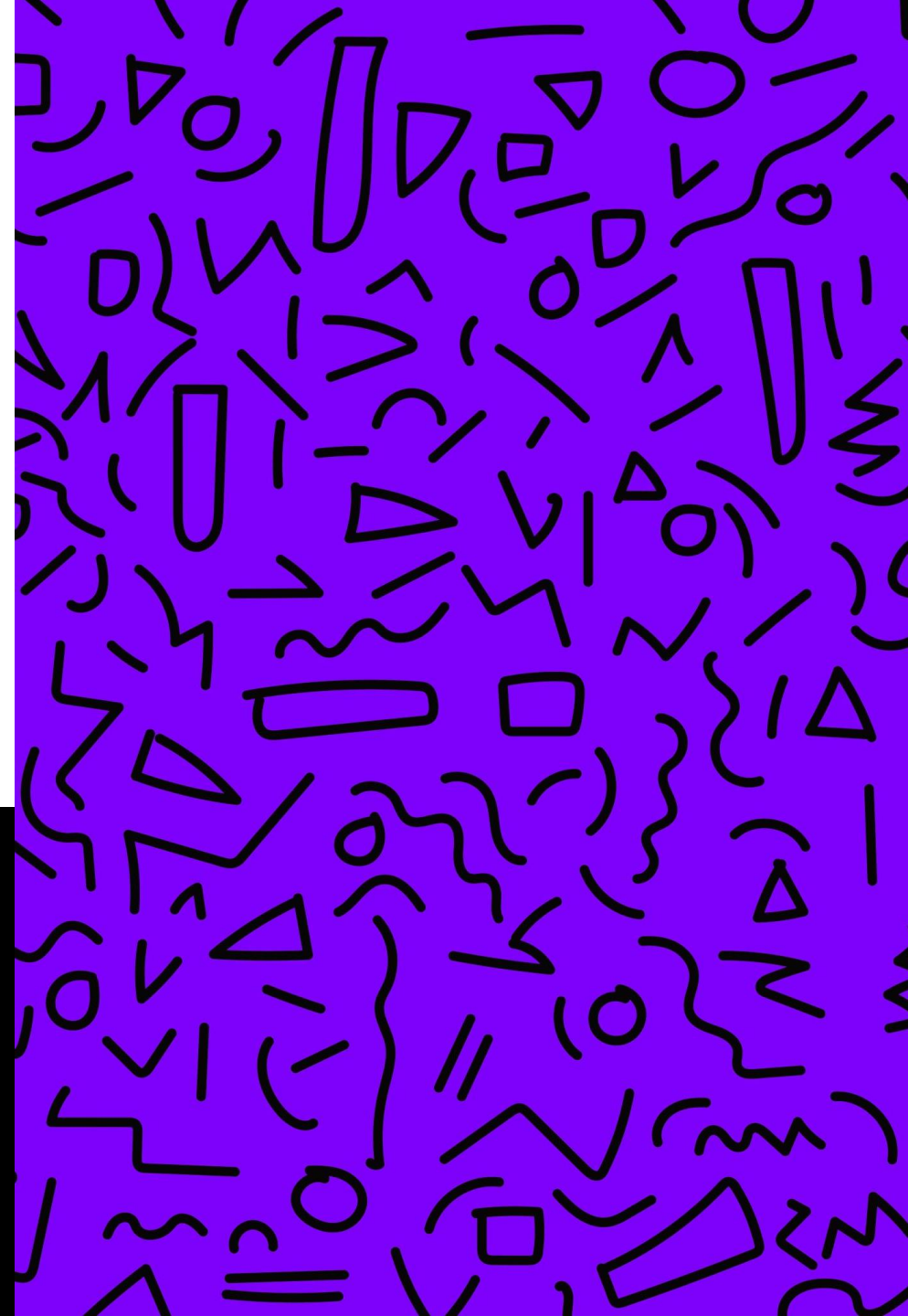


BUG BOUNTY, HOW TO START?

joas antonio



ABOUT ME

joas antonio

Information Security Analyst Red team move

20 years / Asperger

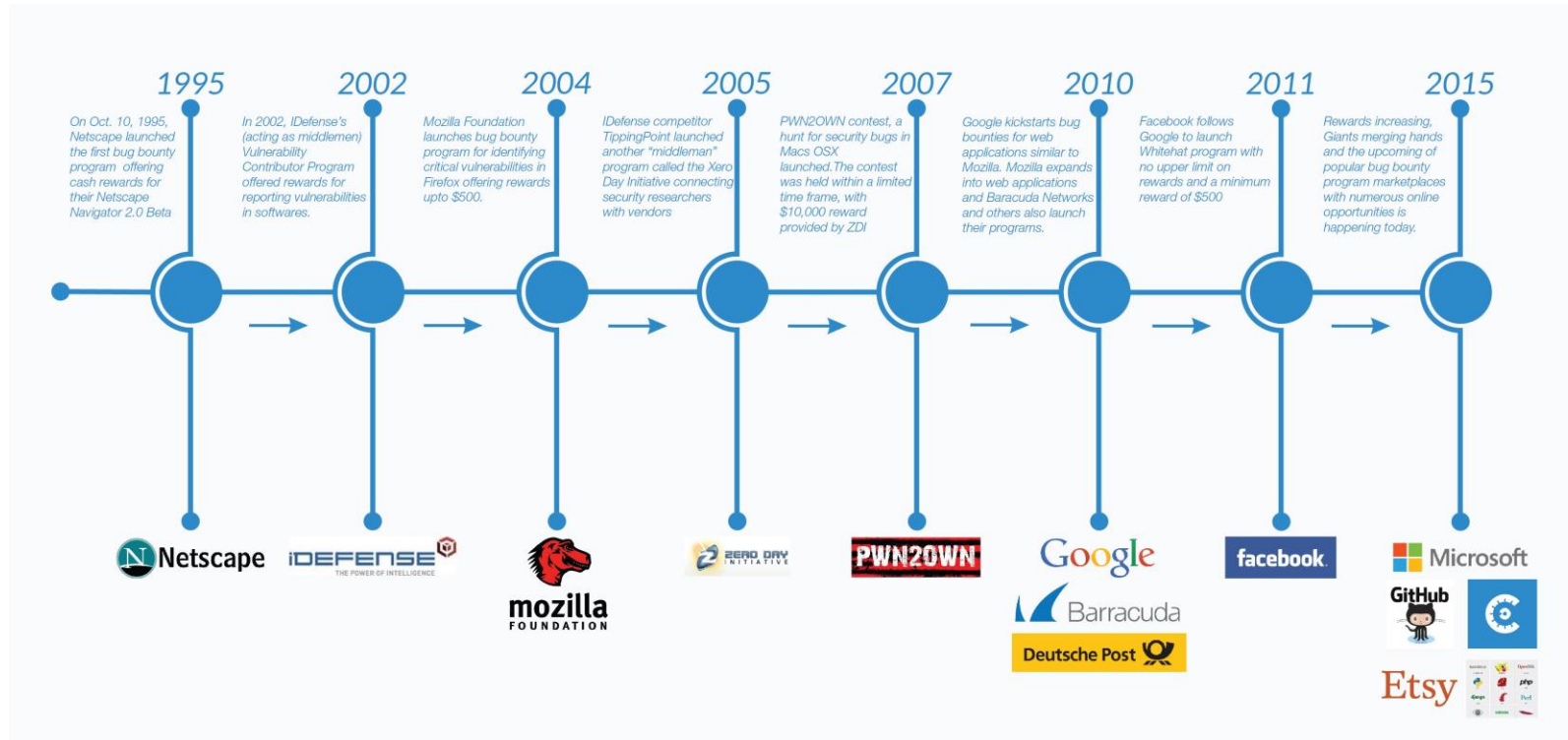
OWASP Project Leader

hacking is not Crime advocate

CEH Master, eWPT,eWPTX,eCPPT,eJPT,eMAPT and OSWP

ABOUT BUG BOUNTY

BUG BOUNTY AND YOUR STORY



PUBLIC VS PRIVATE

Private:

Private programs are programs that are not published to the public. This means that hackers can only see these programs when they receive specific invitations to break into them.

Public:

When programs go public, they open up to reporting submissions from across the hacking community. Making your bug bounty program public is completely optional.

VDP VS. BBP

VDP:

A vulnerability disclosure program only provides clear guidelines on how an organization would like to be notified of potential security vulnerabilities found by external third parties. It is intended to provide instructions to discoverers on how and where to report a vulnerability so that the appropriate team can resolve them.

You VDPs are often called the “see something, say something” of the internet. It is a best practice to have a public-facing vulnerability disclosure policy, as it encourages others to report security risks they notice or encounter.

VDP VS. BBP

BBP:

A bug bounty program encourages external third parties to find security vulnerabilities in a company's software and report them directly to the company so they can be safely addressed. In return, discoverers of the vulnerabilities are rewarded with monetary prizes.

You BBP have the option to be private or public, where you can choose which will work best for you. BBPs are also a little more complex than the VDPs, as there are many more components and settings to configure, such as a reward structure and response targets.

BUG PLATFORMS BOUNTY

1. HackerOne
- two. bugcrowd
3. Intigriti
4. Bug Hunt
5. hackaflag
6. Yogosha
7. zeroday initiative
8. Open Bug bounty
9. YesWeHack
10. Cobalt.io
11. Synack Red team
12. Hacker Security

BUG BOUNTY MILLIONAIRES



TIPS AND TRICKS - RECON

RECONTRICKS

subdomains

- curl + parallel one-liner
- find subdomains with SecurityTrailsAPI
- find related domains via favicon hash
- find subdomains using RapidDNS
- THErecon type to find more subdomains (Shodan)
- find subdomains using ASNs with amass

RECONTRICKS

ContentDiscovery

- Accesshidden sign-up pages
- find hidden pages on Drupal
- findSpring Boot serverswith Shodan
- Forgotten data base dumps
- find RocketMQconsoleswith Shodan
- fuzz listfor GITandSVN files
- Generate content discovery wordlist fromURI
- HTTPrecon automation with httpx
- web serversonnon-standardports(Shodan)
- keeptrackof attacksurfacewith amass
- easy information disclosure with httpx
- find Kubernetes with Shodan
- OneListForAll- "rockyou"wordlistfor webfuzzing
- list of24 Googledorksfor bugbounties

<https://www.infosecmatter.com/bug-bounty-tips/>

RECONTRICKS

sensitive Information

- top 5 bugbountyGoogledorks
- find sensitive information with gf
- find sensitive information with AlienVaultOTX
- find data base secretsin SVNrepository
- GitHubdorksforfinding sensitive information
- sensitivedateleakage using.json
- easy wins with Shodan dorks
- find accesstokenswith ffff and gau
- GitHubdorksforfinding secrets
- Use Google Cacheto find sensitivedate
- phpinfo()with sensitive information
- recon leading to exposeddebugendpoints
- list of14 Googledorksforrecon and easy wins
- GitHubdorksfor AWS,Jira,ok..secrets
- list of9 tools foridentifying sensitive information

<https://www.infosecmatter.com/bug-bounty-tips/>

KINGOFBUGBOUNTY AND EXPLAIN SHELL

<https://github.com/KingOfBugbounty/KingOfBugBountyTips>

<https://explainshell.com/>

TIPS AND TRICKS - VULNS

VULNERABILITIES 2022

- XSS (stored and reflected)
- Log4j :0
- Host Header injection :)
- account takeover :\
 - 2FA bypass :(
- SSRF :|
- CRLF injection ;(
- CSRF :D
- Business logic vulnerability
- Remote code execution in Cloud Services (AWS, GCP, Huawei, Oracle and Azure)
- Information Disclosure
- cache poisoning
- Open redirect
- File Upload vulnerability
- SQL injection
- exploit OAuth
- API exposure
- From and DDoS
- clickjacking
- prototype Pollution
- IDOR
- Github exposure
- subdomain takeover
- Email spoofing

VULNS TRICKS



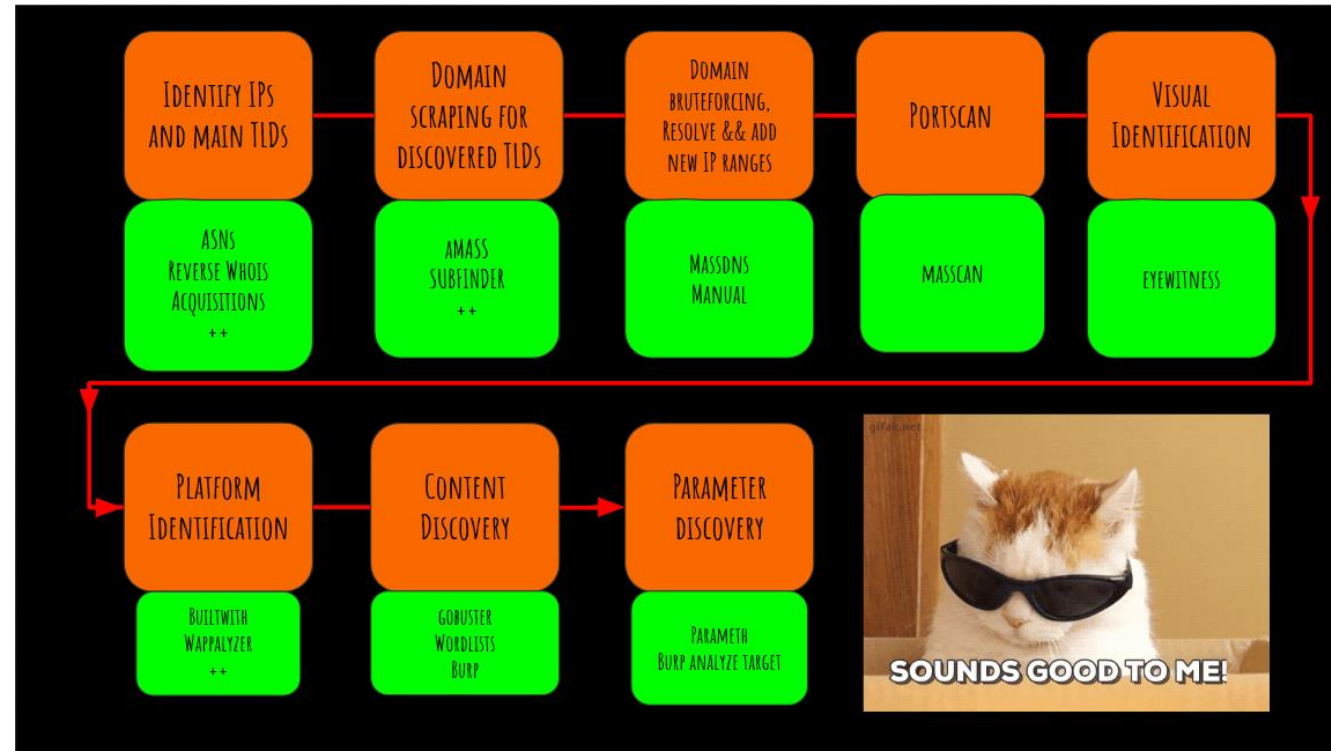
VULNS TRICKS

[https://book.hacktricks.xyz/
pentesting-web/web-
vulnerabilities-methodology](https://book.hacktricks.xyz/pentesting-web/web-vulnerabilities-methodology)

[https://book.hacktricks.xyz/
pentesting/pentesting-web](https://book.hacktricks.xyz/pentesting/pentesting-web)



BUG BOUNTY METHODOLOGY



WSTG VS MSTG

[https://github.com/OWASP/
wstg](https://github.com/OWASP/wstg)

OWASP Top 10 2017		change	OWASP Top 10 2021 proposal	
A1	Injection	as is	A1	Injection
A2	Broken Authentication	as is	A2	Broken Authentication
A3	Sensitive Data Exposure	down 1	A3	Cross-Site Scripting (XSS)
A4	XML eXternal Entities (XXE)	down 1 + A8	A4	Sensitive Data Exposure
A5	Broken Access Control	down 1	A5	Insecure Deserialization (merged with XXE)
A6	Security Misconfiguration	down 4	A6	Broken Access Control
A7	Cross-Site Scripting (XSS)	up 4	A7	Insufficient Logging & Monitoring
A8	Insecure Deserialization	up 3 + A4	A8	NEW: Server Side Request Forgery (SSRF)
A9	Known Vulnerabilities	as is	A9	Known Vulnerabilities
A10	Insufficient Logging & Monitoring	up 3	A10	Security Misconfiguration

WSTG VS MSTG

<https://github.com/OWASP/owasp-mstg>

OWASP Mobile Top 10 Risks

- M1 Insecure Data Storage
- M2 Weak Server Side Controls
- M3 Insufficient Transport Layer Protection
- M4 Client Side Injection
- M5 Poor Authorization and Authentication
- M6 Improper Session Handling
- M7 Security Decisions via Untrusted Inputs
- M8 Side Channel Data Leakage
- M9 Broken Cryptography
- M10 Sensitive Information Disclosure

Alpha Documentation

Mobile Security Project

- Top 10 Risks
- Top 10 Controls
- Threat Model
- Testing Guide
- Tools
- Secure Development

Plan, Launch & Learn: The Bug Bounty Roadmap

PLANNING FOR YOUR BUG BOUNTY PROGRAM



The first step in planning your bug bounty program is establishing program goals and [company objectives](#).



Setting the [scope](#) of your program clearly and thoughtfully is the most important part of launching a successful program.



Based on your security maturity, we help you set initial the [reward range](#) for your program.

Running a successful bug bounty program starts far before the actual program launch and is a continuous process.

This guide outlines the critical steps in planning, launching and learning from your bug bounty program with Bugcrowd's help.

PROGRAM LAUNCH



Prior to launching, it's crucial to [implement internal processes and align expectations](#) between departments.



Once your program has launched, you will start receiving bugs from [Bugcrowd researchers](#).



Bugcrowd's vulnerability disclosure platform, [Crowdcontrol](#), filters and de-duplicates all submissions.



Our team of experts validate deduplicated submissions and suggests priorities for each [valid vulnerability](#).

REPEAT



Your team then authorizes the [reward amount](#) and Crowdcontrol facilitates payment to researchers.



Our [integrations](#) make collaborating with your development teams to prioritize fixes seamless.



Your team receives valid, ready-to-fix [vulnerabilities](#) with necessary reproduction steps.



Crowdcontrol's [reporting](#) provides valuable insights throughout the lifetime of your program.



Bugcrowd's team of experts help you make adjustments to make sure your [program](#) delivers desired results.

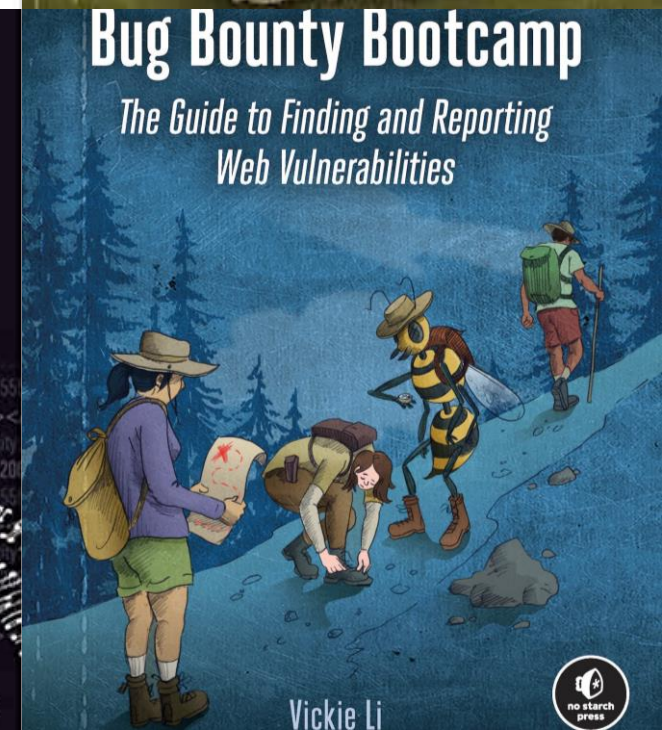
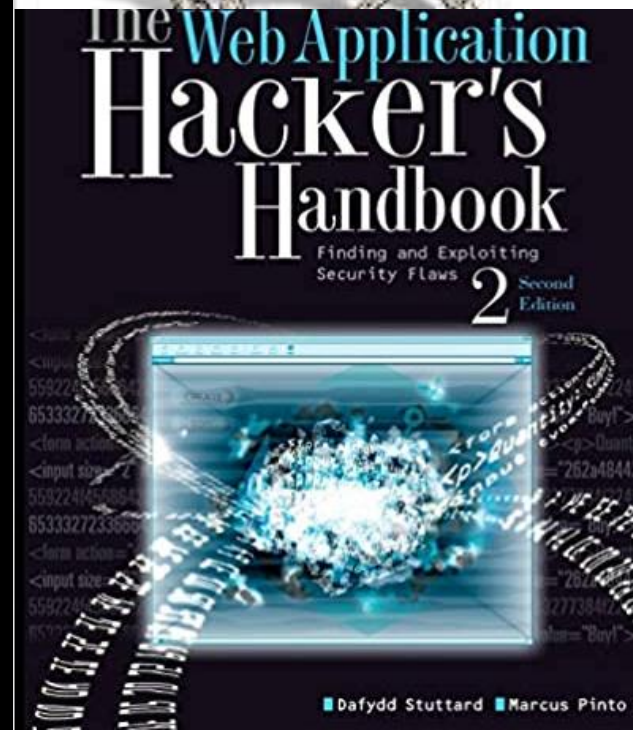
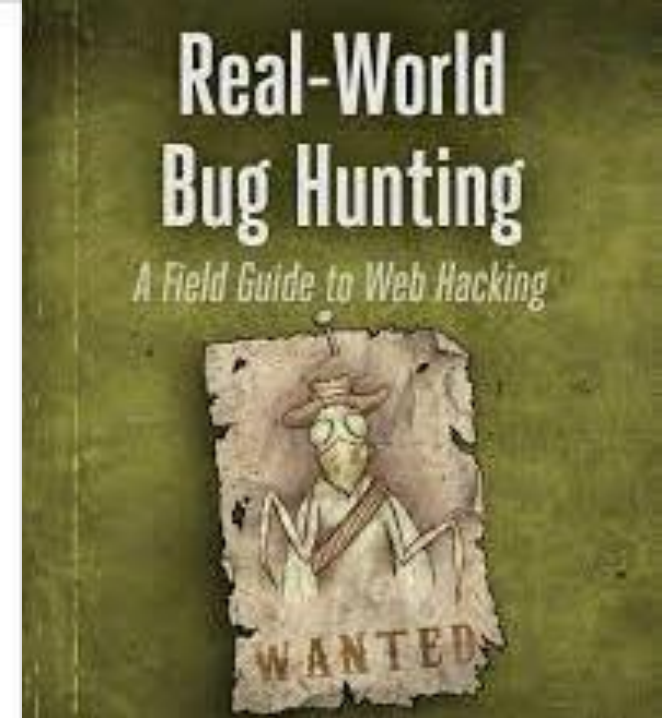
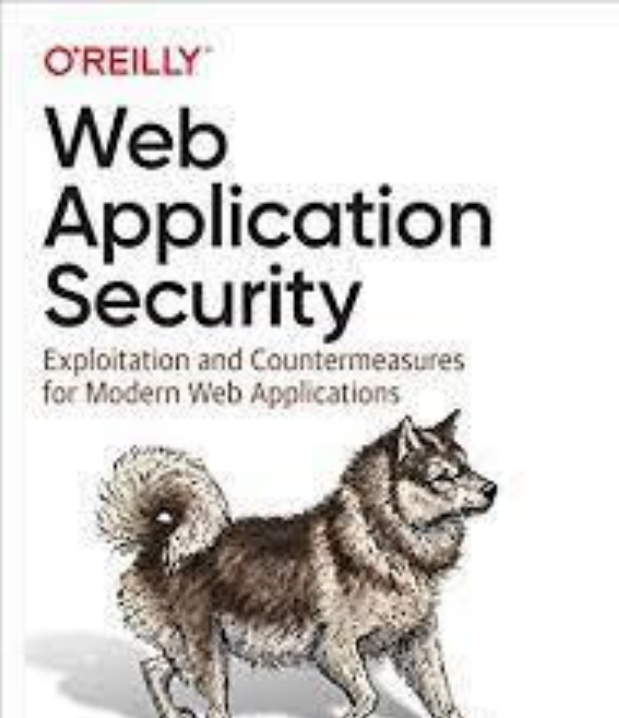
ITERATING AND IMPROVING YOUR PROGRAM

TIPS AND TRICKS - REPORT

TIPS TO CREATE A REPORT

- **Vulnerability Title:** This section should contain a clear and concise title that provides the reader with context about the vulnerability that a pentester it found.
- **Vulnerability Description:** This section should contain a high-level summary of the identified issue and an explanation of the impact it could have if successfully exploited. This is usually kept general and straightforward to give the reader context on the subject. You can talk here about how the issue works, leaving details about the client environment for different sections of the Vulnerability Report.
- **Affected Component:** This section typically contains a URL, Parameter, or other affected resource listed to provide more specific information about where the vulnerability exists.
- **Affected Users:** This section explains which application users could be affected if an attacker successfully exploited the issue.
- **Proof of Concept (Steps to Reproduce):** This section is critical as it contains the detailed steps needed to successfully reproduce an issue. Always make sure that the steps are detailed enough that anyone with little or no security knowledge can successfully reproduce the issue and understand the impact. To create more impactful steps, include appropriate screenshots and video proofs of concept as and when needed.
- **Criticality:** This is an important section because it tells the reader what the overall impact of the vulnerability could be if successfully exploited. Keep the impact description as realistic as possible, rather than writing what theoretically might happen. The best way to do this is to stick to an immediate consequence such as “an attacker could gain access to a user account” and not speculate what the attacker could do with that access (as it could do something unexpected). Criticality can be divided into two parts:
 - **Impact:** This section examines the effect of the discovery on technical and business operations. The OWASP Risk Rating Methodology describes this on a scale of Low to Very High.
 - **Likelihood:** This section explains the likelihood that the vulnerability will be exploited by a threat actor. The OWASP Risk Rating Methodology describes this on a scale of Low to Very High.
 - **Exploitation Complexity:** This is an optional section that describes how complicated it is to exploit a vulnerability and what requirements must be met for an attacker to successfully exploit the vulnerability.

BOOKS



FINAL TIPS

- **BASE IS ESSENTIAL (PROGRAMMING, NETWORKS AND OPERATING SYSTEMS)**
- **PRACTICE YOUR SKILLS IN WEB, MOBILE AND API ATTACKS**
- **STUDY OWASP GUIDES**
- **STUDY BUG BOUNTY WRITEUPS**
- **EXCHANGE IDEAS WITH THE COMMUNITY**

DOUBTS?

<https://www.linkedin.com/in/joas-antonio-dos-santos>