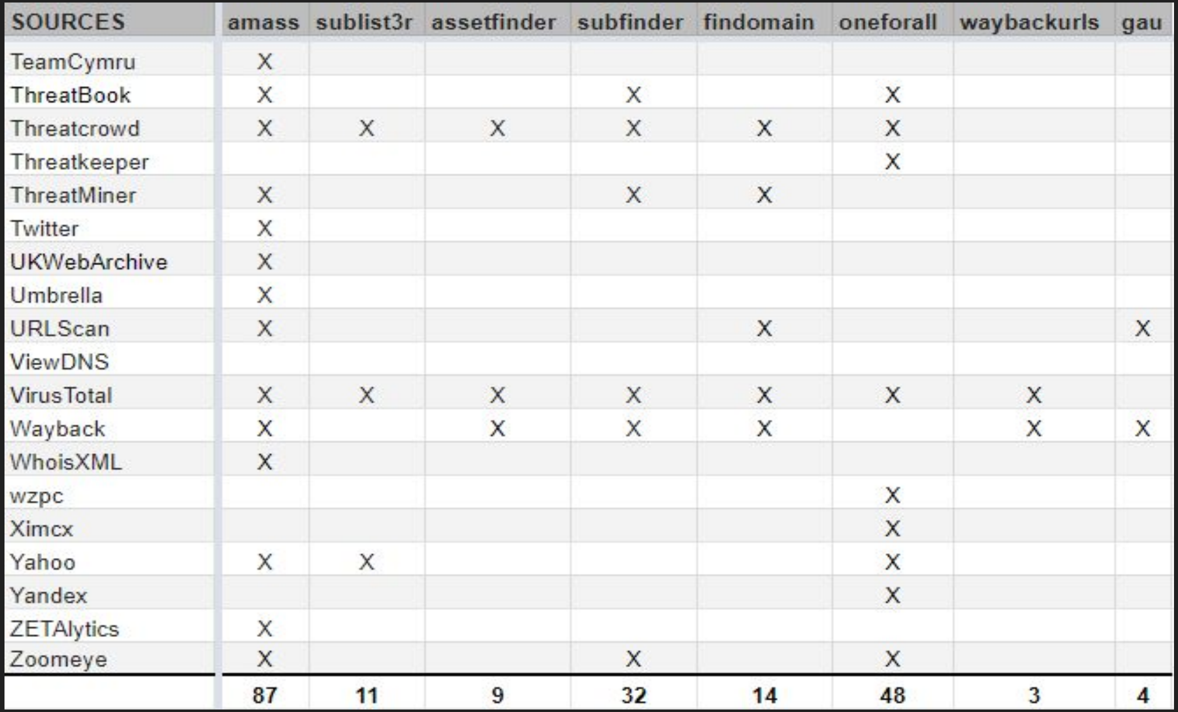
- Subdomains passive tools
- Amass for everything
- Resolvers? What?
- TLS handshake for profit
- JSON is 💖
- Google Analytics
- Archive + robots = win
- Plain text passwords for free
- Q&A

Subdomains passive tools

- Scraping data from 3rd parties
- Free and paid token model
- Daily/monthly quota
- > 70% similar data



Subdomains passive tools



SOURCES	amass	sublist3r	assetfinder	subfinder	findomain	oneforall	waybackurls	gau
TeamCymru	X							
ThreatBook	X			X		X		
Threatcrowd	X	X	X	X	X	X		
Threatkeeper						X		
ThreatMiner	X			X	X			
Twitter	X							
UKWebArchive	X							
Umbrella	X							
URLScan	X				X			X
ViewDNS								
VirusTotal	X	X	X	X	X	X	X	
Wayback	X		X	X	X		X	X
WhoisXML	X							
wzpc						X		
Ximcx						X		
Yahoo	X	X				X		
Yandex						X		
ZETAlytics	X							
Zoomeye	X			X		X		
	87	11	9	32	14	48	3	4

https://docs.google.com/spreadsheets/d/1ssuiovzgvFH2aTK-ymrxy2VezvHw5fnvXxBMH3Jnok4/edit?usp=sharing

Amass for everything

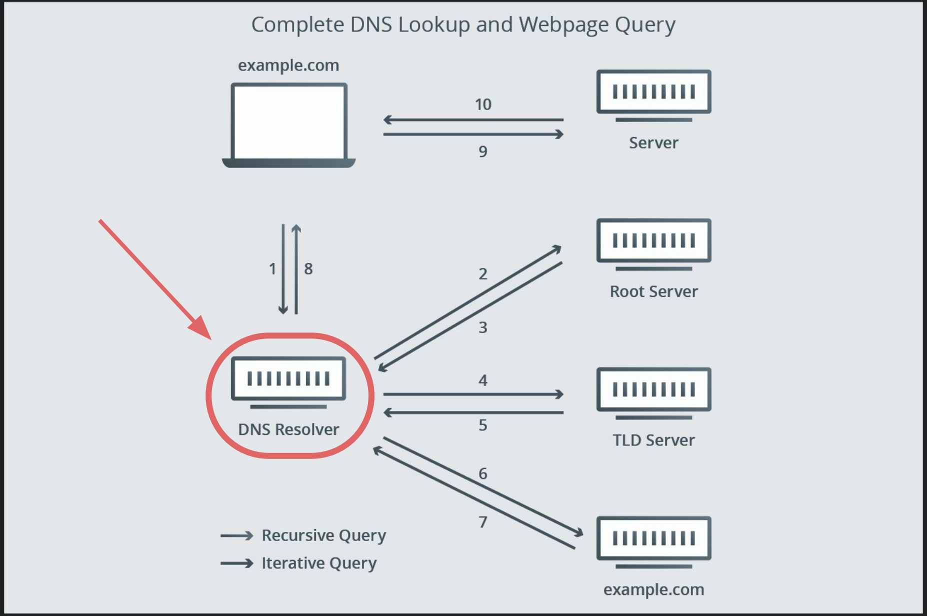
- Swiss army knife for domain recon
- Mostly used for passive subdomains extraction
- Since v3.17 use it for resolving

```
amass enum -d domain.com -w subs.txt -aw perms.txt -rf resolvers.txt -trf resolvers_trusted.txt -trqps 300 -dir output
```

```
amass enum -d domain.com \                      # Define the target
-w subdomains.txt \                             # Wordlist for bruteforcing
-aw permutations_list.txt \                     # Wordlist for alterations
-rf resolvers.txt \                             # Resolvers list
-trf resolvers_trusted.txt \                    # Trusted resolvers list
-trqps 300 \                                    # Trusted resolvers limit
-dir amass_output                               # Output directory
```

Resolvers? What?

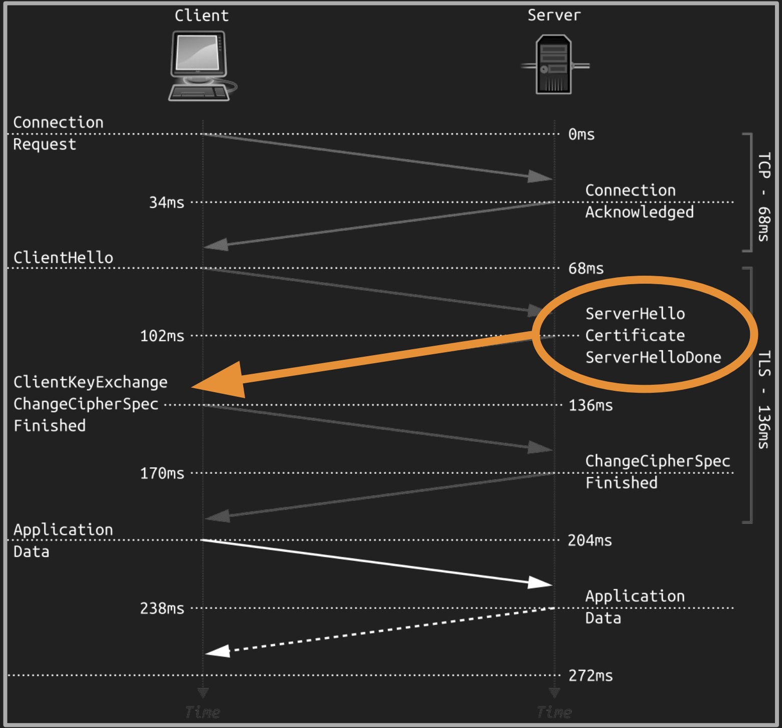
- DNS resolving or BF
- Two types
  - Standard
  - Trusted
- Test
  - 7K resolvers: 25K qps
  - 200 resolvers: 1,5K qps
- Tools
  - dnsvalidator
  - trickest/resolvers



Cloudflare

TLS handshake domain scraping

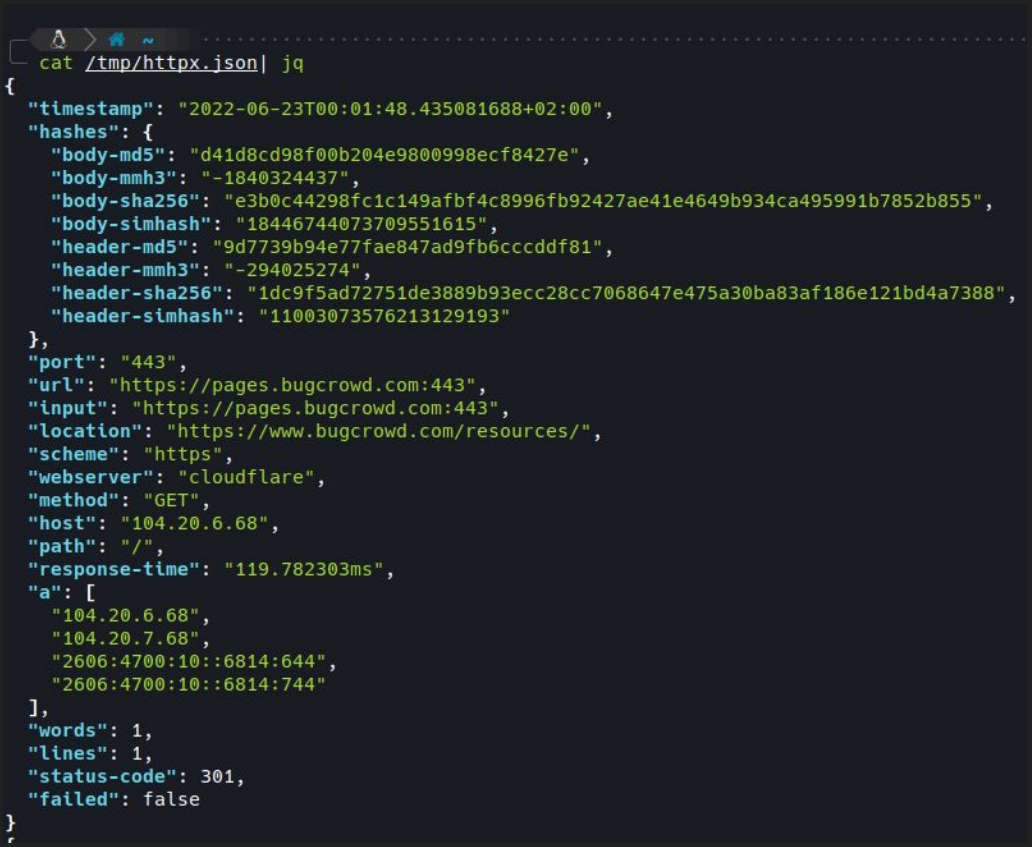
- Collect domains from TLS protocol
- Works on any service with TLS layer
- +140 TLS ports according IANA list
- Tool name: cero



Wikimedia

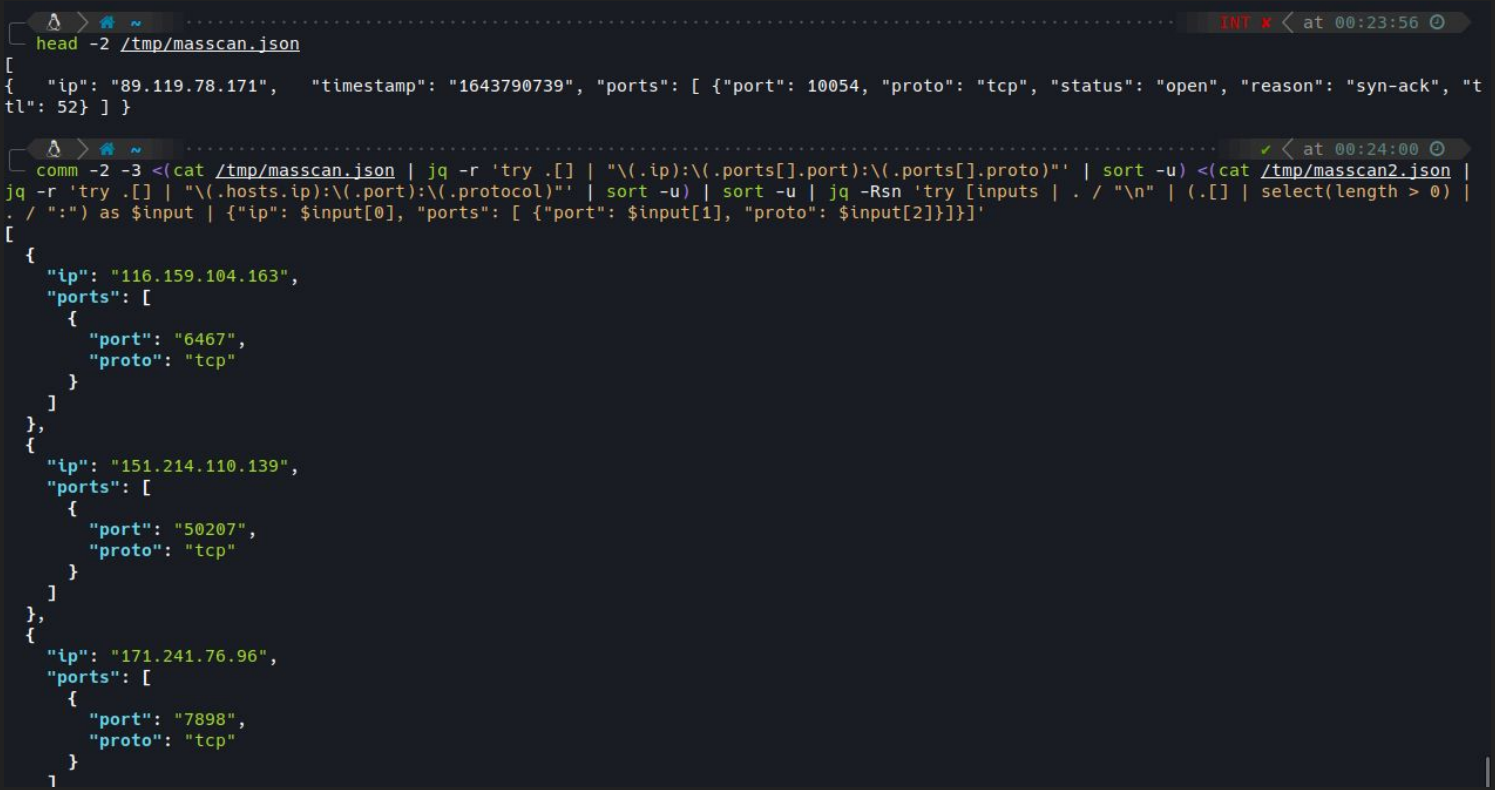
JSON is 💖

- Easy to read & parse
- Multiple tools:
  - amass
  - ffuf
  - theHarvester
  - masscan
  - pdiscovery tools
- jq



```
cat /tmp/httpx.json| jq
{
  "timestamp": "2022-06-23T00:01:48.435081688+02:00",
  "hashes": {
    "body-md5": "d41d8cd98f00b204e9800998ecf8427e",
    "body-mmh3": "-1840324437",
    "body-sha256": "e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855",
    "body-simhash": "18446744073709551615",
    "header-md5": "9d7739b94e77fae847ad9fb6cccddf81",
    "header-mmh3": "-294025274",
    "header-sha256": "1dc9f5ad72751de3889b93ecc28cc7068647e475a30ba83af186e121bd4a7388",
    "header-simhash": "11003073576213129193"
  },
  "port": "443",
  "url": "https://pages.bugcrowd.com:443",
  "input": "https://pages.bugcrowd.com:443",
  "location": "https://www.bugcrowd.com/resources/",
  "scheme": "https",
  "webserver": "cloudflare",
  "method": "GET",
  "host": "104.20.6.68",
  "path": "/",
  "response-time": "119.782303ms",
  "a": [
    "104.20.6.68",
    "104.20.7.68",
    "2606:4700:10::6814:644",
    "2606:4700:10::6814:744"
  ],
  "words": 1,
  "lines": 1,
  "status-code": 301,
  "failed": false
}
```

JSON is 💖

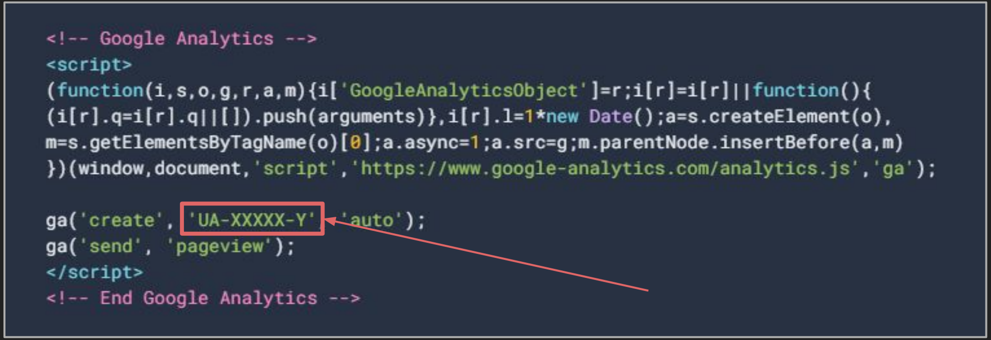


```
                                                                              INT ✘ < at 00:23:56
head -2 /tmp/masscan.json
[
{   "ip": "89.119.78.171",   "timestamp": "1643790739", "ports": [ {"port": 10054, "proto": "tcp", "status": "open", "reason": "syn-ack", "t
tl": 52} ] }

                                                                                  ✔ < at 00:24:00
comm -2 -3 <(cat /tmp/masscan.json | jq -r 'try .[] | "\(.ip):\(.ports[].port):\(.ports[].proto)"' | sort -u) <(cat /tmp/masscan2.json |
jq -r 'try .[] | "\(.hosts.ip):\(.port):\(.protocol)"' | sort -u) | sort -u | jq -Rsn 'try [inputs | . / "\n" | (.[] | select(length > 0) |
. / ":") as $input | {"ip": $input[0], "ports": [ {"port": $input[1], "proto": $input[2]}]}]'
[
  {
    "ip": "116.159.104.163",
    "ports": [
      {
        "port": "6467",
        "proto": "tcp"
      }
    ]
  },
  {
    "ip": "151.214.110.139",
    "ports": [
      {
        "port": "50207",
        "proto": "tcp"
      }
    ]
  },
  {
    "ip": "171.241.76.96",
    "ports": [
      {
        "port": "7898",
        "proto": "tcp"
      }
    ]
```

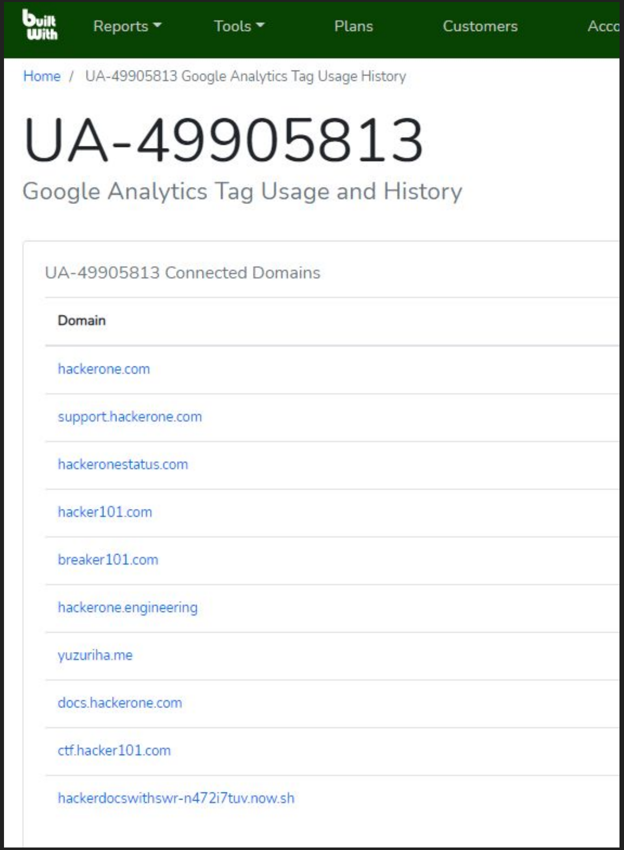
Google Analytics

- Loaded on the website’s JS
- Same ID per user
- Service BuiltWith
- Tool AnalyticsRelationships



```
<!-- Google Analytics -->
<script>
(function(i,s,o,g,r,a,m){i['GoogleAnalyticsObject']=r;i[r]=i[r]||function(){
(i[r].q=i[r].q||[]).push(arguments)},i[r].l=1*new Date();a=s.createElement(o),
m=s.getElementsByTagName(o)[0];a.async=1;a.src=g;m.parentNode.insertBefore(a,m)
})(window,document,'script','https://www.google-analytics.com/analytics.js','ga');

ga('create', 'UA-XXXXX-Y', 'auto');
ga('send', 'pageview');
</script>
<!-- End Google Analytics -->
```



Home / UA-49905813 Google Analytics Tag Usage History

UA-49905813

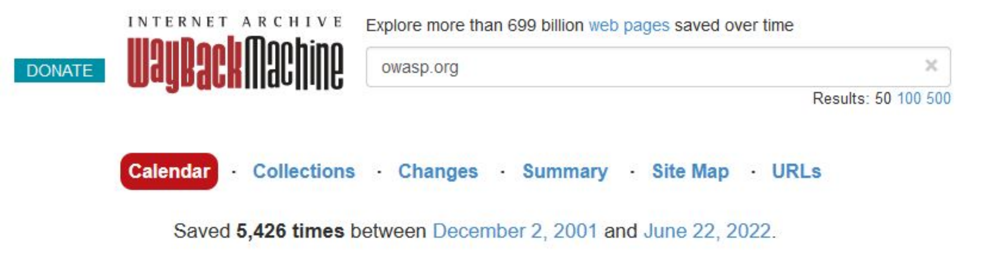
Google Analytics Tag Usage and History

UA-49905813 Connected Domains

Domain
hackerone.com
support.hackerone.com
hackeronestatus.com
hacker101.com
breaker101.com
hackerone.engineering
yuzuriha.me
docs.hackerone.com
ctf.hacker101.com
hackerdocswithswr-n472i7tuv.now.sh

Archive + robots = win

- waybackMachine
- Historic robots.txt
- Grab all disallow entries
- Wordlist for fuzzing
- Tool roboxtractor

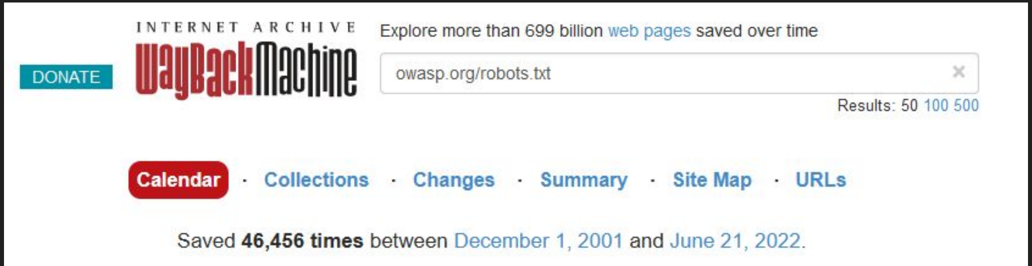


Explore more than 699 billion web pages saved over time

owasp.org

Results: 50 100 500

Saved **5,426 times** between December 2, 2001 and June 22, 2022.



Explore more than 699 billion web pages saved over time

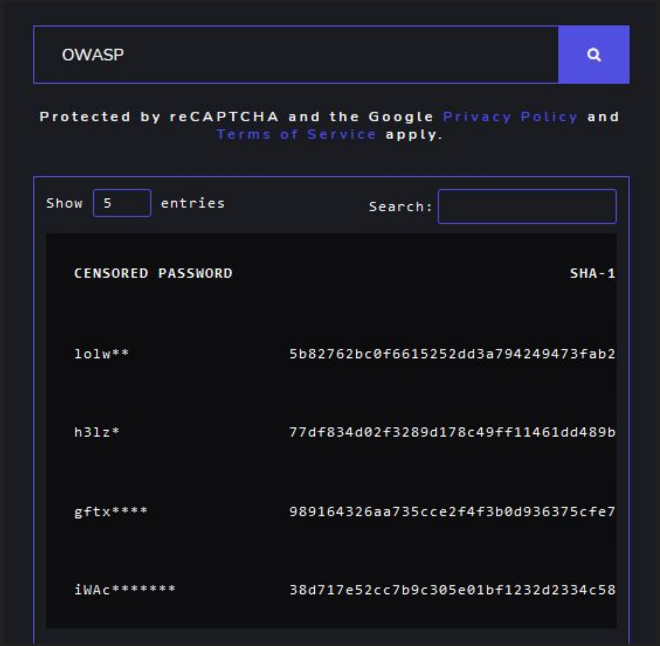
owasp.org/robots.txt

Results: 50 100 500

Saved **46,456 times** between December 1, 2001 and June 21, 2022.

Plain text passwords for free

- breachdirectory.com + hashes.com

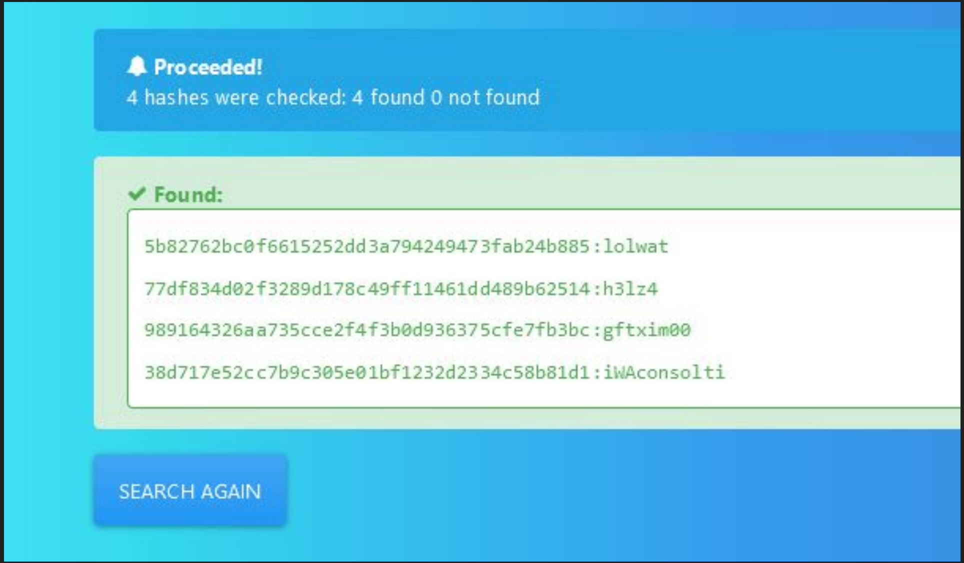


OWASP

Protected by reCAPTCHA and the Google Privacy Policy and Terms of Service apply.

Show 5 entries   Search:

CENSORED PASSWORD	SHA-1
lolw**	5b82762bc0f6615252dd3a794249473fab2
h3lz*	77df834d02f3289d178c49ff11461dd489b
gftx****	989164326aa735cce2f4f3b0d936375cfe7
iWAc*******	38d717e52cc7b9c305e01bf1232d2334c58



Proceeded!

4 hashes were checked: 4 found 0 not found

✓ Found:

```
5b82762bc0f6615252dd3a794249473fab24b885:lolwat
77df834d02f3289d178c49ff11461dd489b62514:h3lz4
989164326aa735cce2f4f3b0d936375cfe7fb3bc:gftxim00
38d717e52cc7b9c305e01bf1232d2334c58b81d1:iWAconsolti
```

Q&A

Thanks