



# Manual broad scope

## Why it matters

I often see people who want to become bug bounty hunters come across tools and guides on how to use them. Despite these tools often being open source, for a lot of hackers, they will remain black box forever. With this we mean that the tester will never truly understand how the tool works and that is a missed opportunity.

In my opinion, test automation is good, but testing manually while your tools run is even better. When we simply run our test automation tools we will miss a lot of issues. Test automation tools are programmed to look exactly at the location or execute exactly the strategy that the programmer wanted it to follow. This is not a bad thing at all but that programmer had a certain goal in mind and our goal might be different, which might require some adaptations of the tool. Above all this, the human eye beats any robot at finding details that are not correct. We need to get to know the process the automation tools implement in order to improve our automation strategies.

You all know that it's very important to keep moving forward. We can't stand still and if we just use the tools that others made without thinking about their implementation we miss a lot of context. Standing still is moving backwards.

## What is it?

Manual broad scope recon will try to go through the processes that the automation tools use so we can see how things work and potentially discover new ways to automate our hunting.

## How can we test for it

When we want to test our target we have several strategies that we learn and implement. I will show you the basics but you will have to practice for yourself to get good at these things.

## Google dorking

If we have a domain with a wide scope \*.target.com then we need to find all the subdomains to investigate which ones are potentially interesting to hack on. To do this automated tools like amass use different sources like [crt.sh](http://crt.sh), [google.com](http://google.com) and [yahoo.com](http://yahoo.com) to find all the possible subdomains and list them for you.

Uncle Rat always find it interesting to repeat this process manually on google to see how to hunt. This process is called google dorking. I start with a basic dork

- Site:\*.target.com -www

Then i will open the first subdomain that i see, if it's interesting, i will investigate and if not, i will move on the next google dork:

- Site:\*.target.com -www -register

This will remove the subdomain from my search results and i'll move on the next subdomain.

Whenever i see a login page i will directory brute force it, but limit the amount of requests to my target as to not unintentionally DoS it. I will also try default credentials to see if i can login using those.

I repeat this cycle until i am content that i've seen most subdomains and move on to the next chapter.

## Waybackmachine

This is an amazing tool, not just for archiving the internet, but also for use hackers to search and consult that history. Actually use the tool to see how it works before you use waybackurls.

INTERNET ARCHIVE

Explore more than 539 billion web pages saved over time

DONATE

WayBackMachine

Results: 50 100 500

Calendar

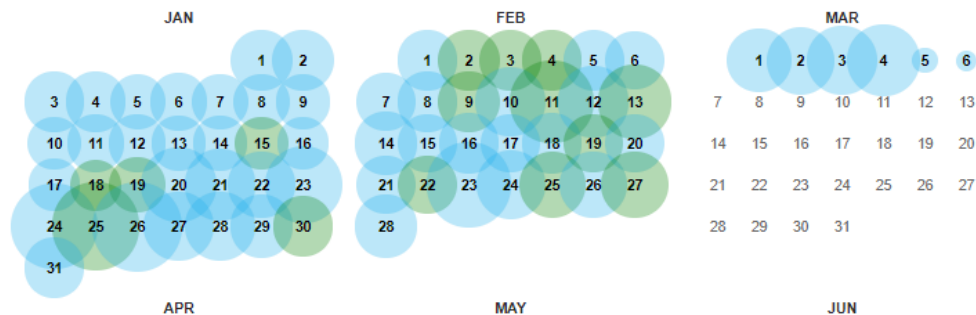
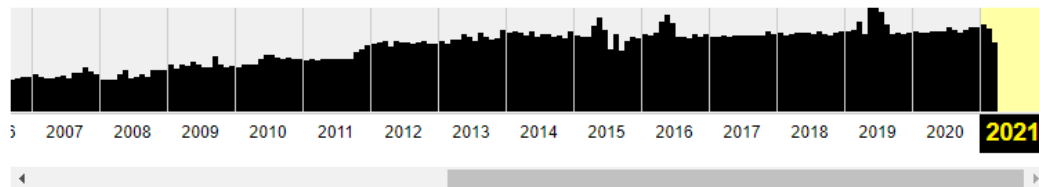
Collections

Changes

Summary

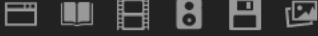

Site Map


Saved **5.921.997** times between **November 11, 1998** and **March 6, 2021**.




Your landing page will show whenever a new snapshot of a website was made. You can also adjust the timeline at the top. We are more interested in the other sections however.

We can use the waybackmachine to investigate a wide scope such as google.com but also a specific subdomain like www.google.com.

INTERNET ARCHIVE  SIGN UP | LOG IN  Search

ABOUT BLOG PROJECTS HELP DONATE  CONTACT JOBS VOLUNTEER PEOPLE

INTERNET ARCHIVE Explore more than 539 billion web pages saved over time

**WayBackMachine** 

DONATE  Results: 50 100 500

Calendar · Collections <sup>beta</sup> · Changes <sup>beta</sup> · **Summary** · Site Map

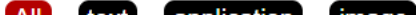
**host twitter.com**

Indexed on September 22, 2020.

**MIME-types**

**Year Start**

**Year End**



Scroll down in the history tab until you encounter the "Explore URLs from ..."

2020

All text application image

## Summary on MIME-types Count

Quick search on MIME-types...

<< < 1 2 > >>

	Captures	URLs	New URLs
text/html	77.707.807.056	75.211.272.674	74.457.448.605
application/json	6.195.271.357	6.088.795.710	6.086.776.219
text/plain	16.924.866	15	2
application/javascript	9.905.443	473.955	20.245
image/gif	4.565.992	4.454.772	4.449.963
image/vnd.microsoft.icon	97.828	9	0
application/xml	416.361	1.425	1.316
application/atom+xml	252.184	98.641	69.882
application/xhtml+xml	40.777	11.714	7.380
image/png	36.695	2.681	2.497

Explore twitter.com URLs

Captures



You can now explore your target in detail about more and see what pages have existed throughout the years. Some might still be up and vulnerable.

## Github dorking

We can apply the same principle we applied to google in our "google dorking" chapter to github. Developers will put anything on github if the companies are not careful and we can find them. You have a lot of fancy filter options available to you but

- organisation: google.com "api"

- user: the\_xss\_rat "key"
- user: the\_xss\_rat "secret"
- user: the\_xss\_rat "username"
- user: the\_xss\_rat "password"

If there is source code available, just go through it manually as well. Static code review is a treasure in bug bounties that so many neglect to take the time to do manually.

## Social media

Follow the developers and employees of your target on social media. They can do stuff like leak teams links that are open, leak feature releases, leak acquisitions,...

## Script kiddie VS hacker

### Script kiddie

- ▶ Only runs scripts other people write
- ▶ Never does original research
- ▶ Only ever goes for low hanging fruit
- ▶ Doesn't grow to their full potential

### Hacker

- ▶ Does manual work and translates that into automation
- ▶ Takes pride in digging deep
- ▶ Doesn't rest until they reach their full potential

## General manual recon tips

- Investigate ALL the subdomains you find, you can't predict what they hold so take the time to look into them.
  - If it's a static web page, move on
  - If you find functionality, test it using your regular main app methodology
- If you have any automation you want to run, start that up and then start your manual recon. This will save you time as you are both learning and running the automation you now understand much better.

- If the subdomain looks interesting, dig deep. Trust your rat instincts on this one. Your intuition is all you have to indicate if something is interesting or not. Everyone is different and we all hunt different.
- ☐ If you come across a new technology, look up how to hack it, this is how i did it in the beginning and if i didn't understand something i'd look that up ass well, untill i understood it.
- It would really help if you could write your own tools or improve existing ones and further strenghten our community if you have the chance.