Anyone who came in the field of cybersecurity wants to earn money. Either he can do some job or he can work as a freelancer. There are endless opportunities for a freelancer in the field of ethical hacking. The most famous opportunity is known as **Bug Bounty Hunting**.

Bug Bounty Hunting is nothing but simply visiting the targeted website and start finding bugs/vulnerabilities in the defined scope of that Bug Bounty Program.

# BEFORE STARTING WITH BUG BOUNTY

I am believing that you are new in the Ethical Hacking field. So, I am mentioning almost every possible step you should follow to start hunting. If you are already experienced in the field of Bug Bounty, then you may leave it because this article is quite long and I won't feel good to waste someone's time. However, you are most welcome to judge me but please don't be harsh with your wordings 

## FINDING WEBSITE

You just can't start testing with a random website. It'll be termed as an illegal offense. I saw many beginners who got knowledge of some automated tools such as sqlmap, they start testing with random sites. But this is wrong and illegal.

You can only test the websites which have a proper Bug Bounty Program or Responsible Disclosure. Don't worry, I'll provide you the techniques to find such websites.

Steps To Find Bug Bounty Programs:

1- **GOOGLE DORKS:**

Google Dorks is one of the most common and efficient ways to find out the Bug Bounty Programs. I used the dorks such as i*ntitle, inurl* etc to find some unlisted bug bounty programs. The main advantage of using these dorks is that you'll end up by finding some uncommon programs which are less popular. Some famous dorks are:

site:.in responsible disclosure

site:.in bug bounty

Here you can replace the .in part to any other domain such as .de or .com

Less Popular Programs = High Chance of Getting Bugs

Please check out this video. It’s the explanation of Google Dorks.

2- **FACEBOOK**:

We can’t deny the fact that we all love to post our achievements on social media. Therefore, social media such as facebook is the hub of such posts. To get the bug bounty programs on facebook we can use the method of hashtags. Below is the example:

#bug_bounty #HallOfFame #hacking etc

Simply search any of the above hashtags and you’ll see a large number of posts with program names. Simply start with that program.

3- **BUG BOUNTY PLATFORMS**

There are many bug bounty platforms designed for hackers. Hackers register with those platforms and start getting programs immediately. I will define those platforms here with the level of difficulty.

**BUGCROWD**

Bugcrowd platform is one of the most famous platforms. As a beginner, I would suggest you to directly visit and register on it. See, we all need motivation. The best part of bugcrowd is that it’ll give you at least Hall Of Fame if the vulnerability goes duplicate. So this will keep you motivated so that further, you can perform well.

**HACKERONE**:

Hackerone is another good platform to test your hacking skills. However, this has a bit high difficulty level. Here you'll not get bugs very easily. You need to practice a lot before starting with this platform.

**OTHER PLATFORMS:**

There are many other platforms such as cobalt, zerocopter, synack, antihack etc. You can register on them too. They have less difficulty level because not so many hackers are registered on it as of now. So, due to the low saturation level, they are very much demanded.

**MY PERSONAL ADVICE:**

Every Platform has 2 types of programs. Paid and non-paid(Points oriented). I strongly advise you to go with non-paid programs in the beginning because:

-> Higher chances of getting a valid bug.

-> It'll boost up your self-confidence (most important)

-> It'll polish up your hunting skills.

-> Points will improve your ranking on the platform.

Once you get 8-10 valid reports, now you can move to bounty programs.

# AUTOMATED SCANNERS:

Automated Scanners are tools designed to find vulnerabilities in the targetted application. Some well known automated scanners are Acunetix, Nikto, Nessus, etc.

They are some famous ones. Some programmers made automated scripts to perform the scanning part.

## WHY NOT USE AUTOMATED SCANNERS:


It is highly recommended from my side that you should avoid using automated scanners on Bug Bounty Programs.

Before understanding why shouldn't, you should know how do they work.

Most of the automated scanners work on the active scanning methodology.

In active scanning, multiple crafted requests is sent to the webserver with new payload each time. Suppose if there are 50,000 payloads, then 50k requests will be entertained by the server in a shorter duration of time.

This might result in a DOS attack. Because by those scanners, usually, the high volume of traffic is sent to the server. This is not normal behavior.

This might crash the web application or it may lead to blacklist your IP address by that program.

See, automated scanners don't have their own intelligence. Usually, they can't detect filters. They have a bunch of payloads which are executed against the application to achieve the desired results.

I hope now you understand the working mechanism. Let's dig deeper to know why we should not use those scanners for our targetted application.

Automated scanners deliver lots of false-positive results. In case if you don’t know false positive, I mean wrong results. So scanners will consume your precious time and deliver you wrong results. Even if you use scanners, please verify the vulnerability manually. Otherwise, you will lose your points. And if you don’t know about that vulnerability, please skip it, learn first and then report.

Another demerit of using an automated scanner is that your IP might get blocked by the program. Then you won’t be able to report any vulnerability again. In fact, in most of the programs, you will find scanners prohibited in their polices.

Now I guess above 2 reasons are sufficient to know why not to use automated scanners.

# APPROACHING THE TARGET

You finally got the target. Now the next step is to start hunting. Below are the steps you should follow:

## BACKEND FRAMEWORK DETECTION

Backend detection means to detect the technology used by the website. It can be some CMS such as WordPress, Joomla or maybe custom-designed application based on PHP, ASP, etc. Backend detection can be done with a chrome plugin Wappalyzer. The amount of information enumerated by this plugin can differ from site to site.

Once you detected the backend successfully, try finding the exploit for the same. If you are lucky and the version is outdated, then you’ll find the exploit easily. To find the exploit simply google it. Some exploits need admin permissions. So it is a way to hard to execute them. If you are knowledgable enough, then try chaining the multiple vulnerabilities together. It might give you perfect results.

You can also detect framework from some online services such as Cmsdetect etc, but I didn’t find any point to dig deeper as Wappalyzer is convenient to do the job. However, once you detect CMS, you can use those online services to enumerate plugins along with their versions. So this will give you more benefit and a broader scope.

## ENUMERATING SUBDOMAINS

Subdomains are the hidden treasure. Never start testing with ~~example.com~~ always test sub.example.com. So we need to find subdomain right. Here I'll tell you 1 single and most effective website to enumerate the subdomains.

https://www.virustotal.com/gui/domain/target.com/details

Virustotal is a Google-owned application. And in my bug bounty career, I used it very much. I also used some scripts too but I didn't like them very much. For me, it worked very well. Simply visit the above URL by replacing target.com to any other website. Click on relations and you will get the list of all subdomains.

Guys it'll be hard to believe but I got $100 for doing nothing  yes seriously. If you also want to earn, follow the steps.

Open subdomains one by one in the new tab. Now watch the response. Some domains will deliver the content, some will give errors.

Now here comes the best part. We need to check those errors. if it's 503 or 404 simply leave it. But in some cases, you'll get framework based errors such as full path disclosure, server version disclosure, etc. Simply report it. In 90% of cases, such bugs are accepted(However, considered as P3 or P4 without exploitation). So you don't need to do anything. By this way, I got 100$ 

## FINDING OTHER VULNERABILITIES

Now, you can search for many other vulnerabilities such as Cross-Site Scripting, SQL injection, Session related issues, etc. Basically, what you need to do is playing with parameters by manipulating their values and observe the response.

Try brute-forcing (if the bug is in scope) sensitive fields, inserting XSS payloads in the profile section, who knows if you can get persistent XSS. There are endless opportunities. Once you'll understand how a website is working then it'll be very easy for you to find

vulnerabilities.

Check if the validation is server-side or client-side. If it is client-side, try bypassing the sensitive fields. Try bypassing XSS filters there using some proxy. Again saying, there are endless opportunities.

Actually, each and every topic deserves a dedicated article. So, it's not possible to explain everything in this single article. I am just giving an overview. After this, I'll write topic by topic from the very basics. So, keep reading.

If you are a true beginner, I know it's a bit hard for you to understand some words I used in this section. But, I did this with an intention. I want you all to visit Google or youtube and search for:

1- Client Side Validation vs Server Side Validation
2- Cross-Site Scripting
3- Bruteforce
4- Proxy – search for Burp Suite too
5- Parameters – in terms of web application

If you come across any issue, you can anytime ask me on my official accounts @officialpranj.

Some rules I suggest you follow while doing your research:

Don't flip your target too much. Be stable. When I started, I give almost 8 hours to a single target, then only I leave for another. See, in starting you will do the enumeration. This will consume most of your time.

Moreover, you aren't an expert so you need to search for different problems you are getting during the testing period. So approx 8 hrs is perfect time duration.

The second main thing is to read the program scope, terms and polices properly. Otherwise, you may end up by spending time in an invalid domain or non-applicable vulnerabilities.

Don’t try uploading shells in Google or Facebook  Start with P3 vulnerabilities. They won’t pay you much but there is a high chance of getting those issues.

If you are testing for the first time, try with some small websites that offer Hall Of Fames or gifts. There is comparatively less competition so high chances of getting valid issues.

Don’t be sad if the vulnerability goes duplicate. It’s a part of Bug Bounty. I saw people getting $1k RCE as duplicate. We can’t do anything in that. So, be relaxed and search for another issue.

# WRITING A REPORT:

Writing a Bug Bounty report is the most crucial part of the whole process. It’s a post step of finding a valid Bug. This report will decide your bounty amount. Before writing, keep the below points in mind:

## DIFFERENT PARTS OF A BUG BOUNTY REPORT:

Following are the different sections of a bug bounty report:

1- Subject (Include Bug-type)
2- Vulnerability definition (not more than 2 lines)
3- Affected parameters(BOLD)
4- How a hacker can exploit it(Explain this part as much as possible by mentioning all the impacts)
5- Proof Of Concept (In the form of a video or a writeup) Make sure to keep it private.
6- Some resources for vulnerability.
7- Your Name and social handles (If you like. Not Mandatory)

So, the above 7 points are enough to consider before writing a valid report. I am going to show you an example below.

Dear team,

I would like to report a Stored XSS issue in your web application.

With the help of this vulnerability, an attacker can even deface a particular webpage or even take over the victim's account of your website and can use it for malicious purpose.

I found this vulnerability under in the name parameter in the profile section.

Please consider the attached video file to reproduce the vulnerability.

Thanks,
Your Name

The length of a report can vary from bug to bug. Stored XSS is a critical issue so, I don't need to explain very much in that. Bug if you are chaining different vulnerabilities then I strongly recommend to write step by step procedure and include a video POC.

# FINAL WORDS:

It's not easy to mention each and everything in a 2k words article. I tried my level best to mention at least important things. It can be considered as part 1. I believe I'll dive into the more technical aspects in the future. In the field of bug bounty, you can only succeed only if you are devoting time and practicing a lot.

Everything depends on your practice. Practice makes a man perfect and the same applies here. So keep practicing and keep reading. I forgot to tell you that Hackerone is a great source of learning new vulnerabilities.

Just visit the website and click on Hactivity. There you'll find the latest vulnerability disclosures. Learn from them. For me personally, nothing can be a better place to learn then Hackerone. If you face any kind of issue, just leave a message. I'll try my best to help you out from that problem.

So stay lucky, happy hacking □

Pranjal Singhal

I am a freelancer Cybersecurity researcher and a digital marketer.  I have already helped Top IT Giants to secure their web applications and maintain a safe environment for their users. Sharing and Caring is my motive. I love to guide beginners about making a successful career in the cybersecurity industry.