Recon for bug bounty: 8 essential tools for performing effective reconnaissance

BY BLACKBIRD-EU · OCTOBER 15, 2024 · LAST UPDATED ON MARCH 6, 2025

We all know that reconnaissance is important in bug bounty, in fact, it is the most important phase in bug bounty & web app pentesting. Bug bounty hunters who perform effective recon are always rewarded well as they come across untouched features and hidden assets more often than others. This provides them an edge and easily increases their chances of finding security vulnerabilities.

However, not every bug bounty hunter can put in the effort and time to perform effective recon to map out everything that's in the scope of your target.

This article is here to help guide you to what it takes to perform better recon and find more security vulnerabilities on your favorite bug bounty program.

Importance of recon

It is clear that reconnaissance is important in bug bounty. It's the first phase of a security test or penetration test. Skipping this process may leave certain in-scope applications or functionalities in web apps untested, resulting in decreased chances of finding vulnerabilities.

However, recon is a repetitive and tedious task that has to be performed every time you start hunting on a new bug bounty program. Luckily for us, several automated open-source tools can help us make our lives a bit easier. Let's get into the top 8 most used automated tools by bug bounty hunters.

Automated tools

In this article, we will cover the top 8 most essential tools that we think you need to help you perform a comprehensive recon scan on your target. Each of these tools has its unique advantages but feel free to look up alternatives if you prefer an additional tool over the one listed below.

Let's start with the first step, performing asset enumeration using Amass.

Amass

Amass is an in-depth attack surface management open-source tool developed by OWASP that can be used to gather assets using both passive as well as active enumeration methods. It combines all these different sources and enumeration methodologies to help you find every host in your list of targets.

To perform a quick subdomain scan for a root domain (utilizing only passive sources), you can use the following command:

```
amass enum -d example.com -passive
```

Amass subdomain enumeration

In case you want to dive deeper into Amass and its full capabilities, we've created a detailed article for you to help you get started that you can read on our blog:

https://blog.intigriti.com/hacking-tools/hacker-tools-amass-hunting-for-subdomains

The tool is also easy to install and configure and is available on Github:

https://github.com/owasp-amass/amass

TIP! Make sure you configure your API keys in your configuration file to allow Amass to use more sources to provide you with better results! Some external sources provide free API keys with limited usage.

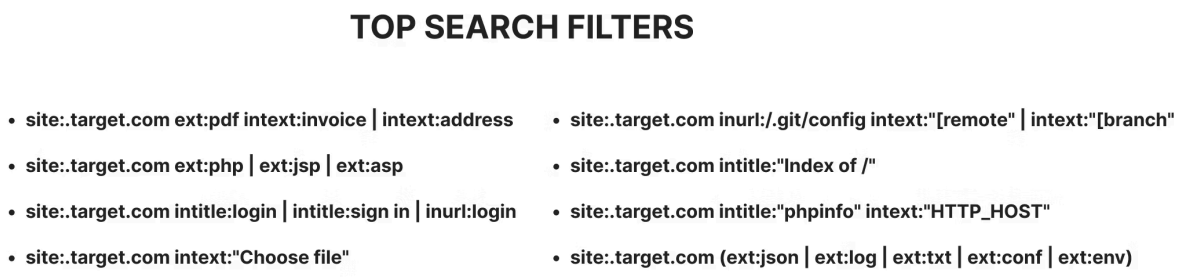
Google/Bing/GitHub dorking

Another way to find linked assets or domains to your target is to use search engines.

Search engines can be used to enumerate more information about our targets, from indexed files, login panels and admin portals to new subdomains. As a bug bounty hunter, you should always consider using Google, Bing and Github to your advantage to help you find more information.

Most used search filters

Here's a list of one of the most common search filters that you can use to find interesting indexed files such as files that commonly expose private data:



Common Google Dorking Search Filters

Github recon

An other commonly used service is Github, companies often make Github to deploy and host code and make use of the collaboration platform that Github provides.

This is why Github reconnaissance is invaluable as some companies accidentally push secrets in their public code bases, often providing access to admin portals with elevated privileges to unauthorized users. In other cases, you can find references to hidden assets, links, files or parameters.

However, Github recon can be a tedious task, especially if you're dealing with a big target and are unsure where to start or what to look for. Luckily for us, some tools are available that we can make use of such as Trufflehog and Gitleaks.

Trufflehog:

https://github.com/trufflesecurity/trufflehog

Gitleaks:

https://github.com/gitleaks/gitleaks

Eyewitness

After you've enumerated your list of targets, hosts and subdomains, it's time to filter out the non-resolving hosts. Eyewitness is an incredible tool to help you not only with probing live hosts and screenshotting them but also perform basic technology fingerprinting. This enables you to quickly fly over your list of targets and assess each host independently.

This process of flying over your target allows you to easily spot login (and admin) panels and just interesting or suspicious HTTP responses in general!

We've curated a detailed article on how to use Eyewitness, including a video! Check it out:

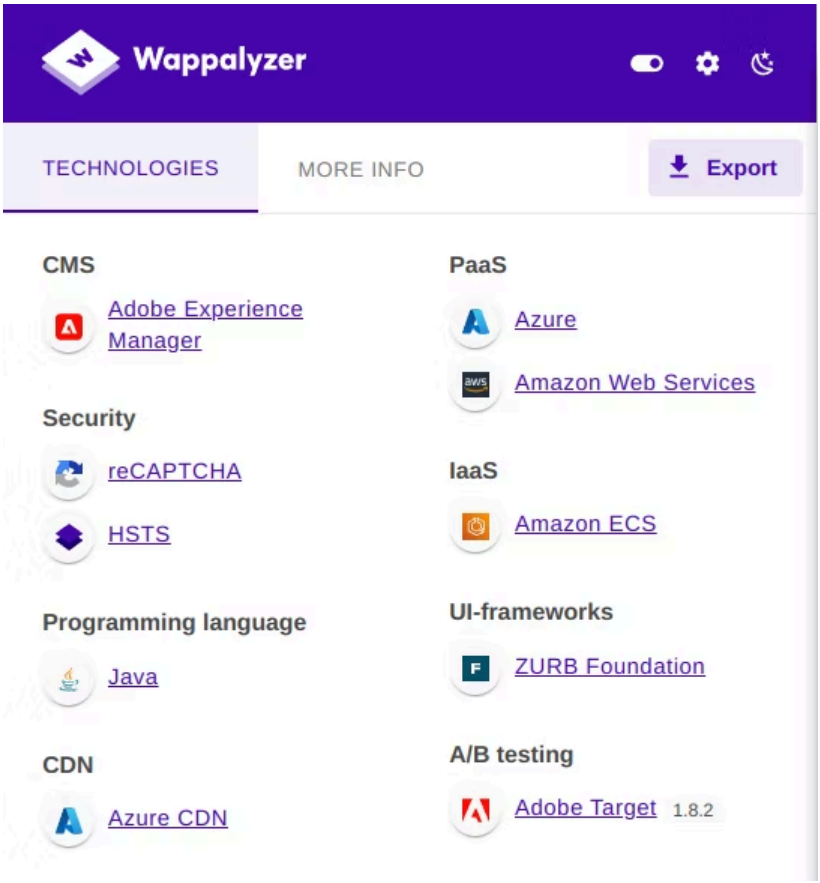
Eyewitness is an open-source tool and is available on Github:

https://github.com/redsiege/EyeWitness

Wappalyzer

Fingerprinting is an excellent way of figuring out what technologies or services your target is running on. Unfortunately, Eyewitness is only capable of performing basic fingerprinting of live hosts.

If you want to take a step further, you can use Wappalyzer (the browser extension) or HTTPX (which relies on the Wappalyzer package) for fingerprinting your list of targets. Understanding what technologies and services your target is relying on is the next step in recon as it will help you tremendously later on when you're actively searching for vulnerabilities.



Wappalyzer extension for technology fingerprinting

GAU (GetAllUrls)

Content discovery plays an important role in recon as we've covered in one of our most recent articles. Bug bounty hunters who perform content discovery are usually rewarded well as they come across untouched and untested features, functionalities and endpoints more often. And this can drastically increase your chances of finding new security vulnerabilities.

GAU (short for GetAllUrls) is an open-source tool that can help you quickly fetch URLs, links and other indexed files from the WaybackMachine (Internet Archive) and other archiving and indexing engines!

It's also easy to use, to return links of a list of targets, you can use the command below:

```
cat targets.txt | gau
```

GAU (GetAllUrls) demo

GAU (GetAllUrls) is Golang tool and its installation instructions are available on Github:

https://github.com/lc/gau

Ffuf

Forced browsing (bruteforcing) is another method to reveal hidden API endpoints, admin panels or other files that were previously not referenced anywhere.

Ffuf is a powerful tool that is capable of bruteforcing almost anything over HTTP, from normal content discovery (such as links and files) to vhosts to parameters at a super fast rate!

Here's the command to perform bruteforcing on your target:

```
ffuf -u https://example.com -w /path/to/wordlist.txt
```

Ffuf demo

You can find the installation and usage guide on Github:

https://github.com/ffuf/ffuf

Additionally, we've also created an article about Ffuf that goes into detail about its capabilities that you can read by clicking the link below:

https://blog.intigriti.com/hacking-tools/hacker-tools-fuff-fuzz-faster-u-fool

Arjun

Query and body parameters are where most user input is accepted and also where most vulnerabilities arise. Parameter bruteforcing is a reliable way to discover accepted input parameters and trigger unintentional behaviors that often lead to security vulnerabilities.

Arjun is a powerful tool that supports parameter bruteforcing in several different HTTP request bodies and content types:

```
arjun -u https://example.com/example.php
```

Arjun parameter discovery tool demo

A detailed installation guide is available on Arjun's Github repository:

https://github.com/s0md3v/Arjun

TIP! If you'd like to perform parameter bruteforcing with Burpsuite, check out the ParamMiner extension! It's capable of finding hidden parameters and this extension is often used to reveal web cache poisoning vulnerabilities!

LinkFinder

Javascript files are a goldmine for bug bounty hunters, and this is often because they contain several references to links and endpoints that could previously not be found through other content discovery methods.

LinkFinder is a simple tool that performs an exceptional job of finding new links, URLs and other referenced files and endpoints in javascript code.

Linkfinder is also easy to use, here's how to quickly analyze a javascript file:

```
python3 linkfinder.py -i https://example.com/app.js
```

https://github.com/GerbenJavado/LinkFinder

Conclusion

We all know that recon is an important phase in bug bounty, this article referenced 8 of the most essential tools that we think can help you perform better reconnaissance and increase your chances of coming across a security vulnerability.

Now it's time to put your skills (and new toolbox) to the test! Intigriti hosts bug bounty programs of some of the biggest companies and organizations in the world, often including wide scopes and hundreds of assets for you to test. Browse through our list of programs and who knows, maybe your next bounty will be rewarded on our platform!

Get Started Today