RÉPUBLIQUE FRANÇAISE
Liberté
Égalité
Fraternité

LES ESSENTIELS

SÉLECTION D'UN LOGICIEL LIBRE

Le choix d'un produit (outil, bibliothèque, cadriciel) pour son organisation nécessite une réflexion préalable car il peut avoir une incidence forte sur la sécurité du système d'information. Dans le cas d'un produit *open source*, les critères peuvent varier par rapport à un produit propriétaire.

Ce document présente une liste de critères visant à prendre conscience des enjeux de cybersécurité et ainsi réaliser un choix éclairé. Il n'est pas nécessaire pour un projet d'obtenir un score parfait.

- **Analyser l'historique et la notoriété du projet.** Une faible activité peut indiquer un niveau de maturité et de stabilité élevé, mais aussi un manque de ressources pour faire vivre le projet et développer de nouvelles fonctionnalités. Peuvent être pris en compte :
  - la date de création, la régularité des contributions, le nombre et expérience des principaux contributeurs ;
  - la maturité du projet telle qu'estimée par ses développeurs (ex. : version « beta ») ou par l'écosystème (utilisateurs, presse professionnelle, SILL, etc.).
- **Évaluer le maintien en condition opérationnelles (MCO) du logiciel** via :
  - la déclaration de mainteneur(s) désigné(s) ;
  - les *commits,* versions récentes (sur la dernière année) et la correction des *bugs* fonctionnels ;
  - les évolutions de l'écosystème autour du projet (plates-formes sous-jacentes, bibliothèques utilisées, etc.), qui nécessitent souvent des efforts d'adaptation ou de portage.
- **Évaluer le maintien en conditions de sécurité (MCS)**, notamment :
  - la présence d'un point de contact sur les aspects de sécurité, ainsi que d'une procédure publique de gestion des vulnérabilités ;
  - le nombre de correctifs de sécurité et le temps de correction des vulnérabilités critiques. S'il est normal qu'un projet soit confronté à des vulnérabilités durant son cycle de vie, il est important qu'il soit en mesure de recevoir et de traiter ce type de rapports de *bugs* ;
  - voir aussi le guide de l'Open Source Security Foundation (OSSF) sur le processus de divulgation des vulnérabilités pour les projets *open source* (uniquement disponible en anglais).
- **Inventorier et surveiller les dépendances du projet** (bibliothèques partagées, cadriciel, etc.) :
  - vérifier la présence d'une liste de dépendances (SBOM dans un format tel que SPDX ou CycloneDX) ;
  - s'assurer que les dépendances soient à jour et exemptes de vulnérabilités connues ;
  - privilégier les outils de gestion des dépendances qui contiennent des informations sur les vulnérabilités.
- **Identifier d'éventuelles analyses de sécurité par des tiers**, telles que la déclaration d'un visa de sécurité de l'ANSSI ou des rapports d'évaluation de la communauté *open source*.

V1.0 (05/25)

www.cyber.gouv.fr / conseil.technique@ssi.gouv.fr

- **Évaluer la qualité du socle technique**, attestée par :
  - la présence de documentation utilisateur et administrateur ;
  - une configuration par défaut et des recommandations de déploiement sécurisé ;
  - l'utilisation de standards et de protocoles ouverts et éprouvés.
- **Identifier d'éventuelles déclarations de pratiques de développement :**
  - conformité à des recommandations d'implémentation sécurisée, notamment les règles de programmation pour le développement sécurisé de logiciels en langage C et pour le développement d'applications sécurisées en Rust publiées par l'ANSSI, ainsi que l'*OWASP Developer Guide* et le guide de la NSA sur la gestion de la mémoire (uniquement disponible en anglais) ;
  - revue de code par les pairs et analyse de code outillée ;
  - présence de tests unitaires et d'intégration continue ;
  - reproductibilité des binaires générés ;
  - bonnes propriétés de sécurité des langages utilisés (sécurité de la gestion de la mémoire, typage fort, etc.).
- **Faire auditer de manière régulière l'évolution des contributions du projet et de sa sécurité**, en combinant analyse manuelle et outillée. On cherchera ainsi à se protéger de l'insertion de code illégitime, comme l'illustre le cas du projet XZ-utils : un contributeur malveillant avait inséré une porte dérobée permettant la prise de contrôle des postes sur lesquels cette bibliothèque était installée.
- **Souscrire, si possible, à un contrat de support (fonctionnel ou de sécurité) auprès d'une communauté ou d'une entité commerciale.** Cette contractualisation peut permettre de soutenir le projet en lui assurant stabilité financière et pérennité, tout en rassurant l'entité qui le déploie.
- **Vérifier si le projet ou ses mainteneurs ont manifesté un besoin de soutien** (financier ou en contributeurs, maintenance, code, sécurité, etc.). Ces appels explicites témoignent de l'état de santé du projet, et peuvent également permettre d'aider le projet et d'assurer sa continuité.