Ch.19 Real-World Bug Hunting "methodology" (by Ryad)

Follow Me for more :)

- Github: https://github.com/KyrillosReyad
- Linkedin: https://www.linkedin.com/in/kyrillos-reyad-54120b19a/
- Twitter: https://twitter.com/kyrillos_reyad

1.RECONNAISSANCE

- Scope => Subdomins => IP addresses => Type of site (open source, Software as a service) => (Technologies, programming language, database type, frameworks)
- Start with the tools you can run in the background
- Be careful ! if Firewall bans you, you might be unable to visit common sites

1- Subdomain Enumeration

- finding subdomains using your VPS
- SubFinder tool, might not find all subdomains, so find some subdomains through brute-forcing
- SubFinder can brute-force subdomains using a common word list
- SecLists has lists of common subdomains
- crt.sh is a reference to check whether wildcard certificates have been registered
- also consider to enumerate subdomains of subdomains

2- Port Scanning

- The results of the port scan can also be indicative of a company's overall security a company with lots of open ports might have better potential for bounties
- Nmap is an older tool and can be slow, But it's great because you give it a list of URL and it will determine the IP address
- Masscan extremely fast and might be best when you have a list of IP addresses
- Different IP address might indicate a custom-built or third-party application that doesn't share the same level of security as the company's core applications

3- Screenshotting

- Gives you a visual overview of the program's scope
- Look for common error messages known to be associated with subdomain takeovers
- Look for sensitive content, watch out for administrative login pages, default installation pages,...etc
- Look for applications that don't match ones that are typical on other subdomains
- HTTPScreenShot use it with a list of IP addresses, Screenshotting, enumerate other subdomains associated with SSL certificates, cluster your results into groups
- Gowitness is a fast, use it when you have a list of URLs instead of IP addresses
- Aquatone

4- Content Discovery

- Attempt to discover files and directories by brute-forcing them, success depends on word list you use
- Gobuster give it a domain and word list
- Meg tool allows you to test multiple paths on many hosts simultaneously
- Burp Suite Pro (built-in content discovery tool or Burp Intruder)
- Exploit DB maintains a database of Google dorks
- Check the company's GitHub.
- Gitrob crawl GitHub repositories for application secrets and other sensitive information.
- review code repositories and find third-party libraries an application is relying on

5- Previous Bugs

- familiarize yourself with previous bugs (Hacker write-ups, disclosed reports, CVEs, published exploits,...etc)
- Just because code is updated doesn't mean all vulnerabilities have been fixed
- When a fix is deployed, it means new code was added, and that new code could contain bugs

2.TESTING THE APPLICATION

1- The Technology Stack

- Identify the technology being used (frameworks, third-party services, locally hosted files, remote files,...etc)
- Wappalyzer very handy for quickly fingerprinting technologies
- leave the default configuration for Burp Suite enabled and walk through the site to understand the functionality
- Some factors to keep in mind
  - Content formats a site expects or accepts
  - Third-party tools or services that are easily misconfigured
  - Encoded parameters and how an application handles them
  - Custom implemented authentication mechanisms, such as OAuth flows

2- Functionality Mapping

Looking for markers of vulnerabilities

- look for behavior commonly associated with vulnerabilities
- When I find something of interest, I stop and begin application testing

Define and work toward a goal

- decide what to do before testing the application (Focus on one vulnerability)
- for Ex: if you're looking for a remote code execution vulnerability, unsanitized HTML returned in a response body wouldn't be of interest.

Follow a checklist

- Both (OWASP and Web Application Hacker's Handbook) provide comprehensive checklists (that's a link for my summary about web app hacker's Handbook Methodology
- following a checklist can help you avoid missing vulnerabilities by forgetting to test

3- Finding Vulnerabilities

- After understanding of how an application works, you can start testing, Beginning by looking for behavior that could indicate a vulnerability
- Most programs don't permit automated scanners, it's unnecessarily noisy, and it requires no skill or knowledge, so You should focus on manual testing, start using the site a customer

- submit payloads wherever input is accepted and look for anomalies
- use the payload <s>000"'\);-//, which includes all the special characters
- Common vulnerabilities and areas to keep an eye out
  - CSRF: types of HTTP requests that change data, using and validating CSRF tokens checking the referrer or origin headers
  - IDOR: any ID parameters that can be manipulated
  - Application Logic: repeat requests across two separate user accounts
  - XXEs: Any XML-accepting HTTP requests
  - Information Disclosures: content that is guaranteed to be, or should be, kept private
  - Open Redirects: URLs that have a redirect-related parameter
  - CRLFs, XSS, and some open redirects: requests that echo URL parameters in the response
  - SQLi: If adding a single quote, bracket, or semicolon to a parameter changes a response
  - RCEs: file upload or image manipulation
  - Race Conditions: Delayed data processing or behaviors related to the time of use or time of check
  - SSRFs: Functionality that accepts URLs, such as webhooks or external integrations
  - Unpatched security bugs: Disclosed server information that can reveal outdated technology

==> look at the takeaway sections in each chapter of this book
- After digging flip back to your file and directory brute-forcing, determine other areas to focus on

- Methodology and techniques you use depend on the type of application you're testing
- "Approach every target like nobody's been there before. Don't find anything? Choose another one."
- hacking web applications isn't magic, it requires (knowledge, observation, and perseverance).
- recognizing where to look requires experience

A. TOOLS

- Also keep in mind that none of these tools should replace observation or intuitive thinking

WEB PROXIES

- Burp Suite (https://portswigger.net/burp)
- Charles (https://www.charlesproxy.com/)
- Fiddler (https://www.telerik.com/fiddler)
- Wireshark (more useful for monitor traffic)
- ZAP Proxy (https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project)

SUBDOMAIN ENUMERATION

- Amass (https://github.com/OWASP/Amass)
- crt.sh (https://crt.sh/)
- Knockpy (https://github.com/guelfoweb/knock/)
- SubFinder (https://github.com/subfinder/subfinder/)

DISCOVERY

- Enumerate files and directories, Doing so can help you find hidden functionality, sensitive files, credentials, and so on

- Gobuster (https://github.com/OJ/gobuster) is a tool you can use to brute-force URIs (directories and files) and DNS subdomains using wildcard support
- SecLists (https://github.com/danielmiessler/SecLists/) is a collection of word lists you can use while hacking, lists include usernames, passwords, URLs, fuzzing directories/files/subdomains, and so on
- Wfuzz (https://github.com/xmendez/wfuzz) allows you to inject any input in any field of an HTTP request

SCREENSHOTTING

- EyeWitness (https://github.com/FortyNorthSecurity/EyeWitness/) you can use it with other tools, like Nmap, to quickly enumerate hacking targets
- Gowitness (https://github.com/sensepost/gowitness/) uses Chrome Headless to generate screenshots, inspired by the EyeWitness tool
- HTTPScreenShot (https://github.com/anthian/httpscreenshot/) accepts IPs as a list of URLs to screenshot, brute-force subdomains, cluster results for easier review

PORT SCANNING

- Masscan (https://github.com/robertdavidgraham/masscan) the world's fastest internet port scanner, produces results similar to Nmap, only faster
- Nmap (https://nmap.org/) free and open source, determine (hosts, services, OS, type of packet filters or firewalls), includes scripts to build additional functionality ==> http-enum

RECONNAISSANCE

- BuiltWith (http://builtwith.com/) helps you fingerprint different technologies used on a target
- Censys (https://censys.io/) Maintains a database of how hosts and websites are configured, have a paid model
- Google Dorks (https://www.exploit-db.com/google-hackingdatabase/) find information not readily available when navigating a website manually
- Shodan (https://www.shodan.io/) search engine for IoT, helpful when discovering the target's infrastructure
- what CMS (http://www.whatcms.org/) you enter a URL and returns the (CMS) the site is most likely using

HACKING TOOLS

- Bucket Finder (https://digi.ninja/files/bucket_finder_1.1.tar.bz2) searches for readable buckets and lists all the files in them
- CyberChef (https://gchq.github.io/CyberChef/) Swiss army knife of encoding and decoding tools
- Gitrob (https://github.com/michenriksen/gitrob) helps you find potentially sensitive files that have been pushed to public repositories on GitHub, It presents its findings via a web interface
- online Hash Crack (https://www.onlinehashcrack.com) useful when identifying the type of hash a website uses.
- sqlmap (http://sqlmap.org/) automate the process of detecting and exploiting SQL injection vulnerabilities
- XSSHunter (https://xsshunter.com/) helps you find blind XSS vulnerabilities, When the XSS fires, it automatically collects information about where it occurred and sends you an email notification
- Ysoserial (https://github.com/frohoff/ysoserial/) a proof-ofconcept tool for generating payloads that exploit unsafe Java object deserialization.

MOBILE

- Being able to break down and analyze the apps' components will help you learn how they work and how they might be vulnerable
- dex2jar (https://sourceforge.net/projects/dex2jar/) converts dalvik executables (.dex files) to Java .jar files
- Hopper (https://www.hopperapp.com/) Reverse Engineering tool that lets you disassemble, decompile, and debug applications, It's useful for auditing iOS applications
- JD-GUI (https://github.com/java-decompiler/jd-gui/) exploring Android apps, displays Java sources from CLASS files

BROWSER PLUG-INS

- FoxyProxy: advanced proxy management, improves Firefox's built-in proxy capabilities
- User Agent Switcher: adds a menu and toolbar button, allows you to switch your user agent, use this feature to spoof your browser while performing some attacks
- Wappalyzer: helps you identify the technologies a site uses

BUG BOUNTY PLATFORMS

- Bounty Factory (https://bountyfactory.io/)
- Bugbounty JP (https://bugbounty.jp/)
- Bugcrowd (https://www.bugcrowd.com/)
- Cobalt (https://cobalt.io/)
- HackerOne (https://www.hackerone.com/)
- Intigriti (https://www.intigriti.com/)
- Synack (https://www.synack.com/)
- Zerocopter (https://www.zerocopter.com/)

3.GOING FURTHER

- save time is by automating your work
- Focus on testing the APIs the app interacts with, it's similar to web hacking
- To track new functionality on the websites you're testing, read the engineering blogs of the sites, monitor their engineering Twitter feeds, sign up for their newsletters,...etc
- You can also discover new site functionality by tracking JavaScript files
- pay for access to functionality, Ron Chan "Bug Hunter" paid a couple of thousand dollars to test an application and found a significant number of vulnerabilities that made the investment very worthwhile.
- more you know how a technology works, the more likely you are to find bugs, OAuth is a great example of deep diving into a technology that numerous websites use