BBP summary for flow chart

1) Setup or append to Wiki notes.

- Methodology is for the process, like the flow chart
- Targets is for each program you have
  - Scope is for all the program's allowed and disallowed assets.
  - Domains is for all the subdomains (sub domains should have finger print info)
  - Notes section containing dates for pages, containing notes and scan outputs
  - Log is for when you run a tool, tell it in the log so you can reference the date later

Recon

1) Horizontal correlation, find all acquistions, CIDR ranges, domains.

- https://viewdns.info/reversewhois will find domains from domains.
- https://domaineye.com/ get domains from reversing mail servers
- https://domaineye.com/ get domains from reversing name servers
- https://domaineye.com/ get domains from CIDR ranges
- Google dork for websites
  - intext: + something on every site like the footer copyright.
- `amass intel -org <company name here>` to get a list of ASN numbers. False positives exist, check this.
- Lookup ASNs https://mxtoolbox.com/asn.aspx
- `whois -h whois.radb.net -- '-i origin <ASN Number Here>' | grep -Eo "([0-9.]+){4}/[0-99.]+){4}/[0-9]+"]+" | sort -u` use this to find CIDR ranges from ASN numbers
- `amass intel -asn <ASN numb>` will find domains from ASN numbers.
- `amass intel -cidr <CIDR range here>` to get domains from CIDR range
- `amass intel -whois -d <Domain Name Here>` will get domains from the whois method

2) Vertical correlation, get all the subdomains for each domain

- Use crt.sh to get subdomains (https://github.com/ghostlulzhacks/CertificateTransparencyLogs cli tool)
- Google dork to find subdomains, `site:domain.com`, `-site` can be used to exclude

- Get subdomains by checking the https://opendata.rapid7.com/sonar.fdns_v2/ dataset with the `zgrep ‘\.domain\ .com”,’ path_to _dataset.json.gz` command
- Use https://github.com/gwen001/github-search/blob/master/github-subdomains.py to find subdomains from github source code
- bruteforce dns subdomains to get subdomains (this doesnt send packets to domain) `gobuster dns -d domain.com -w wordlist.txt` (use good wordlist)
- `amass enum -passive -d domain.com` passivley get subdomains
- LAST STEP. find permutations with https://github.com/infosec-au/altdns

3) Discover content, endpoints, interesting files etc.

- you can crawl the site to find endpoints with `https://github.com/ghostlulzhacks/crawler/blob/master/crawler.py` . If the site is built with js and cant be crawled, use https://github.com/GerbenJavado/LinkFinder.
- Search for interesting files on wayback machine, (.bak, .zip, .config, /admin/, /api/) https://github.com/ghostlulzhacks/waybackMachine
- search for interesting keywords/params on waybackmachine, all different owasp types, tomnomnoms' gf.
- Search common crawl for the same things on the wayback machine https://github.com/ghostlulzhacks/commoncrawl
- Directory bruteforce for secrets, backup files, core dumps, configs, etc https://github.com/OJ/gobuster (with -k)
- Parse JS files for API keys, AWS creds, etc https://github.com/incogbyte/jsearch
- Dork to find things
  - exploit google dorks to find hidden assets, creds, vulnerable endpoints, etc https://www.exploit-db.com/google-hacking-database
  - use `ext:` to find PDFs, dbs, zip files, backups, configs, etc.
  - Search Third party vendors for credentials, internal links, docs, API keys, sensitive info, etc. `site:<3rd part> "company name"`
    - Codepad.co
    - scribd.com
    - npmjs.com
    - npm.runkit.com
    - coggle.it
    - papaly.com
    - trello.com
    - prezi.com
    - jsdelivr.net
    - codepen.io

    - pastebin.com
    - repl.it
    - gitter.im
    - butbucket.org
    - *.atlassian.net
    - inurl:gitlab "company name"

4) Fingerprint assets (IPs and Domains)

- Using Shodan get all assets from a CIDR range with `net:<"CIDR,CIDR,CIDR">`
- Using Shodan get all assets via org name `org:<"org name">`
- Using Shodan get all assets via SSL certs `ssl:<"Org name">`
- Repeat the last three with https://censys.io/ipv4
- Use Masscan to get desired ports (80, 443, 2375, 9200, 10250) https://github.com/robertdavidgraham/masscan. Grab banners for http ports
- Use Wappalyzer to get technology stack for each domain. Manually doing it will get more information, however you can use the command line version for less info https://github.com/vincd/wappylyzer
- Find out what firewall is used for each site https://github.com/EnableSecurity/wafw00f
- Find bypasses for firewalls https://github.com/0xInfection/Awesome-WAF#known-bypasses

Exploit

5) https://github.com/haccer/subjack run this tool to find subdomains that are vulnerable to takeover, this is will find false positives

- if you get a hit, `dig <hit>`, check the cname section, can you register the domain?

6) Github

- dork to find api keys, creds, ssh keys, password files, bash_history, log files. etc
    - `filename:.bash_history domainName`
    - https://github.com/techgaun/github-dorks/blob/master/github-dorks.txt
- Go to their company github page, collect all their employees, look for secrets in their repos

7) Search for misconfigured cloud storages

- For S3 Buckets, dork for it `site:.s3.amazonaws.com "target"` . Then brute force it https://github.com/ghostlulzhacks/s3brute
- For GCloud, brute force it https://github.com/RhinoSecurityLabs/GCPBucketBrute

- For Digital Ocean Spaces, dork for it `site:digitaloceanspaces.com <domain here>` and then brute force for it https://github.com/appsecco/spaces-finder
- For azure blob, it is impossible to brute, but can be dorked `site:core.windows.net <domain name>` , `site:"dev.azure.com" <domain name>`

8) Check for exposed ports

- port 9200 for elasticsearch DB API.
- Port 2375 for Docker API.
- Port 10250 for Kubernets API

9) use ffuf with all known http domains to look for `.git` or `.svn`

10) Exploit CMSes

- For WP, run wpscan. and check `/wp-content/uploads` for juice files
- For Joomla, run joomscan
- For Drupal, run droopescan
- For Adobe AEM, run aem-hacker
- For others, check exploit-db, google for exploits, and search for a specific cms scanner

11) Check for OWASP vulns

- Burp free doesnt let you ctrl f through history, so right click > save items (unencoded) > open it in code, search for `<?xml version` and if you get a hit, find the request in burp and try to XXE it.
- Test a WAF bypassing xss payload on the site, for each reflected, stored and dom xss
- Test XSS via SVG file, every image location possible.
- Search endpoints for ones with get parameters that are URLS, check for ssrf vulns
- On, change emails, change passwords, dangerous operations, check if there is a CSRF token, if not try an CSRF vuln.
- Throw `'` into search bars, and if you get any kind of error, test for an sqli
- If the app uses websockets, always check for CSWSH as you can likely hyjack the socket.

Text