

# A GUIDE TO SUBDOMAIN TAKEOVERS

Hacker Resources

🕒 Aug 15 2018

👤 EdOverflow

SHARE [f](#) [t](#) [in](#)

HackerOne's Hactivity feed — a curated feed of publicly-disclosed reports — has seen its fair share of subdomain takeover reports. Since Detectify's [fantastic series on subdomain takeovers](#), the bug bounty industry has seen a rapid influx of reports concerning this type of issue. The basic premise of a subdomain takeover is a host that points to a particular service not currently in use, which an adversary can use to serve content on the vulnerable subdomain by setting up an account on the third-party service. As a hacker and a security analyst, I deal with this type of issue on a daily basis. My goal today is to create an overall guide to understanding, finding, exploiting, and reporting subdomain misconfigurations. This article assumes that the reader has a basic understanding of the [Domain Name System](#) (DNS) and knows how to set up a subdomain.

## INTRODUCTION TO SUBDOMAIN TAKEOVERS

We use cookies to collect information to help us personalise your experience and improve the functionality and performance of our site. By continuing to use our site, you consent to our use of cookies. For more information see our [cookies policy](#). [I AGREE](#)

```
example.com
```

is the target and that the team running

```
example.com
```

have a bug bounty programme. While enumerating all of the subdomains belonging to

```
example.com
```

— a process that we will explore later — a hacker stumbles across

```
subdomain.example.com
```

, a subdomain pointing to GitHub pages. We can determine this by reviewing the subdomain's DNS records; in this example,

```
subdomain.example.com
```

has multiple A records pointing to GitHub's **dedicated IP addresses for custom pages**.

```
$ host subdomain.example.com
```

```
subdomain.example.com has address 192.30.252.153
```

```
subdomain.example.com has address 192.30.252.154
```

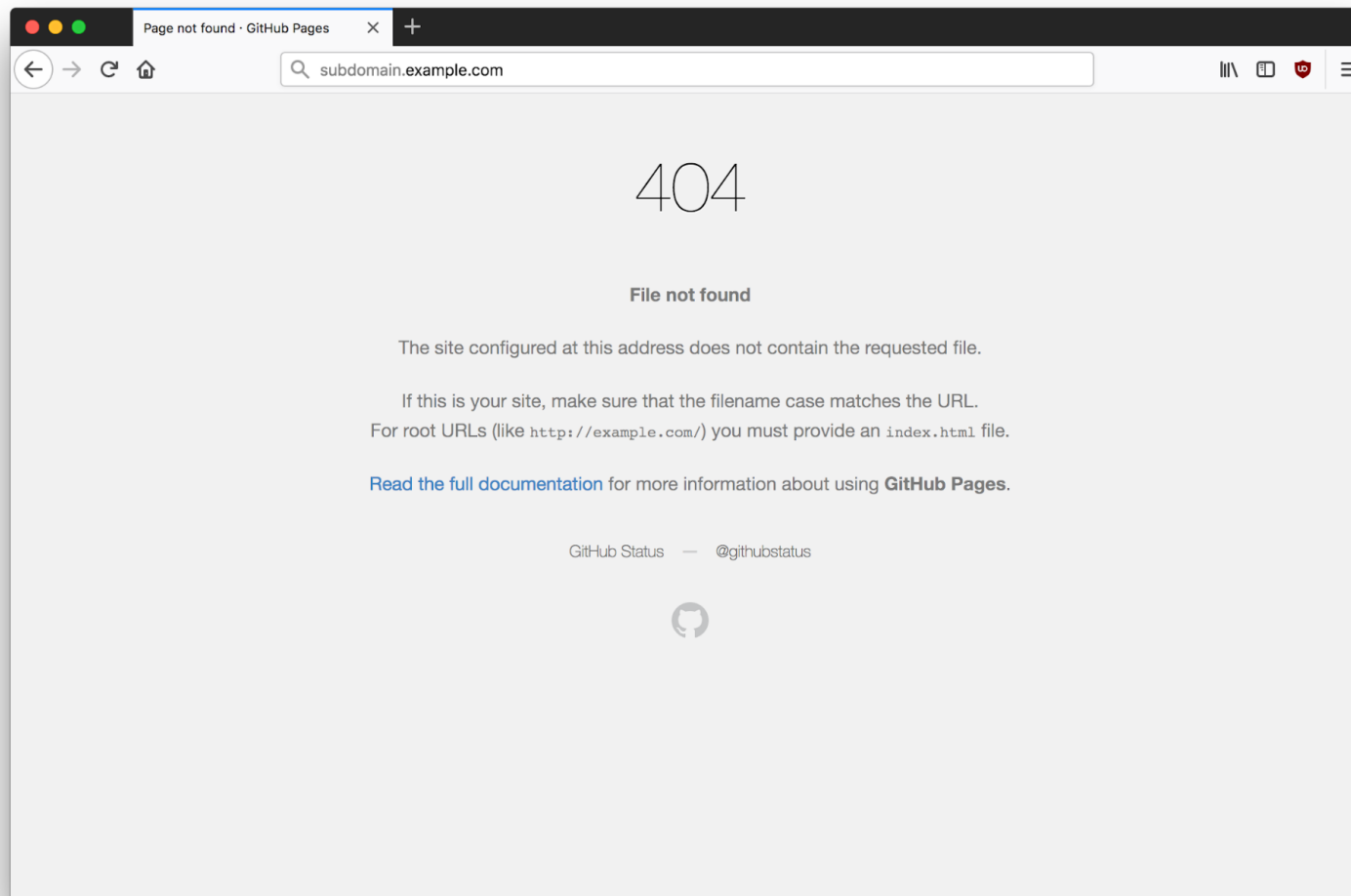
```
$ whois 192.30.252.153 | grep "OrgName"
```

```
OrgName: GitHub, Inc.
```

When navigating to

We use cookies to collect information to help us personalise your experience and improve the functionality and performance of our site. By continuing to use our site, you consent to our use of cookies. For more information see our [cookies policy](#). [I AGREE](#)

, we discover the following 404 error page.



Most hackers' senses start tingling at this point. This 404 page indicates that no content is being served under the top-level directory and that we should attempt to add this subdomain to our personal GitHub repository.

We use cookies to collect information to help us personalise your experience and improve the functionality and performance of our site. By continuing to use our site, you consent to our use of cookies. For more information see our [cookies policy](#). [I AGREE](#)

We use cookies to collect information to help us personalise your experience and improve the functionality and performance of our site. By continuing to use our site, you consent to our use of cookies. For more information see our [cookies policy](#). [I AGREE](#)

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

✓ Your site is published at <http://subdomain.example.com/>

**Source**

Your GitHub Pages site is currently being built from the master branch. [Learn more.](#)

master branch ▾ Save

**Theme Chooser**

Select a theme to publish your site with a Jekyll theme. [Learn more.](#)

Choose a theme

**Custom domain**

Custom domains allow you to serve your site from a domain other than doesfranshaveashe11.com. [Learn more.](#)

subdomain.example.com Save

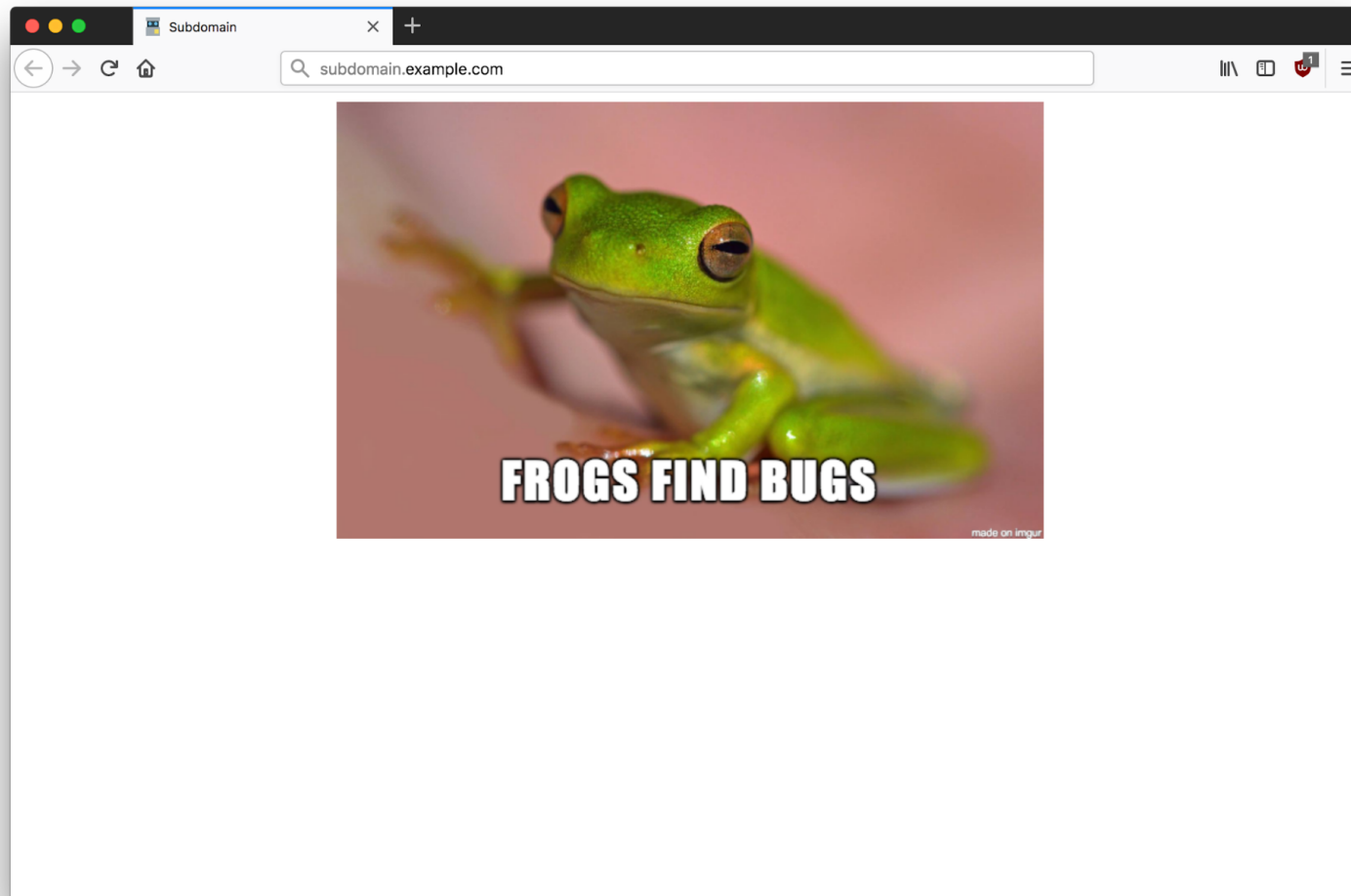
☐ **Enforce HTTPS** — Unavailable for your site because your domain is not properly configured to support HTTPS ([subdomain.example.com](#)) — [Troubleshooting custom domains](#)

HTTPS provides a layer of encryption that prevents others from snooping on or tampering with traffic to your site. When HTTPS is enforced, your site will only be served over HTTPS. [Learn more.](#)

Once the custom subdomain has been added to our GitHub project, we can see that the contents of the repository are served on

We use cookies to collect information to help us personalise your experience and improve the functionality and performance of our site. By continuing to use our site, you consent to our use of cookies. For more information see our [cookies policy](#). [I AGREE](#)

— we have successfully claimed the subdomain. For demonstration purposes, the index page now displays a picture of a frog.

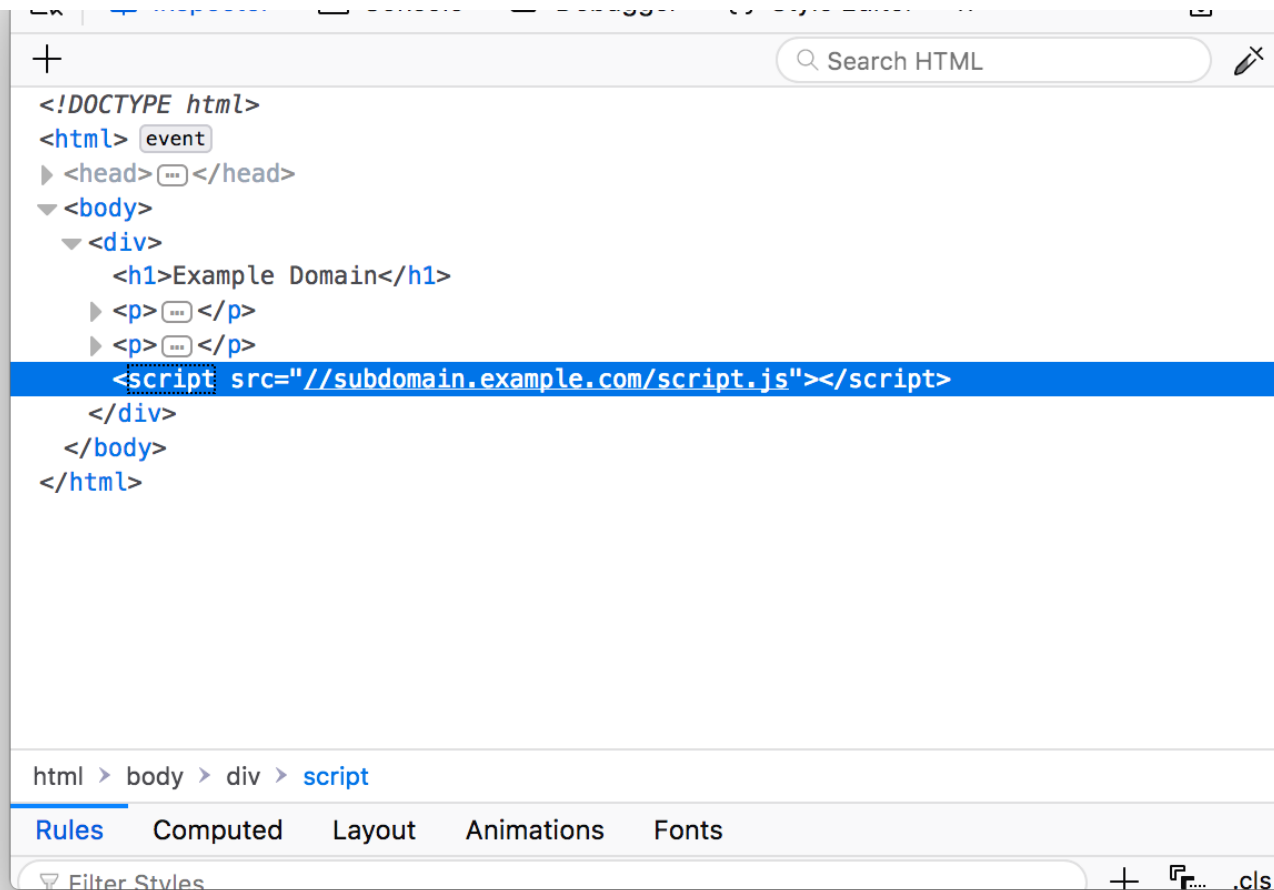


## SECOND-ORDER SUBDOMAIN TAKEOVERS

We use cookies to collect information to help us personalise your experience and improve the functionality and performance of our site. By continuing to use our site, you consent to our use of cookies. For more information see our [cookies policy](#). [I AGREE](#)

that a resource is being imported on the target page, for example, via a blob of JavaScript and the hacker can claim the subdomain from which the resource is being imported. Hijacking a host that is used somewhere on the page can ultimately lead to stored cross-site scripting, since the adversary can load arbitrary client-side code on the target page. The reason why I wanted to list this issue in this guide, is to highlight the fact that, as a hacker, I do not want to only restrict myself to subdomains on the target host. You can easily expand your scope by inspecting source code and mapping out all the hosts that the target relies on.

We use cookies to collect information to help us personalise your experience and improve the functionality and performance of our site. By continuing to use our site, you consent to our use of cookies. For more information see our [cookies policy](#). [I AGREE](#)



This is also the reason why, if you manage to hijack a subdomain, it is worth investing time to see if any pages import assets from your subdomain.




## SUBDOMAIN ENUMERATION AND DISCOVERY

We use cookies to collect information to help us personalise your experience and improve the functionality and performance of our site. By continuing to use our site, you consent to our use of cookies. For more information see our [cookies policy](#). [I AGREE](#)



Before diving right in, we must first differentiate between scraping and brute forcing, as both of these processes can help you discover subdomains, but can have different results. Scraping is a passive reconnaissance technique whereby one uses external services and sources to gather subdomains belonging to a specific host. Some services, such as DNS Dumpster and VirusTotal, index subdomains that have been crawled in the past allowing you to collect and sort the results quickly without much effort.

We use cookies to collect information to help us personalise your experience and improve the functionality and performance of our site. By continuing to use our site, you consent to our use of cookies. For more information see our [cookies policy](#). [I AGREE](#)

HTTP: Varnish HTTPS: Varnish		
alb.reddit.com  HTTP: Microsoft-IIS/8.5 HTTPS: Microsoft-IIS/8.5	34.198.84.247 ec2-34-198-84-247.compute-1.amazonaws.com	AS14618 Amazon.com, Inc. United States
out.reddit.com  HTTP: Microsoft-IIS/8.5 HTTPS: Microsoft-IIS/8.5	34.233.195.79 ec2-34-233-195-79.compute-1.amazonaws.com	AS14618 Amazon.com, Inc. United States
origin.reddit.com  	34.237.33.164 ec2-34-237-33-164.compute-1.amazonaws.com	AS14618 Amazon.com, Inc. United States
rhs.reddit.com  	34.237.33.164 ec2-34-237-33-164.compute-1.amazonaws.com	AS14618 Amazon.com, Inc. United States
mailp236.reddit.com  	184.173.153.236 mail-p236.reddit.com	AS36351 SoftLayer Technologies Inc. United States
mx02.reddit.com  	52.205.61.79 mx-02.reddit.com	AS14618 Amazon.com, Inc. United States
mx03.reddit.com  	54.172.97.247 mx-03.reddit.com	AS14618 Amazon.com, Inc. United States

Results for subdomains belonging to

reddit.com

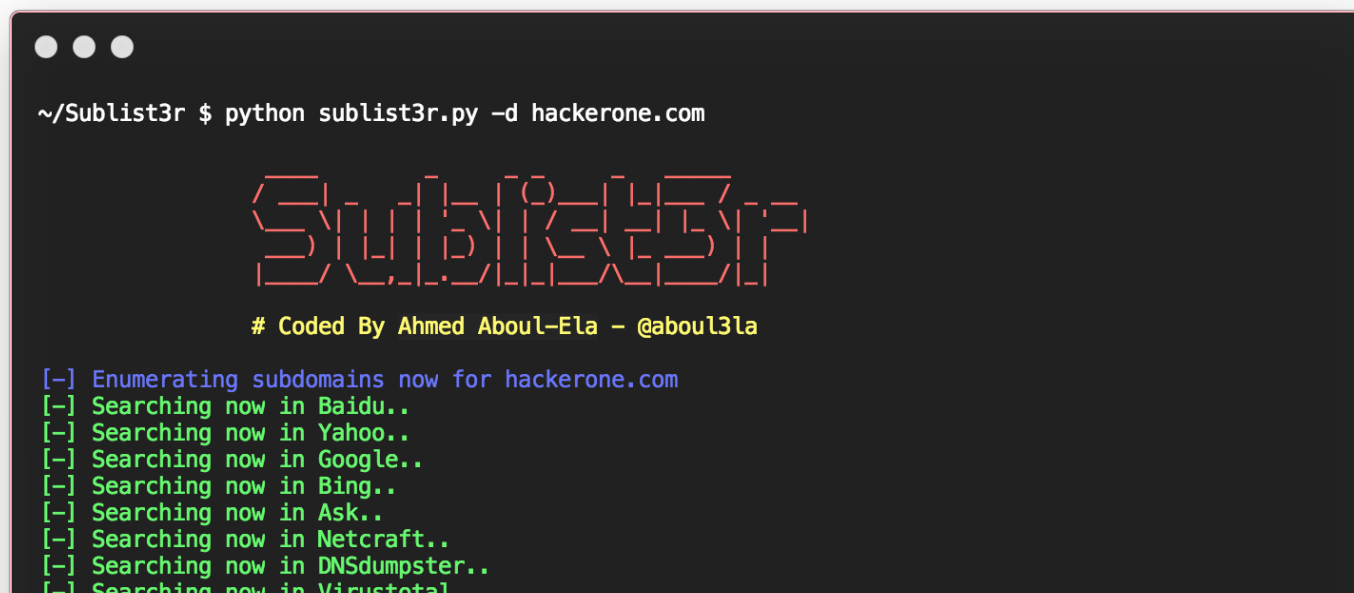
on DNS Dumpster.

We use cookies to collect information to help us personalise your experience and improve the functionality and performance of our site. By continuing to use our site, you consent to our use of cookies. For more information see our [cookies policy](#). [I AGREE](#)

Content Security Policy headers, source code, issue trackers, etc. The list of sources is endless, and I constantly discover new methods for increasing my results. Often you will find that the more peculiar your technique is, the more likely you will end up finding something that nobody else has come across; so be creative and test your ideas in practice against vulnerability disclosure programmes.

**Sublist3r** by Ahmed Aboul-Ela is arguably the simplest subdomain scraping tool that comes to mind. This light-weight Python script gathers subdomains from numerous search engines, SSL certificates, and websites such as *DNS Dumpster*. The set-up process on my personal machine was as straightforward as:

```
$ git clone https://github.com/aboul3la/Sublist3r.git
$ cd Sublist3r
$ sudo pip install -r requirements.txt
```



```
~/Sublist3r $ python sublist3r.py -d hackerone.com

Sublist3r

# Coded By Ahmed Aboul-Ela - @aboul3la

[-] Enumerating subdomains now for hackerone.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..
```

We use cookies to collect information to help us personalise your experience and improve the functionality and performance of our site. By continuing to use our site, you consent to our use of cookies. For more information see our [cookies policy](#). [I AGREE](#)

```
www.hackerone.com
api.hackerone.com
go.hackerone.com
info.hackerone.com
links.hackerone.com
a.ns.hackerone.com
b.ns.hackerone.com
support.hackerone.com
~/Sublist3r $
```

When brute forcing subdomains, the hacker iterates through a wordlist and based on the response can determine whether or not the host is valid. Please note, that it is very important to always check if the target has a wildcard enabled, otherwise you will end up with a lot of false-positives. Wildcards simply mean that all subdomains will return a response which skews your results. You can easily detect wildcards by requesting a seemingly random hostname that the target most probably has not set up.

```
$ host randomifje8z193hf8jafvh7g4q79gh274.example.com
```

For the best results while brute forcing subdomains, I suggest creating your own personal wordlist with terms

We use cookies to collect information to help us personalise your experience and improve the functionality and performance of our site. By continuing to use our site, you consent to our use of cookies. For more information see our [cookies policy](#). [I AGREE](#)

vulnerable Atlassian Jira and GIT instances.

If you are planning on brute forcing subdomains, I highly recommend taking a look at Jason Haddix's [word list](#). Jason went to all the trouble of merging lists from subdomain discovery tools into one extensive list.

## FINGERPRINTING

To increase your results when it comes to finding subdomains, no matter if you are scraping or brute forcing, one can use a technique called fingerprinting. Fingerprinting allows you to create a custom word list for your target and can reveal assets belonging to the target that you would not find using a generic word list.

## NOTABLE TOOLS

There is a wide variety of tools out there for subdomain takeovers. This section contains some notable ones that have not been mentioned so far.

### Altdns

In order to recursively brute force subdomains, take a look at Shubham Shah's [Altdns](#) script. Running your custom word list after fingerprinting a target through Altdns can be extremely rewarding. I like to use Altdns to generate word lists to then run through other tools.

### Commonspeak

Yet another tool by Shubham, [Commonspeak](#) is a tool to generate word lists using Google's [BigQuery](#). The goal is to generate word lists that reflect current trends, which is particularly important in a day and age where

technology is rapidly evolving. It is worth reading <https://pentester.io/commonspeak-bigquery-wordlists/> if

We use cookies to collect information to help us personalise your experience and improve the functionality and performance of our site. By continuing to use our site, you consent to our use of cookies. For more information see our [cookies policy](#). [I AGREE](#)

# SubFinder

A tool that combines both scraping and brute forcing beautifully is **SubFinder**. I have found myself using SubFinder more than Sublist3r now as my general-purpose subdomain discovery tool. In order to get better results, make sure to include API keys for the various services that SubFinder scrapes to find subdomains.

## Massdns

**Massdns** is a blazing fast subdomain enumeration tool. What would take a quarter of an hour with some tools, Massdns can complete in a minute. Please note, if you are planning on running Massdns, make sure you provide it with a list of valid resolvers. Take a look at <https://public-dns.info/nameservers.txt>, play around with the resolvers, and see which ones return the best results. If you do not update your list of resolvers, you will end up with a lot of false-positives.

```
$ ./scripts/subbrute.py lists/names.txt example.com | ./bin/massdns -r lists/resolvers.txt -t A -o S -w results.txt
```

## AUTOMATING YOUR WORKFLOW

When hunting for subdomain takeovers, automation is key. The top bug bounty hunters constantly monitor targets for changes and continuously have an eye on every single subdomain that they can find. For this guide, I do not believe it is necessary to focus on monitoring setups. Instead, I want to stick to simple tricks that can save you time and can be easily automated.

The first task that I love automating is filtering out live subdomains from a list of hosts. When scraping for

We use cookies to collect information to help us personalise your experience and improve the functionality and performance of our site. By continuing to use our site, you consent to our use of cookies. For more information see our [cookies policy](#). **I AGREE**

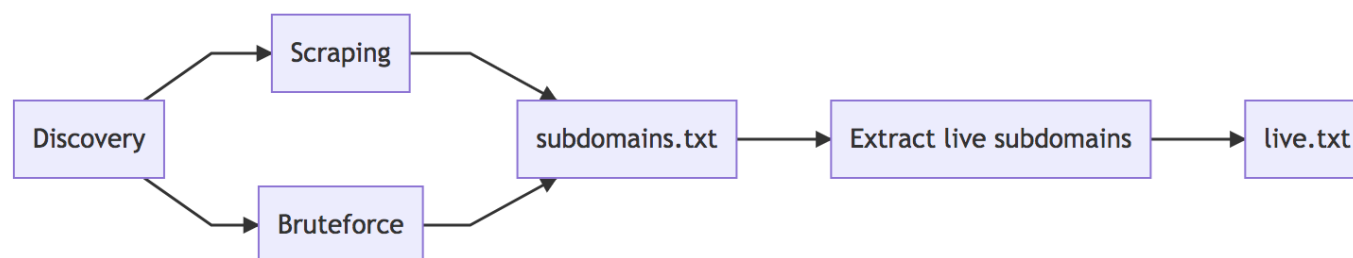
necessarily mean it cannot be hijacked. This task can easily be accomplished by using the

```
host
```

command — a subdomain that is no longer live will return an error.

```
while read subdomain; do
if host "$subdomain" > /dev/null; then
# If host is live, print it into
# a file called "live.txt".
echo "$subdomain" >> live.txt
fi
done < subdomain-list.txt
```

So if we put everything that we have so far together, we end up with the following workflow.



The next step is to get an overview of the various subdomains. There are two options: The first is to run a screenshot script across all subdomains; the second one requires storing the contents of the page in a text file.

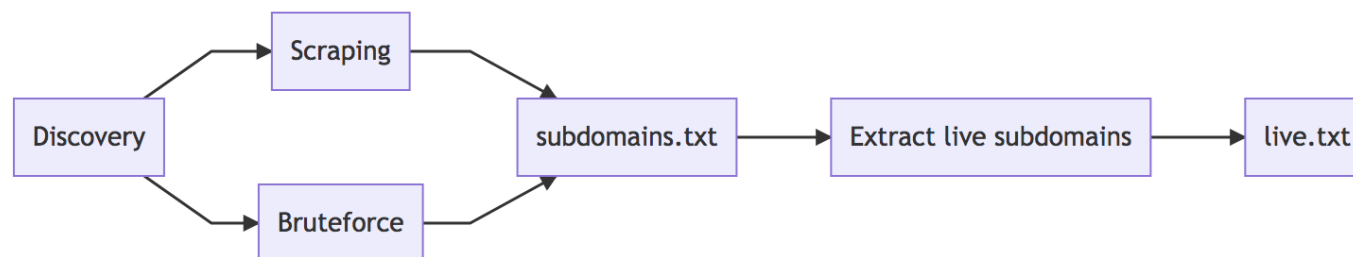
We use cookies to collect information to help us personalise your experience and improve the functionality and performance of our site. By continuing to use our site, you consent to our use of cookies. For more information see our [cookies policy](#). [I AGREE](#)

containing all the screenshots, response bodies, and headers from your list of hosts.

```
$ ./EyeWitness -f live.txt -d out --headless
```

EyeWitness can be a little too heavy for some cases and you might only want to store the page's contents via a simple GET request to the top-level directory of the subdomain. For cases like these, I use Tom Hudson's `meg`. `meg` sends requests concurrently and then store the output into plain-text files. This makes it a very efficient and light-weight solution for sieving through your subdomains and allows you to grep for keywords.

```
$ meg -d 10 -c 200 / live.txt
```



## Special cases

There is a special case that we need to look out for, one that Frans Rosén highlighted in his talk "[DNS hijacking using cloud providers – No verification needed](#)". Whenever you encounter dead DNS records, do not just assume that you cannot hijack that subdomain. As Frans points out, the

host

We use cookies to collect information to help us personalise your experience and improve the functionality and performance of our site. By continuing to use our site, you consent to our use of cookies. For more information see our [cookies policy](#). [I AGREE](#)



```
alg
```

will unveil the dead records.

## Exploitation

Right, now you control a subdomain belonging to the target, what can you do next? When determining plausible attack scenarios with a misconfigured subdomain, it is crucial to understand how the subdomain interacts with the base name and the target's core service.

## Cookies

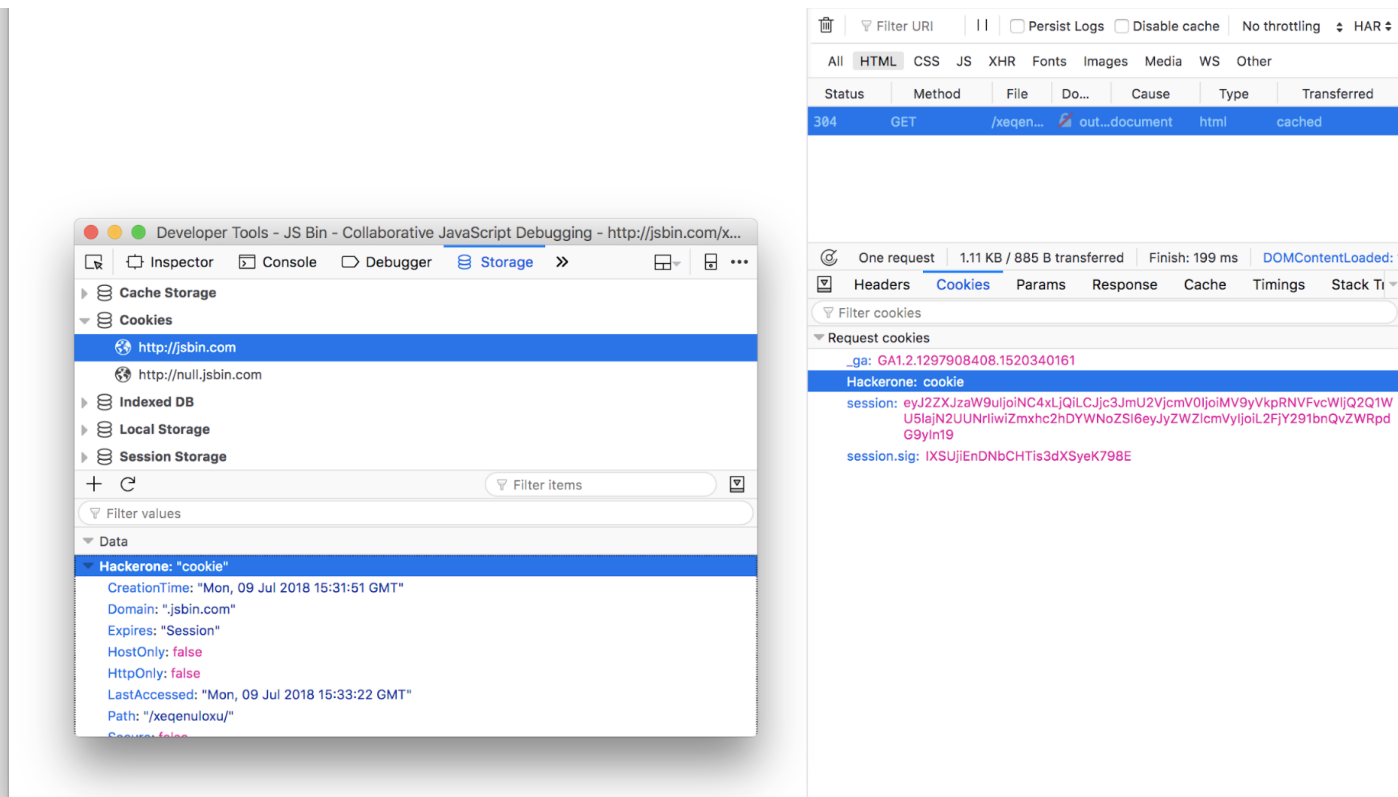
```
subdomain.example.com
```

can modify cookies scoped to

```
example.com
```

. This is important to remember as this could potentially allow you to hijack a victim's session on the base name.

We use cookies to collect information to help us personalise your experience and improve the functionality and performance of our site. By continuing to use our site, you consent to our use of cookies. For more information see our [cookies policy](#). [I AGREE](#)



From output.jsbin.com, we can set cookies for jsbin.com.

If the base name is vulnerable to session fixation and uses **HTTPOnly cookies**, you can set a cookie and then when the user restarts their browser, your malicious cookie will take precedence over the newly generated cookie because cookies are sorted by age.

## Cross-Origin Resource Sharing

We use cookies to collect information to help us personalise your experience and improve the functionality and performance of our site. By continuing to use our site, you consent to our use of cookies. For more information see our [cookies policy](#). [I AGREE](#)

data. Some applications permit subdomains to make cross-origin HTTP requests with the assumption that subdomains are trusted entities. When you hijack a subdomain look for CORS headers — Burp Suite Pro's scanner usually picks them up — and see if the application whitelists subdomains. This could allow you to steal data from an authenticated user on the main application.

## Oauth whitelisting

Similar to Cross-Origin Resource Sharing, the **Oauth** flow also has a whitelisting mechanic, whereby developers can specify which callback URLs should be accepted. The danger here once again is when subdomains have been whitelisted and therefore you can redirect users during the Oauth flow to your subdomain, potentially leaking their Oauth token.

## Content-Security Policies

The **Content-Security Policy** (CSP) is yet another list of hosts that an application trusts, but the goal here is to restrict which hosts can execute client-side code in the context of the application. This header is particularly useful if one wants to minimise the impact of cross-site scripting. If your subdomain is included in the whitelist, you can use your subdomain to bypass the policy and execute malicious client-side code on the application.

```
$ curl -sI https://hackerone.com | grep -i "content-security-policy"
content-security-policy: default-src 'none'; base-uri 'self'; block-all-mixed-content; child-src
www.youtube-nocookie.co
m; connect-src 'self' www.google-analytics.com errors.hackerone.net; font-src 'self'; form-action 'self';
frame-ancestor
s 'none'; img-src 'self' data: cover-photos.hackerone-user-content.com hackathon-photos.hackerone-
user-content.com profi
```

We use cookies to collect information to help us personalise your experience and improve the functionality and performance of our site. By continuing to use our site, you consent to our use of cookies. For more information see our [cookies policy](#). [I AGREE](#)

```
elf' hackerone-us-west-2-production-attachments.s3-us-west-2.amazonaws.com; script-src 'self'
www.google-analytics.com;
style-src 'self' 'unsafe-inline'; report-uri https://errors.hackerone.net/api/30/csp-report/?
sentry_key=61c1e2f50d21487c
97a071737701f598
```

## Clickjacking

As described in the "[Cure53 Browser Security White Paper](#)", Internet Explorer, Edge, and Safari support the

```
ALLOW-FROM
```

directive in the

```
X-Frame-Options
```

header, which means if your subdomain is whitelisted, you can frame the target page and therefore perform clickjacking attacks.

## Password managers

This is not necessarily one that you would include in a report, but it is worth noting that some password managers will automatically fill out login forms on subdomains belonging to the main application.

We use cookies to collect information to help us personalise your experience and improve the functionality and performance of our site. By continuing to use our site, you consent to our use of cookies. For more information see our [cookies policy](#). [I AGREE](#)

Just learned @LastPass autofills passwords on all subdomains by default. This is equivalent to a website setting a non-httponly overscoped cookie containing your plaintext password... ie bonkers  
lastpass.com/support.php?cm ...

7:54 am - 26 Jun 2018

22 Retweets 75 Likes



6 22 75

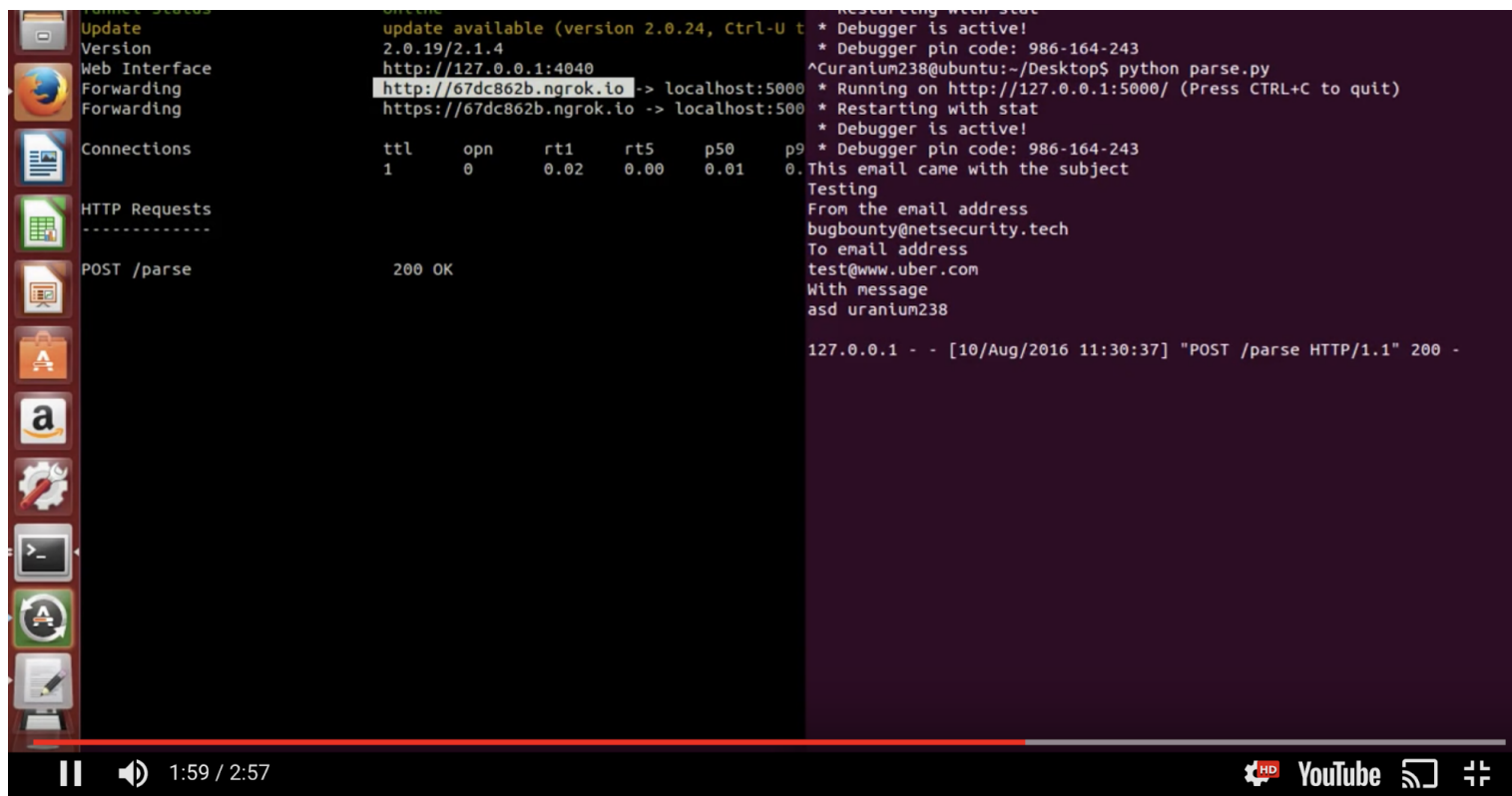
## Intercepting emails

Rojan Rijal **demonstrated** how he was able to intercept emails by claiming a subdomain belonging to uber.com on SendGrid.

Reading some Uber Emails: Status-Fixed



We use cookies to collect information to help us personalise your experience and improve the functionality and performance of our site. By continuing to use our site, you consent to our use of cookies. For more information see our [cookies policy](#). [I AGREE](#)



## REPORTING SUBDOMAIN TAKEOVERS

Before you even attempt to report a subdomain takeover, make sure that you are in fact able to serve content on the subdomain. However, whatever you do, do not publish anything on the index page, even if it is a harmless picture of a frog as demonstrated earlier. It is best practice to serve an HTML file on a hidden path containing a secret message in an HTML comment. That should be enough to demonstrate the issue when initially contacting the programme about your finding. Only once the team has given you permission, should you attempt to escalate the issue and actually demonstrate the overall impact of the vulnerability. In most cases though, the team should already be aware of the impact and your report should contain information

We use cookies to collect information to help us personalise your experience and improve the functionality and performance of our site. By continuing to use our site, you consent to our use of cookies. For more information see our [cookies policy](#). [I AGREE](#)

Take your time when writing up a report about a subdomain takeover as this type of issue can be extremely rewarding and nobody can beat you to the report since you are — hopefully — the only one that has control over the subdomain.

Author's note: I have only ever witnessed one duplicate report for a subdomain takeover, so while there is still the possibility, the chances of this ever happening to you are fairly slim.

We have reached the end of this guide and I look forward to triaging your subdomain takeover reports on HackerOne. Remember to practice and apply the tricks listed in this write-up when hunting for subdomain takeovers.

On a final note, I would like to thank [Frans Rosén](#), [Filedscriptor](#), [Mongo](#), and [Tom Hudson](#) for exchanging ideas concerning subdomain takeovers. Their research has been the foundation for a lot of what I had discovered throughout my journey.

[@EdOverflow](#)

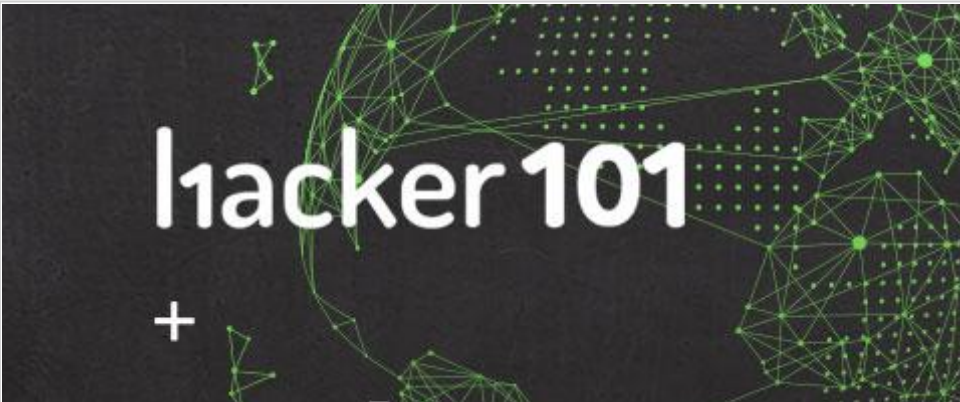
## RELATED POSTS

We use cookies to collect information to help us personalise your experience and improve the functionality and performance of our site. By continuing to use our site, you consent to our use of cookies. For more information see our [cookies policy](#). [I AGREE](#)



## HACKERS HAVE EARNED MORE THAN \$50M IN BUG BOUNTY CASH ON HACKERONE: TIME TO CELEBRATE!

[Read More>](#)



## TEST YOUR HACKING SKILLS ON REAL-WORLD SIMULATED BUGS

[Read More>](#)

We use cookies to collect information to help us personalise your experience and improve the functionality and performance of our site. By continuing to use our site, you consent to our use of cookies. For more information see our [cookies policy](#). [I AGREE](#)





## HACKER101 CTF++: FIND FLAGS, GET PRIVATE BUG BOUNTY PROGRAM INVITATIONS

[Read More>](#)

### FOR BUSINESS

[Product Overview](#)

[HackerOne Response](#)

[HackerOne Bounty](#)

[HackerOne Challenge](#)

[Services](#)

[Resources](#)

We use cookies to collect information to help us personalise your experience and improve the functionality and performance of our site. By continuing to use our site, you consent to our use of cookies. For more information see our [cookies policy](#). [I AGREE](#)

Live Hacking

Business Support

## FOR HACKERS

Start Hacking

Hacker101

Leaderboard

Hackitivity

Program Directory

Hacker Support

Disclosure Guidelines

Disclosure Assistance

## COMMUNITY

Community Edition

Internet Bug Bounty

Zero Daily Newsletter

## COMPANY

About Us

Events

Documentation

We use cookies to collect information to help us personalise your experience and improve the functionality and performance of our site. By continuing to use our site, you consent to our use of cookies. For more information see our [cookies policy](#). [I AGREE](#)



[Press](#)

[Careers](#)

[Customers](#)

[Partners](#)

[Security](#)

## **CONTACT**

[Contact Sales](#)

[Report a Bug](#)

[Support](#)

## **TRY HACKERONE**

[Start a Program](#)

[Start Hacking](#)



hackerone

2019 © HackerOne. [Terms](#) | [Privacy](#) | [Security](#)

We use cookies to collect information to help us personalise your experience and improve the functionality and performance of our site. By continuing to use our site, you consent to our use of cookies. For more information see our [cookies policy](#). [I AGREE](#)