06-Burp-Suite-Cheat-Sheet

Burp Suite Cheat Sheet

Quick reference for efficient use of Burp Suite during web application penetration testing.

---

1. Proxy Setup

- Set browser/system proxy to `127.0.0.1:8080` (default)
- Install Burp's CA certificate in the browser to intercept HTTPS traffic
- Use FoxyProxy (or similar) for quick proxy toggling in the browser
- **Scope:** Define target scope under *Target > Scope* to filter noise across all tools

2. Target Tab

- **Site map:** Passive record of all traffic seen; review for hidden/forgotten endpoints
- Right-click → "Add to scope" to focus subsequent tooling
- Use *Filter* bar to hide out-of-scope/static content (images, CSS, JS) for cleaner review

3. Proxy > HTTP History

- Review every request/response pair captured
- Use filters (status code, MIME type, search term) to narrow results
- Right-click any request to send to Repeater, Intruder, Comparer, or Decoder

4. Repeater — Manual Request Manipulation

- Send a request from Proxy history (Ctrl+R) to manually tweak and resend
- Use tabs to keep multiple variations organized
- Great for: testing IDOR (changing ID values), auth bypass, injection payload refinement
- Use "Inspector" panel to quickly edit headers/params without hand-editing raw text

5. Intruder — Automated Fuzzing

- **Sniper:** Single payload set, one position at a time — good for fuzzing one parameter
- **Battering ram:** Same payload in all positions simultaneously
- **Pitchfork:** Multiple payload sets, iterated in parallel (paired values)
- **Cluster bomb:** Multiple payload sets, all combinations — good for credential brute force (username × password)
- Payload types: Simple list, numbers, brute forcer, custom iterator
- Use "Grep - Match" and "Grep - Extract" to flag interesting responses automatically
- Community Edition rate-limits Intruder — budget time accordingly

6. Decoder

- Quick encode/decode: URL, Base64, HTML, Hex, ASCII hex, Gzip
- Use "Smart decode" to auto-chain multiple encodings
- Useful for decoding JWTs, session tokens, or obfuscated parameters

7. Comparer

- Diff two requests/responses side-by-side (word or byte level)
- Useful for spotting subtle differences between valid/invalid login responses (user enumeration), or authorized/unauthorized access attempts

8. Sequencer

- Analyzes randomness/entropy of tokens (session IDs, reset tokens, CSRF tokens)
- Capture 100+ samples, then run analysis to check for predictability

9. Extender / BApp Store (Essential Extensions)

- **Logger++** — enhanced logging and filtering across all tools
- **Autorize** — automated access control/IDOR testing
- **JSON Web Tokens** — decode/edit/attack JWTs, including alg confusion testing
- **Turbo Intruder** — high-speed fuzzing for race conditions and large wordlists
- **Param Miner** — discover hidden/unlinked parameters and headers
- **Active Scan++** — additional active scan checks (Pro)
- **Backslash Powered Scanner** — deeper injection detection logic
- **Software Vulnerability Scanner** — dependency/version-based CVE detection

10. Burp Scanner (Pro)

- Active vs. Passive scanning — passive is always-on and non-intrusive; active sends attack payloads
- Scope scans carefully to avoid hammering production/out-of-scope systems
- Review and manually verify every automated finding — false positives are common

11. Match and Replace

- Under *Proxy* > *Options*, use Match and Replace rules to automatically modify requests/responses on the fly (e.g., force a header, bypass a client-side check, downgrade a security header for testing)

12. Session Handling Rules

- Under *Project Options* > *Sessions*, configure macros to auto-refresh CSRF tokens or re-authenticate during automated scans/Intruder runs

13. Workflow Tips

- Save project files regularly (`.burp`) for evidence and resumability
- Use descriptive Repeater tab names for report writeback later
- Export request/response pairs directly for report evidence (right-click → "Save item")
- Combine with command-line tools (ffuf, sqlmap) by exporting requests as raw `.txt` for reuse
- Use "Highlight" and "Comment" on HTTP history items to track findings as you go

---

Reference: PortSwigger Web Security Academy (free) for hands-on labs matching each tool above.