Burp Suite — Fiche d'utilisation et composants

Rédigé par : Assistant

16 octobre 2025

Résumé

Document synthétique expliquant les composants principaux de Burp Suite, leur rôle dans un workflow de test d'applications web, bonnes pratiques et recommandations opérationnelles. Ce guide est orienté formation et usage responsable : n'effectuez jamais de tests sur des cibles sans autorisation écrite.

Table des matières

1 Introduction **1**

2 Architecture générale **2**

3 Composants principaux **2**
3.1 Proxy (Intercept) . . . 2
3.2 Target (Sitemap & Scope) . . . 2
3.3 Repeater . . . 2
3.4 Intruder . . . 2
3.5 Scanner (Professional) . . . 2
3.6 Spider / Crawler . . . 2
3.7 Decoder . . . 2
3.8 Comparer (Comparer) . . . 3
3.9 Sequencer . . . 3
3.10 Collaborator . . . 3
3.11 Extender & BApp Store . . . 3

4 Workflow type pour un test manuel **3**

5 Bonnes pratiques et recommandations **3**

6 Extensions couramment utiles (exemples) **4**

7 Ressources recommandées **4**

1 Introduction

Burp Suite est une plateforme intégrée pour l'analyse de la sécurité des applications web. Elle centralise des outils pour intercepter, manipuler, automatiser et analyser le trafic HTTP(S). Il existe plusieurs éditions (Community, Professional) : certaines fonctionnalités avancées (scanner, collaboration OOB complet, vitesse Intruder) sont réservées à l'édition professionnelle.

2 Architecture générale

Burp agit comme un *proxy* interceptant entre le navigateur (client) et l'application (serveur). Autour du proxy gravitent des modules dédiés à des tâches spécifiques : découverte, manipulation manuelle, fuzzing, analyse automatique, encodage/décodage, et extensions.

3 Composants principaux

3.1 Proxy (Intercept)

- **Rôle :** intercepter, afficher et modifier les requêtes et réponses HTTP(S) en temps réel.
- **Usage typique :** configurer le navigateur pour pointer vers le proxy local (ex. 127.0.0.1 :8080). Activer/désactiver l'interception, modifier puis forwarder.
- **Remarque :** utile pour comprendre flux, cookies, headers et pour manipulations manuelles.

3.2 Target (Sitemap & Scope)

- **Rôle :** répertorier les URLs visitées et définir le périmètre des tests.
- **Usage typique :** marquer un domaine/chemin comme *in-scope* pour autoriser actions automatiques (scan, spider).
- **Bonnes pratiques :** définir clairement le scope pour éviter des actions sur des cibles externes.

3.3 Repeater

- **Rôle :** rejouer et modifier manuellement une requête de façon itérative.
- **Usage typique :** envoyer une requête interceptée, modifier corps/headers, renvoyer, analyser la réponse.
- **Remarque :** excellent pour tests manuels pas-à-pas et diagnostic.

3.4 Intruder

- **Rôle :** outil d'injection/fuzzing automatisé (placements de payloads, listes de mots).
- **Usage typique :** définir positions, choix de payloads, stratégies (Sniper, Pitchfork, ClusterBomb).
- **Remarque :** puissant mais potentiellement intrusif ; à utiliser sur environnement autorisé et contrôlé.

3.5 Scanner (Professional)

- **Rôle :** scanner automatique à la recherche de vulnérabilités (XSS, SQLi, RCE, etc.).
- **Usage typique :** lancer un scan sur des cibles en scope ; analyser les vulnérabilités reportées.
- **Remarque :** complémentaire à l'analyse manuelle ; générateur de faux positifs/fausse charge si mal utilisé.

3.6 Spider / Crawler

- **Rôle :** découvrir automatiquement pages et endpoints en suivant liens et formulaires.
- **Usage typique :** remplir le sitemap avant un scan.

3.7 Decoder

- **Rôle :** encoder/décoder différentes formes (URL, Base64, hex, HTML entities, etc.).

— **Usage typique :** analyser ou construire données encodées.

3.8 Comparer (Comparer)

— **Rôle :** mettre en évidence différences entre deux réponses (diff).
— **Usage typique :** détecter variations subtiles après modifications de la requête.

3.9 Sequencer

— **Rôle :** analyser la qualité (entropie, prévisibilité) d'un jeton ou d'un cookie.
— **Usage typique :** collecter de nombreux échantillons de tokens et calculer des métriques d'entropie.

3.10 Collaborator

— **Rôle :** détecter les interactions hors-bande (OOB) générées par la cible (DNS/HTTP) vers un domaine contrôlé.
— **Usage typique :** injecter un domaine Collaborator dans une requête, attendre des callbacks côté Burp.
— **Remarque :** indispensable pour trouver certaines vulnérabilités aveugles (SSRF, blind SQLi OOB).

3.11 Extender & BApp Store

— **Rôle :** ajouter des extensions (plugins) pour étendre Burp (Hackvertor, Logger++, Autorize, ActiveScan++...).
— **Usage typique :** installer extensions depuis le BApp Store pour automatiser tâches ou interpréter formats particuliers.

4 Workflow type pour un test manuel

1. **Configurer le proxy** et le navigateur (certificat HTTPS si nécessaire).
2. **Naviguer dans l'application** pour remplir le *sitemap* (Target).
3. **Intercepter** des requêtes dans le Proxy, envoyer les requêtes pertinentes vers Repeater ou Intruder.
4. **Tester manuellement** avec Repeater pour observer effets, puis automatiser des variations si approprié (Intruder).
5. **Scanner** automatiquement si la licence le permet (Scanner), puis valider manuellement les trouvailles.
6. **Utiliser Collaborator** pour détecter interactions hors-bande si nécessaire.
7. **Documenter** toutes les étapes, logs et preuves pour remédiation.

5 Bonnes pratiques et recommandations

— **Toujours** disposer d'une autorisation écrite avant d'effectuer des tests.
— Définir un **scope clair** et l'appliquer dans Target.
— Préférer les environnements de test/staging pour les actions intrusives.
— Activer/désactiver les modules lourds (Scanner/Intruder) selon la sensibilité de l'environnement.
— Surveiller les logs applicatifs et la charge serveur lors de tests (éviter DoS involontaire).
— Conserver un historique des requêtes et des réponses pour faciliter le debugging et la remédiation.

— Compléter l'automatisation par une revue manuelle : les outils peuvent produire des faux positifs.
— Tenir Burp et les extensions à jour.

6 Extensions couramment utiles (exemples)

— **Hackvertor** : encodage/obfuscation multi-format (pratique pour tests contrôlés).
— **Logger++** : logging avancé et filtrage.
— **Autorize** : tests d'autorisation horizontale/verticale.
— **ActiveScan++** : renforce les capacités du scanner.

7 Ressources recommandées

— Documentation officielle Burp Suite (PortSwigger).
— PortSwigger Academy — cours et labs (légal).
— OWASP — bonnes pratiques et guides sur sécurité web.