🔍 Bug Bounty Web Checklist

Track your web pentesting progress by checking each subcategory.

🛰️ Reconnaissance

1. Subdomain Enumeration (amass, subfinder, crt.sh)
2. Port Scanning (nmap, rustscan)
3. Directory Bruteforcing (ffuf, dirsearch)
4. Wayback Machine / Archive Recon
5. JS File Analysis (endpoints, keys, secrets)
6. Content Discovery (robots.txt, sitemap.xml)
7. Dorks (Google, Github, Shodan, Censys)
8. WHOIS & DNS Recon
9. DNS Zone Transfers
10. URLextractor

📢 Information Disclosure

1. Debug messages in responses
2. Leaked `.git/` directory
3. Leaked `.env` file
4. Stack trace on exception
5. Verbose error messages
6. Sensitive info in `robots.txt`
7. Exposed backup files (.bak, .old, .zip)
8. API keys in JavaScript files
9. Internal IPs in response headers
10. Credit card info in logs
11. Misconfigured GitHub repo (public leaks)
12. Environment variables in response body
13. Exposed Sentry / monitoring logs
14. Source maps exposed in production
15. Leaked memory dumps
16. User tokens in HTML comments
17. Full path disclosure
18. Version disclosure via headers
19. Sensitive data in Referer headers

20. Email/password pairs in export files

Authentication

1. Brute Force Login
2. 2FA Bypass
3. No rate limiting on login
4. Missing account lockout
5. User enumeration on login
6. Reusable password reset token
7. Reset link doesn't expire
8. Password reset sent to admin email
9. Weak password policy
10. 2FA bypass via fallback method
11. Session not invalidated after password change
12. Session fixation
13. Insecure "remember me" token
14. OAuth login without reauthentication
15. Open registration to admin role
16. Predictable password reset tokens
17. Public registration for internal application
18. Bypassing login with null/empty password
19. Fallback login method enabled (e.g., SSH)
20. Login allowed with unverified email
21. Reset token leaked in Referer header
22. Session ID exposed in URL

Authorization

1. IDOR (Insecure Direct Object Reference)
2. Accessing others' data via UUID guessing
3. Horizontal Privilege Escalation
4. Vertical Privilege Escalation
5. No access control on sensitive endpoints
6. Admin-only feature accessible by normal users
7. Misconfigured feature toggles
8. Changing roles via PUT/POST body
9. Accessing data by changing GraphQL ID
10. JWT with upgradable role claims

11. Tampering group ID to escalate privileges
12. Misconfigured middleware (e.g., no auth check)
13. Authorization missing in async jobs
14. Lack of validation in frontend
15. Bypass auth using mobile API endpoints
16. Auth enforced only via UI
17. Disclosure of access control matrix
18. Local file request bypasses proxy RBAC
19. Wildcard permissions misused
20. Access via soft-deleted accounts

💉 Input Validation

1. Cross Site Scripting (XSS)

1. Reflected XSS
2. Stored XSS
3. DOM-based XSS
4. XSS in file name
5. SVG upload with JavaScript
6. XSS in redirect URL
7. XSS in JSON response
8. XSS in markdown renderer
9. XSS in PDF export
10. XSS in 404 page
11. Payload in document.write
12. CSP bypass
13. Legacy browser XSS vector
14. Drag-and-drop XSS
15. XSS via input autofill
16. Mutation XSS in React
17. Unescaped template variables
18. XSS in <title> tag (tabnabbing)
19. XSS via email field
20. Nested JSON XSS

2. Injection Attacks

1. SQL Injection (classic)
2. Blind SQLi (timing-based)
3. Second-order SQLi
4. NoSQL Injection (MongoDB)

5. Command Injection
6. LDAP Injection
7. SSTI (Server-Side Template Injection)
8. XXE (XML External Entity)
9. CRLF Injection
10. Log Injection
11. Regex DoS
12. Host Header Injection
13. Code Injection in sandbox
14. XPath Injection
15. GraphQL Injection
16. OS-level injection via file parser
17. Deserialization attacks
18. PHP object injection
19. YAML deserialization
20. Dynamic language eval injection

3. Command Injection Types

1. Classic Command Injection
2. Blind Command Injection
3. Time-based Command Injection
4. Reverse Shell Injection
5. Blind Reverse Shell Injection
6. File Injection via Command
7. OS Command Injection via Parameter
8. Command Injection via Environment Variables
9. Injection via Shell Metacharacters (e.g., `;`, `&&`)
10. Injection via Pipes and Redirects (e.g., `|`, `>`)
11. Injection via Backticks (`` ` ``)
12. Injection via $( ) command substitution
13. Blind Time-Delay Command Injection

💻 Client-Side

1. Cross-Site Request Forgery (CSRF)
    1. CSRF on payment
    2. CSRF on settings change
    3. Logout CSRF
    4. CSRF on password change
    5. No CSRF token in form
    6. Misconfigured SameSite attribute

    7. Referer leakage causes CSRF
    8. CSRF via mobile endpoints
    9. CSRF on 2FA toggle
    10. CSRF + XSS combo
    11. CORS misused as CSRF protection
    12. Content-type based CSRF
    13. JSON CSRF
    14. No CSRF on multipart upload
    15. Preflight bypass via GET
    16. CSRF on profile picture upload
    17. CSRF in legacy iframe
    18. Bypass via null origin
    19. DNS rebinding to trigger CSRF
2. Clickjacking
    1. Clickjacking login iframe

🗂 File Handling

1. Unrestricted File Upload
    1. Uploading executable file
    2. Double extension bypass
    3. Bypassing MIME type check
    4. File overwrite
    5. Uploading with SSRF vector
    6. LFI via filename
    7. XSS via uploaded filename
    8. Upload to web root
    9. Misconfigured CDN cache
    10. Polyglot file upload
    11. Image with malicious EXIF
    12. RAR/ZIP bombs
    13. Uploading large file to cause DoS
    14. Upload with local file path in name
    15. SVG with embedded JS
    16. Backup file upload
    17. Ghostscript RCE via uploaded file
    18. File upload directory traversal
    19. Insecure PDF parsing
    20. Upload bypass via nested multipart
2. Path Traversal

📊 Business Logic

1. Coupon/Reward Abuse
2. Rate Limiting Issues
    1. Buying product with negative price
    2. Skipping payment step
    3. Infinite coupon redemption
    4. Price manipulation in cart
    5. Loyalty points fraud
    6. Refer-a-friend abuse
    7. Duplicate request = multiple rewards
    8. Logic flaw in rate limit
    9. Changing plan without paying
    10. Gifting subscription bypass
    11. Inventory bypass
    12. Uploading same receipt multiple times
    13. Applying expired discount
    14. Refund logic abuse
    15. Abuse of trial periods
    16. Bonus triggered without conditions met
    17. OAuth token reuse
    18. Redeeming coupons on others' accounts
    19. Buying restricted item as guest
    20. Flawed voting/rating logic

🧪 Miscellaneous

1. Open Redirect
2. Server Side Request Forgery (SSRF)
    1. Subdomain takeover
    2. Blind SSRF
    3. DNS rebinding
    4. Prototype pollution
    5. WebSocket hijacking
    6. JWT None algorithm
    7. JWT unsigned tokens accepted
    8. Path traversal (../etc/passwd)
    9. Unrestricted internal redirection
    10. Broken CAPTCHA bypass
    11. Cache poisoning

12. Host header injection (reset link)
13. Misconfigured cronjob leading to RCE
14. Publicly accessible S3 bucket
15. Desync attack (HTTP Request Smuggling)
16. Insecure HTTP method enabled (PUT, TRACE)
17. Abuse of X-Forwarded headers
18. Webhook injection

🌐 CORS Misconfigurations

1. Wildcard origin + credentials
2. Unvalidated reflected origin
3. CORS on admin panel
4. Misconfigured preflight
5. Allowed subdomain CORS leak
6. JSONP with CORS enabled
7. Internal service exposed via CORS
8. Cross-origin token access
9. Wildcard in Access-Control-Allow-Headers
10. Overly permissive CORS on private API
11. Unauthenticated endpoints with CORS
12. CORS allowed via * but credentials sent
13. Origin spoof bypass
14. Allowed local origins (localhost)
15. Legacy browser CORS bypass
16. CORS in error handling endpoint
17. Multi-origin bypass
18. Malicious iframes triggering CORS
19. CORS on logout endpoint
20. API key leak via misused CORS

🖇️ API Security Testing

1. Authentication Bypass
2. Broken Object Level Authorization (BOLA)
3. Broken Function Level Authorization (BFLA)
4. IDOR (Insecure Direct Object Reference)
5. Rate Limiting Bypass
6. HTTP Method Abuse (GET/POST/PUT/DELETE)

7. Mass Assignment
8. Sensitive Data Exposure
9. Token Leakage (JWT/API Keys)
10. Injection Attacks (SQL, NoSQL, Command)
11. API Version Exposure
12. Verbose Error Messages
13. SSRF via API
14. CORS Misconfigurations in API
15. GraphQL Endpoint IDOR
16. JSON Hijacking
17. Swagger/Docs Exposure
18. Replay Attacks (No nonce/timestamp)
19. Unrestricted File Upload via API
20. API Cache Poisoning
21. Unauthenticated API Access
22. Abuse of Batch Request APIs
23. Over-Permissioned Tokens
24. WebSocket Security in API

📱 Mobile App Security Testing

1. Insecure Local Storage (Shared Prefs, SQLite, Keychain)
2. Unencrypted Network Traffic (HTTP)
3. SSL Pinning Misconfigured or Bypassed
4. Debuggable Build Enabled
5. Reverse Engineering (APK/IPA/Dex/Smali)
6. Sensitive Data in System/Crash Logs
7. Leakage via Screen Capture/Snapshot
8. Insecure Intent Handling / Intent Sniffing
9. Bypass Root/Jailbreak Detection
10. Exported Activity/Services/Broadcasts
11. Runtime Hooking (Frida/Xposed Detection)
12. Hardcoded Secrets/API Keys in App
13. Insecure Deeplinks / URL Schemes
14. Sensitive Data Leaked to Clipboard
15. WebView Misuse (JavaScript Injection, File Access)
16. Backup Enabled for Sensitive Data
17. Injection via Autofill or Paste Events
18. Accessing Root Files/System Resources
19. SSL/TLS Disabled in Some Components

20. Insecure Authentication Flow (Tokens, OTPs)

IoT Device Security Testing

1. Physical Access (JTAG, UART, SWD, Debug Ports)
2. Firmware Extraction & Analysis (Binwalk, strings, Ghidra)
3. Cleartext Credentials or API Keys in Firmware
4. Unsigned/Unencrypted Firmware Updates
5. Exposed Network Services (Telnet, FTP, Web UI, SSH)
6. Web Interface Authentication & Session Handling
7. Cloud API & Mobile App Communication Security
8. Bluetooth/BLE Vulnerabilities (Unauthenticated Pairing, Sniffing)
9. Zigbee/Z-Wave/NFC/WiFi Attack Surface
10. Default Credentials or Hardcoded Logins
11. Access via Serial Interfaces (Bootloader/Recovery)
12. Insecure Boot Process / No Secure Boot
13. Open Ports Enumeration & Exploitation
14. Use of Weak or Custom Encryption Protocols
15. Buffer Overflows or Memory Corruption (Stack/Heap)
16. Hardcoded SSL Certificates or Private Keys
17. Insecure Mobile App Integration with IoT Device
18. Debug Logs or Verbose Output Enabled in Prod
19. LFI/RCE/Command Injection in Device Web Server
20. Certificate Pinning Implemented/Bypassable

Network Security Testing

1. Perform Active & Passive Reconnaissance
2. Port & Service Scanning (Nmap, Masscan, Rustscan)
3. Enumerate Services (SMB, RDP, FTP, SNMP, RPC)
4. Test for Default Credentials on Services
5. Packet Sniffing & Traffic Analysis (Wireshark, tcpdump)
6. Man-in-the-Middle (MITM) Attacks (ARP Spoofing, DNS Poisoning)
7. Run Vulnerability Scanners (Nessus, OpenVAS, Nexpose)
8. Attempt Firewall, IDS/IPS Evasion (Fragmentation, Encoding)
9. Check for Lateral Movement & Pivoting (proxychains, socks5)
10. Wi-Fi Attacks (WPA2 Cracking, Evil Twin, Deauth, Rogue AP)
11. Rogue DHCP Server Setup & Poisoning
12. DNS Tunneling & Exfiltration Techniques

13. VPN Exposure & Split-Tunnel Testing
14. SMB Relay Attacks (NTLMv1/v2 Capture, Pass-the-Hash)
15. Broadcast Protocol Enumeration (LLMNR, NBNS, mDNS)
16. Test SNMP (v1/v2) for Community String Bruteforce
17. Insecure Protocol Use (Telnet, FTP, Rlogin)
18. SSL/TLS Misconfigurations (Weak Ciphers, Expired Certs)
19. ICMP Tunneling (Data Exfil via Ping)
20. Misconfigured Proxies (open SOCKS, open HTTP)
