Bug Bounty Checklist for Web App

> This checklist may help you to have a good methodology for bug bounty hunting
> When you have done a action, don't forget to check ;)
> Happy hunting !

Table of Contents

Recon on wildcard domain

- [ ] Run amass
- [ ] Run subfinder
- [ ] Run assetfinder
- [ ] Run dnsgen
- [ ] Run massdns
- [ ] Use httprobe
- [ ] Run aquatone (screenshot for alive host)

Single Domain

Scanning

- [ ] Nmap scan
- [ ] Burp crawler
- [ ] ffuf (directory and file fuzzing)
- [ ] hakrawler/gau/paramspider
- [ ] Linkfinder
- [ ] Url with Android application

Manual checking

- [ ] Shodan
- [ ] Censys
- [ ] Google dorks
- [ ] Pastebin
- [ ] Github
- [ ] OSINT

Information Gathering

- [ ] Manually explore the site
- [ ] Spider/crawl for missed or hidden content
- [ ] Check for files that expose content, such as robots.txt, sitemap.xml, .DS_Store
- [ ] Check the caches of major search engines for publicly accessible sites
- [ ] Check for differences in content based on User Agent (eg, Mobile sites, access as a Search engine Crawler)
- [ ] Perform Web Application Fingerprinting
- [ ] Identify technologies used
- [ ] Identify user roles
- [ ] Identify application entry points
- [ ] Identify client-side code
- [ ] Identify multiple versions/channels (e.g. web, mobile web, mobile app, web services)
- [ ] Identify co-hosted and related applications
- [ ] Identify all hostnames and ports
- [ ] Identify third-party hosted content

- [ ] Identify Debug parameters

Configuration Management

- [ ] Check for commonly used application and administrative URLs
- [ ] Check for old, backup and unreferenced files
- [ ] Check HTTP methods supported and Cross Site Tracing (XST)
- [ ] Test file extensions handling
- [ ] Test for security HTTP headers (e.g. CSP, X-Frame-Options, HSTS)
- [ ] Test for policies (e.g. Flash, Silverlight, robots)
- [ ] Test for non-production data in live environment, and vice-versa
- [ ] Check for sensitive data in client-side code (e.g. API keys, credentials)

Secure Transmission

- [ ] Check SSL Version, Algorithms, Key length
- [ ] Check for Digital Certificate Validity (Duration, Signature and CN)
- [ ] Check credentials only delivered over HTTPS
- [ ] Check that the login form is delivered over HTTPS
- [ ] Check session tokens only delivered over HTTPS
- [ ] Check if HTTP Strict Transport Security (HSTS) in use

Authentication

- [ ] Test for user enumeration
- [ ] Test for authentication bypass
- [ ] Test for bruteforce protection
- [ ] Test password quality rules
- [ ] Test remember me functionality
- [ ] Test for autocomplete on password forms/input
- [ ] Test password reset and/or recovery
- [ ] Test password change process
- [ ] Test CAPTCHA
- [ ] Test multi factor authentication
- [ ] Test for logout functionality presence
- [ ] Test for cache management on HTTP (eg Pragma, Expires, Max-age)
- [ ] Test for default logins

- [ ] Test for user-accessible authentication history
- [ ] Test for out-of channel notification of account lockouts and successful password changes
- [ ] Test for consistent authentication across applications with shared authentication schema / SSO

Session Management

- [ ] Establish how session management is handled in the application (eg, tokens in cookies, token in URL)
- [ ] Check session tokens for cookie flags (httpOnly and secure)
- [ ] Check session cookie scope (path and domain)
- [ ] Check session cookie duration (expires and max-age)
- [ ] Check session termination after a maximum lifetime
- [ ] Check session termination after relative timeout
- [ ] Check session termination after logout
- [ ] Test to see if users can have multiple simultaneous sessions
- [ ] Test session cookies for randomness
- [ ] Confirm that new session tokens are issued on login, role change and logout
- [ ] Test for consistent session management across applications with shared session management
- [ ] Test for session puzzling
- [ ] Test for CSRF and clickjacking

Authorization

- [ ] Test for path traversal
- [ ] Test for bypassing authorization schema
- [ ] Test for vertical Access control problems (a.k.a. Privilege Escalation)
- [ ] Test for horizontal Access control problems (between two users at the same privilege level)
- [ ] Test for missing authorization

Data Validation

- [ ] Test for Reflected Cross Site Scripting
- [ ] Test for Stored Cross Site Scripting
- [ ] Test for DOM based Cross Site Scripting
- [ ] Test for Cross Site Flashing
- [ ] Test for HTML Injection
- [ ] Test for SQL Injection

- [ ] Test for LDAP Injection
- [ ] Test for ORM Injection
- [ ] Test for XML Injection
- [ ] Test for XXE Injection
- [ ] Test for SSI Injection
- [ ] Test for XPath Injection
- [ ] Test for XQuery Injection
- [ ] Test for IMAP/SMTP Injection
- [ ] Test for Code Injection
- [ ] Test for Expression Language Injection
- [ ] Test for Command Injection
- [ ] Test for Overflow (Stack, Heap and Integer)
- [ ] Test for Format String
- [ ] Test for incubated vulnerabilities
- [ ] Test for HTTP Splitting/Smuggling
- [ ] Test for HTTP Verb Tampering
- [ ] Test for Open Redirection
- [ ] Test for Local File Inclusion
- [ ] Test for Remote File Inclusion
- [ ] Compare client-side and server-side validation rules
- [ ] Test for NoSQL injection
- [ ] Test for HTTP parameter pollution
- [ ] Test for auto-binding
- [ ] Test for Mass Assignment
- [ ] Test for NULL/Invalid Session Cookie

Denial of Service

- [ ] Test for anti-automation
- [ ] Test for account lockout
- [ ] Test for HTTP protocol DoS
- [ ] Test for SQL wildcard DoS

Business Logic

- [ ] Test for feature misuse

- [ ] Test for lack of non-repudiation
- [ ] Test for trust relationships
- [ ] Test for integrity of data
- [ ] Test segregation of duties

Cryptography

- [ ] Check if data which should be encrypted is not
- [ ] Check for wrong algorithms usage depending on context
- [ ] Check for weak algorithms usage
- [ ] Check for proper use of salting
- [ ] Check for randomness functions

Risky Functionality - File Uploads

- [ ] Test that acceptable file types are whitelisted
- [ ] Test that file size limits, upload frequency and total file counts are defined and are enforced
- [ ] Test that file contents match the defined file type
- [ ] Test that all file uploads have Anti-Virus scanning in-place.
- [ ] Test that unsafe filenames are sanitised
- [ ] Test that uploaded files are not directly accessible within the web root
- [ ] Test that uploaded files are not served on the same hostname/port
- [ ] Test that files and other media are integrated with the authentication and authorisation schemas

Risky Functionality - Card Payment

- [ ] Test for known vulnerabilities and configuration issues on Web Server and Web Application
- [ ] Test for default or guessable password
- [ ] Test for non-production data in live environment, and vice-versa
- [ ] Test for Injection vulnerabilities
- [ ] Test for Buffer Overflows
- [ ] Test for Insecure Cryptographic Storage
- [ ] Test for Insufficient Transport Layer Protection
- [ ] Test for Improper Error Handling
- [ ] Test for all vulnerabilities with a CVSS v2 score > 4.0
- [ ] Test for Authentication and Authorization issues
- [ ] Test for CSRF

HTML 5

- [ ] Test Web Messaging
- [ ] Test for Web Storage SQL injection
- [ ] Check CORS implementation
- [ ] Check Offline Web Application

Source:
OWASP