

## Aide-mémoire pour le téléchargement de fichiers

### Introduction

Le téléchargement de fichiers est devenu une fonctionnalité essentielle de toute application, permettant à l'utilisateur de télécharger sa photo, son CV ou une vidéo présentant un projet en cours. L'application doit être capable de se protéger contre les fichiers frauduleux et malveillants afin de garantir la sécurité de l'application et des utilisateurs.

En résumé, les principes suivants doivent être respectés pour parvenir à une mise en œuvre sécurisée du téléchargement de fichiers :

- Liste des extensions autorisées. N'autorisez que les extensions sûres et essentielles au bon fonctionnement de l'entreprise.
  - Assurez-vous que la [validation des entrées est effectuée](#). est appliquée avant la validation des extensions.
- Vérifiez le type de fichier, ne vous fiez pas à l' [en-tête Content-Type](#). car il peut être falsifié
- Modifiez le nom du fichier pour qu'il soit généré par l'application.
- Définissez une limite de longueur pour les noms de fichiers. Limitez les caractères autorisés si possible.
- Définir une limite de taille de fichier
- Seuls les utilisateurs autorisés sont autorisés à télécharger des fichiers
- Stockez les fichiers sur un serveur différent. Si cela n'est pas possible, stockez-les en dehors du système.  
racine Web
  - En cas d'accès public aux fichiers, utilisez un gestionnaire qui est associé aux noms de fichiers au sein de l'application (someid -> fichier.ext).
- Analysez le fichier avec un antivirus ou un environnement de test (sandbox) si disponible afin de vérifier qu'il ne contient pas de virus.  
contiennent des données malveillantes
- Traiter le fichier via CDR (Content Disarm & Reconstruct) si le type est applicable (PDF, DOCX, etc.)
- Assurez-vous que toutes les bibliothèques utilisées sont correctement configurées et tenues à jour.
- Protéger le téléchargement de fichiers contre les attaques [CSRF](#) attaques

### Menaces liées au téléchargement de fichiers

Afin d'évaluer et de déterminer précisément les contrôles à mettre en œuvre, il est essentiel de bien comprendre les risques encourus pour protéger vos actifs. Les sections suivantes présenteront les risques liés à la fonctionnalité de téléchargement de fichiers.

## Fichiers malveillants

L'attaquant transmet un fichier à des fins malveillantes, par exemple :

1. Exploiter les vulnérabilités du module d'analyse ou de traitement des fichiers (par exemple, [l'exploit ImageTrick, XXE](#))
2. Utiliser le fichier pour du phishing (par exemple, un formulaire de candidature).
3. Envoyer des bombes ZIP, des bombes XML (également connues sous le nom d'attaque du milliard de rires), ou simplement des fichiers volumineux de manière à saturer l'espace de stockage du serveur, ce qui nuit à sa disponibilité.
4. Écraser un fichier existant sur le système
5. Contenu actif côté client (XSS, CSRF, etc.) susceptible de mettre en danger d'autres utilisateurs si les fichiers sont accessibles au public.

## Récupération de fichiers publics

Si le fichier téléchargé est accessible publiquement, des menaces supplémentaires peuvent être prises en compte :

1. Divulcation publique d'autres fichiers
2. Lancez une attaque par déni de service (DoS) en demandant un grand nombre de fichiers. Les requêtes sont petites, mais les réponses sont... beaucoup plus grand
3. Contenu de fichier pouvant être considéré comme illégal, offensant ou dangereux (par exemple, données personnelles, des données protégées par le droit d'auteur, etc.) qui feront de vous un hôte pour de tels fichiers malveillants.

## Protection contre le téléchargement de fichiers

Il n'existe pas de solution miracle pour valider le contenu utilisateur. La mise en œuvre d'une approche de défense en profondeur est essentielle pour renforcer la sécurité du processus de téléchargement et l'adapter aux besoins et exigences du service. Il est crucial et recommandé d'utiliser plusieurs techniques, car aucune ne suffit à elle seule pour garantir la sécurité du service.

## Validation de l'extension

Veillez à ce que la validation intervienne après le décodage du nom de fichier et qu'un filtre approprié soit mis en place afin d'éviter certaines techniques de contournement connues, telles que les suivantes :

- Les doubles extensions, par exemple `.jpg.php` , permettent de contourner facilement l'expression régulière `\.jpg`
- Octets nuls, par exemple `.php%00.jpg` , où `.jpg` est tronqué et `.php` devient le nouveau extension
- Expression régulière générique et de mauvaise qualité, insuffisamment testée et vérifiée. Évitez de créer votre propre logique si vous ne possédez pas de connaissances suffisantes sur le sujet.

Consultez le [CS de validation des entrées](#) pour analyser et traiter correctement l'extension.

## Liste des extensions autorisées

Veillez à n'utiliser que les extensions critiques pour l'activité, en interdisant toute extension non indispensable. Par exemple, si le système requiert :

- Téléchargement d'images : autoriser un seul type convenu pour répondre aux exigences de l'entreprise ;
- Téléchargement de CV, extensions docx et pdf acceptées.

En fonction des besoins de l'application, veillez à utiliser les types de fichiers les moins nuisibles et présentant le moins de risques.

## Extensions de blocs

Identifiez les types de fichiers potentiellement dangereux et bloquez les extensions que vous jugez nuisibles à votre système. service.

Veillez noter que le blocage d'extensions spécifiques constitue une méthode de protection insuffisante à lui seul. [Vulnérabilité liée au téléchargement de fichiers sans restriction](#) Cet article décrit comment les attaquants peuvent tenter de contourner un tel contrôle.

## Validation du type de contenu

Le type de contenu des fichiers téléchargés est fourni par l'utilisateur et, de ce fait, ne peut être considéré comme fiable, car il est très facile à falsifier. Bien qu'il ne faille pas s'y fier pour garantir la sécurité, il permet une vérification rapide afin d'éviter que les utilisateurs ne téléchargent involontairement des fichiers de type incorrect.

Outre la définition de l'extension du fichier téléchargé, son type MIME peut être vérifié pour une protection rapide contre les attaques par téléchargement de fichiers simples.

Cela peut se faire de préférence avec une approche par liste blanche ; sinon, cela peut se faire avec une approche par liste noire.

## Validation de la signature du fichier

En parallèle de [la validation du type de contenu](#), la signature du fichier peut être vérifiée et comparée au fichier attendu.

Il ne faut pas utiliser cette méthode seule, car la contourner est assez courant et facile.

## Sécurité des noms de fichiers

Les noms de fichiers peuvent mettre le système en danger de plusieurs manières, notamment en utilisant des caractères non autorisés ou des noms de fichiers spéciaux et restreints. Pour Windows, consultez le [guide MSDN suivant](#).

Pour un aperçu plus complet des différents systèmes de fichiers et de leur traitement des fichiers, consultez [la page Wikipédia consacrée aux noms de fichiers](#).

Afin d'éviter la menace mentionnée ci-dessus, il est essentiel de générer une chaîne aléatoire comme nom de fichier, par exemple un UUID ou un GUID. Si le nom de fichier est indispensable aux besoins métier, une validation rigoureuse des entrées doit être mise en œuvre pour les attaques côté client (par exemple, contenu actif susceptible d'entraîner des attaques XSS et CSRF) et côté serveur (par exemple, écrasement ou création de fichiers spécifiques). La longueur des noms de fichiers doit être limitée par le système de stockage, chaque système ayant ses propres contraintes. Si l'utilisation de noms de fichiers personnalisés est requise, il est recommandé d'implémenter les mesures suivantes :

- Mettre en œuvre une longueur maximale
- Limitez les caractères à un sous-ensemble autorisé spécifique, comme les caractères alphanumériques, le tiret, les espaces et les points.
  - Il serait judicieux d'indiquer à l'utilisateur quel est un nom de fichier acceptable.
  - Limiter l'utilisation des points de début (fichiers cachés) et des points séquentiels (parcours de répertoire).
- Limitez l'utilisation d'un tiret ou d'espaces en début de nom pour sécuriser le traitement des fichiers par des scripts shell.
- Si cela n'est pas possible, bloquez les caractères dangereux susceptibles de mettre en danger le système et l'infrastructure qui stockent et utilisent les fichiers.

## Validation du contenu des fichiers

Comme indiqué dans la section [Récupération de fichiers publics](#) , le contenu des fichiers peut contenir des éléments malveillants. données inappropriées ou illégales.

En fonction du type attendu, une validation spéciale du contenu des fichiers peut être appliquée :

- Pour les images, l'application de techniques de réécriture d'images détruit tout type de contenu malveillant injecté dans une image ; cela pourrait être réalisé par [randomisation](#).
- Pour les documents Microsoft, l'utilisation d' [Apache POI](#) permet de valider les données téléchargées documents.
- Les fichiers ZIP ne sont pas recommandés car ils peuvent contenir tous types de fichiers, et les vecteurs d'attaque qui leur sont associés sont nombreux.

Le service de téléchargement de fichiers devrait permettre aux utilisateurs de signaler les contenus illégaux et aux titulaires de droits d'auteur de signaler les abus.

Si les ressources le permettent, un examen manuel des fichiers devrait être effectué dans un environnement isolé avant leur diffusion au public.

L'automatisation du processus d'analyse pourrait s'avérer utile. Ce processus, complexe, doit être étudié en détail avant toute utilisation. Certains services (comme VirusTotal) proposent des API permettant d'analyser les fichiers à l'aide d'empreintes numériques de fichiers malveillants connus. Certains frameworks peuvent vérifier et valider les données brutes.

type de contenu et sa validation par rapport à des types de fichiers prédéfinis, comme dans [la bibliothèque de dessin ASP.NET](#).

Attention aux risques de fuite de données et à la collecte d'informations par les services publics.

#### Emplacement de stockage des fichiers

L'emplacement de stockage des fichiers doit être choisi en fonction des exigences de sécurité et des impératifs métier. Les points suivants sont définis par ordre de priorité en matière de sécurité et sont non exhaustifs :

1. Stockez les fichiers sur un serveur différent, ce qui permet une séparation complète des tâches.  
entre l'application qui sert l'utilisateur et l'hôte qui gère les téléchargements de fichiers et leur stockage.
2. Stockez les fichiers en dehors du répertoire racine du site web, où seul l'accès administratif est autorisé.
3. Stockez les fichiers dans le répertoire racine du site web et configurez-les en mode écriture uniquement. – Si un accès en lecture est requis, il est impératif de mettre en place des contrôles d'accès appropriés (par exemple, adresse IP interne, utilisateur autorisé, etc.).

Stocker les fichiers de manière structurée dans des bases de données est une technique supplémentaire. Elle est parfois utilisée pour les sauvegardes automatiques, les attaques hors système de fichiers et les problèmes de permissions. En revanche, elle peut engendrer des problèmes de performance (dans certains cas), des contraintes de stockage pour la base de données et ses sauvegardes, et exposer les utilisateurs à des injections SQL. Son utilisation est recommandée uniquement en présence d'un administrateur de base de données et si elle s'avère plus efficace que le stockage sur le système de fichiers.

Certains fichiers sont envoyés par courriel ou traités dès leur téléchargement et ne sont pas stockés sur le serveur. Il est essentiel d'appliquer les mesures de sécurité décrites dans cette fiche avant toute manipulation de ces fichiers.

#### Autorisations de l'utilisateur

Avant tout accès à un service de téléchargement de fichiers, une validation appropriée doit être effectuée à deux niveaux pour l'utilisateur qui télécharge un fichier :

- Niveau d'authentification
  - L'utilisateur doit être un utilisateur enregistré ou identifiable pour pouvoir définir des restrictions et des limitations concernant ses capacités de téléchargement.
- Niveau d'autorisation
  - L'utilisateur doit disposer des autorisations appropriées pour accéder aux fichiers ou les modifier.

#### Autorisations du système de fichiers

Définissez les permissions des fichiers selon le principe du moindre privilège.

Les fichiers doivent être stockés de manière à garantir :

- Seuls les utilisateurs système autorisés peuvent lire les fichiers.
- Seuls les modes requis sont définis pour le fichier.
  - Si l'exécution est nécessaire, il est recommandé, par mesure de sécurité, d'analyser le fichier avant de l'exécuter afin de s'assurer qu'aucune macro ni aucun script caché n'est présent.

## Limites de téléchargement et d'envoi

L'application doit définir des limites de taille appropriées pour le service de chargement afin de préserver la capacité de stockage des fichiers. Si le système doit extraire ou traiter les fichiers, la limite de taille doit être prise en compte après la décompression et en utilisant des méthodes sécurisées pour calculer la taille des fichiers ZIP. Pour plus d'informations, consultez la documentation sur l' [extraction sécurisée de fichiers depuis ZipInputStream](#). Flux d'entrée Java pour la gestion des fichiers ZIP.

L'application devrait également définir des limites de requêtes appropriées pour le service de téléchargement, le cas échéant, afin de protéger le serveur contre les attaques par déni de service (DoS).

## Extraits de code Java

[Protection contre le téléchargement de documents](#) dépôt écrit par Dominique pour certains types de documents dans Java.