AGENCE DU NUMÉRIQUE EN SANTÉ
La transformation commence ici

Guide d’utilisation

Formulaire du test d’intrusion

Statut : En cours | Classification : Restreinte | Version : v15

Agence du Numérique en Santé
2 - 10 Rue d'Oradour-sur-Glane, 75015 Paris
T. 01 58 45 32 50
esante.gouv.fr

SOMMAIRE

1 INTRODUCTION

1.1 Le contexte du Ségur du Numérique en Santé

Le Ségur du Numérique en Santé a été créé dans l'objectif de **généraliser le partage fluide et sécurisé des données de santé entre professionnels et usagers** pour mieux prévenir et mieux soigner. Ce programme viendra alimenter Mon espace santé, qui permet à chaque citoyen de disposer d'une vision consolidée de son parcours de soins afin d'être acteur de sa santé. L'Etat a donc mis en place un mécanisme d'achat au bénéfice des acteurs de l'offre de soins, sous la forme d'un système ouvert et non sélectif (SONS) de référencements d'éditeurs de solutions logicielles.

La sécurité des données de santé est aujourd'hui **au cœur des préoccupations**. Il est par conséquent essentiel que les solutions logicielles proposées par les éditeurs permettent un partage sécurisé des données de santé entre professionnels et usagers. Elles doivent notamment pouvoir s'interfacer de façon sécurisée avec les services numériques en santé tels que le **Dossier Médical Partagé** ou encore **Mon espace santé**. Ainsi, le référencement des solutions logicielles par les éditeurs est un prérequis à l'obtention d'un financement. Les éditeurs candidats au référencement doivent pour cela suivre un processus qui permet de s'assurer que leurs solutions respectent **l'ensemble des exigences techniques et fonctionnelles décrites dans un dossier de spécifications de référencement (DSR)** qui est mis à leur disposition.

1.2 Objet du document

Les éditeurs souhaitant candidater pour le référencement d'une solution qu'ils produisent doivent donc passer par **un processus d'évaluation de la conformité de cette solution à des exigences portant sur la sécurité des systèmes d'information.** La **réalisation d'un test d'intrusion** portant sur la solution candidate est notamment exigée. Celui-ci donne lieu au remplissage d'un formulaire par l'auditeur qui permet de fixer le périmètre du test et atteste des résultats obtenus. Il constitue une preuve requise pour le référencement Ségur.

Le présent document vise à décrire le mode opérationnel à suivre pour réaliser ce test d'intrusion et transmettre le formulaire complété dans ce cadre.

2 CONDITIONS & PERIMETRE DE REALISATION DU TEST D'INTRUSION

2.1 Prérequis

Le test d'intrusion doit être réalisé par un prestataire d'audit, à la demande de l'éditeur. Afin de garantir les compétences du prestataire d'audit sélectionné et ainsi l'équité du processus, **il est demandé de faire appel à un prestataire d'audit de la sécurité des systèmes d'information qualifié** (PASSI : https://www.ssi.gouv.fr/entreprise/qualifications/prestataires-de-services-de-confiance-qualifies/prestataires-daudit-de-la-securite-des-systemes-dinformation-passi-qualifies/).

Cette prestation est à la charge de l'éditeur candidat au référencement de sa solution.

La réalisation d'un audit PASSI (conditions de réalisation spécifiques et auditeur certifié PASSI) n'est pas requise. L'unique prérequis est de faire réaliser le test d'intrusion par un organisme qualifié PASSI.

2.2 Déroulement du test d'intrusion

Le test d'intrusion se déroule généralement sur une période d'une semaine. Il se découpe en trois phases représentées sur le schéma ci-dessous :

2.2.1 Phase de cadrage

L'audit technique doit être précédé d'une phase de cadrage organisée par l'éditeur pour planifier avec le prestataire sélectionné l'audit à réaliser. Ce cadrage doit permettre d'établir les modalités de réalisation de l'audit et de valider notamment :

- Les dates de réalisation de l'audit, l'environnement mis à disposition ;
- Les personnes à contacter ;
- Les éléments décrivant le fonctionnement nominal de la solution et toute information utile à l'auditeur ;
- L'ensemble des prérequis nécessaires pour l'auditeur à la réalisation de l'audit (cf. Chapitre 3. Rôle de l'éditeur).

Recommandation
Lors de la phase de cadrage, il est recommandé de prévoir **une clause de revoyure** avec l'auditeur pour les exigences non validées lors du test d'intrusion. En effet, il sera demandé à l'auditeur **d'intervenir une seconde fois (réalisation d'un contre audit) afin de vérifier si des mesures de sécurité ont été implémentées** sur les exigences précédemment non validées.

2.2.2 Phase de réalisation des tests

Durant cette phase, l'auditeur adopte le rôle d'un attaquant potentiel cherchant à identifier les vulnérabilités de l'application. En vue de vérifier des points de contrôle spécifiques, il disposera d'un accès à des éléments de l'application afin d'approfondir son analyse. La durée de la phase de test fluctue en accord avec la complexité de la solution, cependant, en moyenne, elle s'étend sur environ une semaine.

La majorité des points à contrôler sont basés sur le référentiel de l'**Open Web Application Security Project - OWASP** (cf. Annexe 1 : Définition et concepts généraux & cf. annexe 5).

Ils sont listés dans le formulaire à compléter par l'auditeur :

- **18 points de contrôle communs à toutes les solutions :** Ceux-ci doivent être obligatoirement analysés par l'auditeur quelle que soit la solution concernée ;
- **21 à 24 points de contrôle spécifiques au type de la solution** (application web, application mobile, client lourd avec une architecture à trois tiers ou plus et client lourd avec une architecture à moins de trois tiers).

Les points de contrôles au niveau du formulaire du test d'intrusion sont répartis en trois catégories :

- **Gravité haute : La non-conformité au point de contrôle attendu est éliminatoire**. L'éditeur ne sera pas éligible au référencement dans ce cas ;

- **Gravité moyenne : Jusqu'à 10 réponses négatives à des points de contrôles de gravité moyenne peuvent être acceptées au maximum** sans remettre en cause l'éligibilité au référencement sur l'ensemble du formulaire du test d'intrusion ;
- **Non applicable (NA) :** Points de contrôle s'appliquant **uniquement aux clients lourds avec une architecture moins de trois tiers** et bien qu'ils **doivent être évalués,** ils ne sont pas pris en compte dans le processus de référencement. Cette exclusion découle de l'incompatibilité entre la nature de l'architecture et les critères de contrôle.

Si un ou plusieurs manquements aux règles de sécurité sont présents, ces derniers doivent être corrigées **avant la fin du processus de référencement** afin de garantir la conformité aux seuils de validation définis ci-dessus (Aucune réponse négative pour les points de contrôle de gravité haute et jusqu'à un maximum de 10 réponses négatives pour les points de contrôle de gravité moyenne).

En cas de vulnérabilité majeure découverte sur la solution ayant un score CVSS v3 (Common Vulnerability Scoring System) supérieur ou égal à 8 (cf. Annexe 1 : Définition et concepts généraux, **l'auditeur doit en informer l'éditeur qui transmettra l'information au CERT santé**.

Par ailleurs, si l'auditeur juge que **l'exploitation d'une vulnérabilité est complexe** en raison du contexte applicatif de la solution ou des mesures de sécurité mises en place pour la protéger, il peut confirmer les règles de sécurité en incluant ces détails dans un commentaire explicatif (point de contrôle vérifié par les règles C14 & C15 dans le formulaire du test d'intrusion).

2.2.3 Phase de rapports

Une fois les tests finalisés, l'auditeur constituera le rapport du test d'intrusion en **remplissant le formulaire**. Ce document regroupera l'ensemble des **résultats** du test d'intrusion ainsi que le **référencement** de l'application.

De plus, cette phase aura pour finalité d'échanger sur le rapport du formulaire du test d'intrusion. En effet, si un ou plusieurs manquements aux règles de sécurité sont relevés, l'objectif de l'auditeur sera de transmettre **les résultats du test d'intrusion** à l'éditeur en lui expliquant :

- Les détails techniques des vulnérabilités identifiées ;
- L'impact potentiel des manquements aux règles de sécurité et un niveau de risque associé ;
- Les solutions concrètes permettant de corriger les potentielles failles ;
- La définition des prochaines étapes (définition d'une date permettant d'évaluer de nouveau les vulnérabilités recensées lors d'un audit de contrôle).

2.2.4 Phase d'audit(s) de contrôle éventuel

Dans le cas où l'éditeur doit apporter **des corrections aux vulnérabilités de sa solution pour se conformer** au seuil de validation des points de contrôle de gravité haute et/ou moyenne du test d'intrusion, **il doit informer l'auditeur des vulnérabilités corrigées**.

Ensuite, l'auditeur **teste de nouveau ces points pour vérifier les corrections** et **met à jour le rapport du test d'intrusion** avec les résultats obtenus. Une version actualisée du formulaire, incluant tous les résultats du test ainsi que le référencement de l'application, est ensuite fournie.

3 ROLE DE L'EDITEUR

L'éditeur intervient essentiellement en amont de l'audit afin d'assurer sa préparation. Cette phase est essentielle afin que l'auditeur dispose de l'ensemble des éléments nécessaires et soit ainsi en mesure de réaliser l'ensemble de la prestation dans les conditions attendues et dans le respect des délais prévus.

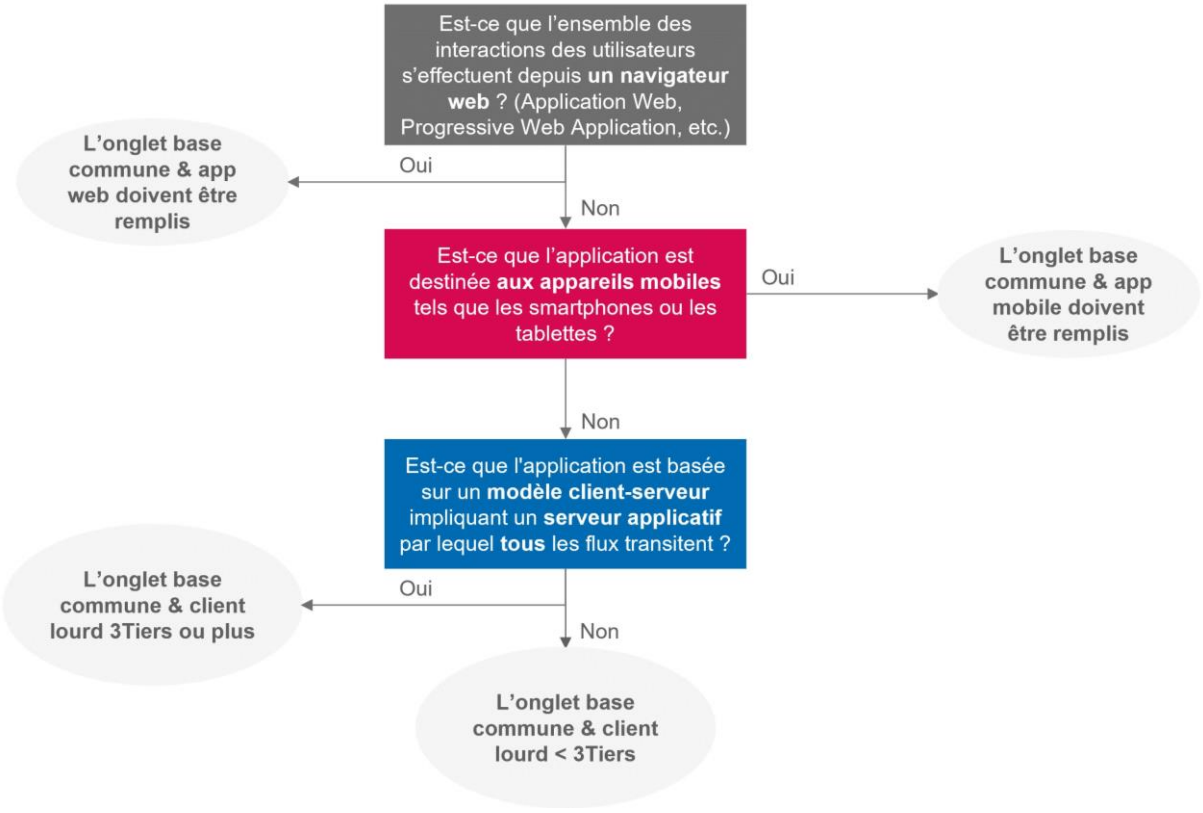
3.1 Mise à disposition de l'environnement

Concernant l'environnement mis à disposition pour la réalisation de l'audit, il est notamment indispensable de prévoir les éléments suivants :

- Le périmètre du test d'intrusion doit être parfaitement cadré pour limiter les impacts sur l'application ou le système d'information. Ainsi, la prestation doit être réalisée de préférence sur un **environnement « iso-prod »** (tel qu'un environnement de développement, de test, etc.) plutôt que sur un environnement de production ;
- L'application testée doit correspondre à la **même version majeure que celle en production** (données similaires, niveau de sécurité suffisant, paramétrage identique, etc.) afin d'obtenir une vision réaliste de la sécurité ;
    - Il est essentiel de donner les moyens à l'auditeur de tester le produit tel qu'il est commercialisé que ce soit en tant qu'application web installée localement sur l'appareil d'un utilisateur ou déployée à distance via une plateforme de virtualisation (ex : Citrix, etc.). Une désactivation de l'ensemble des dispositifs de sécurité **(WAF, sondes, passerelles, etc.)** est nécessaire, s'ils ne font pas partie de la solution commercialisée.
- Le formulaire du test d'intrusion se concentre sur **l'évaluation des vulnérabilités et des failles de sécurité au niveau logiciel**. Ainsi, **les parties matérielles des solutions** (borne physique, poste d'un usager, etc.) sont exclues du test d'intrusion.

3.2 Préparation du formulaire pour l'auditeur

Pour assurer la complétude du formulaire, l'éditeur **doit renseigner l'ensemble des informations propres à son application*** dans le formulaire du test d'intrusion (fichier Excel, onglet « 1 – Résultat Formulaire »). Une aide pour déterminer le type d'application concernée peut être trouvée grâce au logigramme ci-dessous :



**Les définitions détaillées des architectures 2-tiers et 3-tiers sont explicitées dans l'annexe 1*

3.3 Eléments à communiquer à l'auditeur

Pour assurer la réalisation de la prestation, **l'éditeur doit communiquer à l'auditeur** :

Pour une application Web

- **L'application**, une URL ou une adresse IP ;

Pour une application Mobile

- Une **APK sans pinning de certificat et sans vérification du débridage** du téléphone. Si possible, des terminaux débridés (accès complet pour déverrouiller les fonctionnalités) ;

Pour un client lourd

- **Le client et des accès pour examiner les configurations** possibles de l'application ;

Pour toutes les solutions :

- **Plusieurs comptes** avec différents niveaux de privilèges (compte utilisateur, compte à privilège) ;
- La **liste des comptes génériques** pour vérifier l'exposition de ces derniers ;
- **Une matrice des flux** spécifiant les flux essentiels au fonctionnement du système ;
- Une **extraction des logs techniques** sur les tests réalisés lors du **premier jour opérationnel** afin de suivre les tentatives d'authentification, la présence de données sensibles (toute donnée à caractère personnel, qu'elle soit ou non de santé, ou participant à la sécurité du système d'information constitué par le système seul ou le système d'information auquel il participe) et le format des événements. Ces logs doivent être transmis le plus rapidement à l'auditeur.
- **L'éditeur doit informer l'auditeur** de **toutes les vulnérabilités connues**, ainsi que des **mesures de sécurité mises en œuvre** pour les contourner.

Dans le cas des applications de type clients lourds, il est essentiel que les auditeurs disposent de **toutes les informations permettant de faciliter la prise en main de l'outil dès la phase de cadrage** afin d'éviter toute perte de temps lors de la réalisation de l'audit et limiter ainsi les dépassements des délais de la prestation.

Une fois le formulaire rempli, celui-ci sera **généré sous forme de PDF puis signé électroniquement** par l'auditeur et sera retourné à l'éditeur qui le déposera sur l'outil Convergence. Une fiche de synthèse des éléments ci-dessus est mise à disposition en Annexe 2 : Synthèse de rappel pour les éditeurs.

Outre ces prérequis, dans l'optique de permettre un audit approfondi et efficace, l'éditeur pourra mettre à disposition de l'auditeur tous les détails et informations qu'il juge pertinents concernant la solution à référencer (ex : Dossier d'Architecture Technique, documentation du projet, schémas techniques, etc.).

4 ROLE DE L'AUDITEUR

L'auditeur intervient sur la phase de la réalisation de l'audit. Cette phase est essentielle afin que l'auditeur réalise l'ensemble des tests et puisse permettre de vérifier si la solution de l'éditeur est sécurisée.

4.1 Prérequis

Lors de la phase de cadrage, l'auditeur doit s'assurer que l'ensemble des éléments lui permettant d'assurer la réalisation de la prestation lui ont été transmis (ou le seront au début de la phase de test dans le cas des logs).

Les éléments suivants doivent notamment lui être communiqués par l'éditeur (Cf. tableau. 3.3 Eléments à communiquer à l'auditeur) **:**

- L'application, une URL, ou une IP ;

- Plusieurs comptes avec différents niveaux de privilèges (compte utilisateur, compte à privilège) ;
- Une extraction des logs techniques portant sur les tests réalisés lors du premier jour opérationnel afin de vérifier les tentatives d'authentification, la présence de données sensibles et le format des événements. Ces logs doivent être mis à disposition le plus rapidement possible par l'éditeur.
- Une matrice des flux spécifiant les flux essentiels au fonctionnement du système ;
- La liste des comptes génériques.

En complément, l'auditeur pourra demander à l'éditeur lors de la phase de cadrage toute information jugée utile à la réalisation du test d'intrusion. Cette phase est essentielle afin de bien cadrer avec l'éditeur les éléments entrants à produire pour plus d'efficacité lors de la phase d'audit.

4.2 Remplissage du formulaire

Il est attendu de la part de l'auditeur de tester l'ensemble des points de contrôle listés dans le formulaire de test d'intrusion et d'évaluer la conformité de l'application à ces différents points. Il peut alors renseigner le formulaire de test d'intrusion. Plusieurs périmètres sont couverts dans le formulaire afin de donner une vision complète du niveau de sécurité de l'application. **Les consignes suivantes doivent impérativement être respectées** :

Gestion des onglets

Uniquement deux onglets doivent être obligatoirement remplis par les auditeurs. L'onglet « Base Commune » ainsi que l'onglet correspondant au type d'application audité (information complétée par l'éditeur dans l'onglet 1 – Résultat formulaire sur la ligne « Type d'application »)

« N/A ou Non »

La notation "N/A" ou "NON" doit être **limitée** et **nécessite une justification** en commentaire. La notation N/A est utilisée lorsque la règle de sécurité **ne dépend pas de l'éditeur mais de la structure utilisatrice** (ex : gestion des serveurs physiques dans le cadre d'une solution SaaS) ou **n'est pas applicable au système.**

Notation des règles

Seule la notation "Oui, Non ou N/A" est autorisée afin de savoir si l'ensemble des mesures de sécurité sont mises en œuvre. **Pour chacune des règles de sécurité, une notation doit être attribuée.**

Gestion des formulaires

Si l'application possède plusieurs types d'environnements (ex : application Web et application mobile), **un formulaire doit être réalisé pour chaque partie** (ex : un formulaire pour l'application web et un second pour l'application mobile). Il en est de même si l'application possède plusieurs types d'infrastructures (SaaS, on-premise, etc.)

4.2 Génération PDF & Signature électronique

Une fois le formulaire rempli, **un fichier PDF reprenant les éléments qu'il contient doit être généré à partir de la macro du formulaire du test d'intrusion puis signé électroniquement, avec l'approbation d'un TSP** (Trust Service Provider). Pour apposer cette signature électronique, deux options sont possibles (détails des étapes disponibles dans l'annexe 4 : processus autour de la signature électronique) :

- Via une application bureautique telle qu'Adobe avec un certificat de signature délivré par un TSP ;
- Via une plateforme de signature électronique reconnue telle que DocuSign, Docaposte, YouSign, etc. ;

Ce fichier signé électroniquement, avec l'approbation du TSP via un certificat de signature ou via une plateforme de signature électronique, constitue une preuve essentielle que l'éditeur devra soumettre dans le cadre de sa candidature pour le référencement Ségur.

En plus du rapport, dans le cas où certains points de contrôles auraient été jugés non conformes, l'auditeur pourra proposer à l'éditeur des mesures correctives à mettre en œuvre. Une fiche de synthèse avec les principaux éléments qui concernent l'auditeur est mise à disposition en Annexe 3 : Synthèse de rappel pour les auditeurs.

Annexe 1 : Définition et concepts généraux

Ce guide d'utilisation ainsi que le formulaire du test d'intrusion font référence à des concepts précis de la sécurité des systèmes d'information. Ainsi, certaines précisions et définitions sont proposées ci-dessous afin d'assurer une compréhension commune pour faciliter leur compréhension.

Donnée sensible (est considérée comme telle) :

- Toute donnée à caractère personnel, qu'elle soit ou non de santé ;
- Toute information participant à la sécurité du système d'information constitué par le système seul ou le système d'information auquel il participe.

Opération sensible, dans l'utilisation du système (est considérée comme telle) :

- Tout déclenchement d'action susceptible d'induire directement ou indirectement l'enregistrement, l'affichage, la transmission, la modification ou l'effacement d'information sensible, sauf à ce que l'analyse de risque du système ait déterminé que l'action déclenchée sur les informations considérées n'était pas porteuse d'enjeu particulier ;
- Tout déclenchement d'action susceptible d'induire des effets négatifs sur la santé de patients ;
- Tout autre déclenchement d'action identifiée par l'analyse de risque du système comme devant faire l'objet de mesures de protection et/ou de contrôle renforcées.

L'Open Web Application Security Project (OWASP) est une fondation à but non lucratif qui œuvre à l'amélioration de la sécurité des logiciels. La fondation OWASP est une source d'information mondialement reconnue par les spécialistes et suivie par les développeurs pour sécuriser les applications de type web.

La réalisation du test d'intrusion dont il est question dans le présent document et les thématiques abordées dans le formulaire associé se basent essentiellement sur le Top 10 de l'OWASP. Il s'agit d'un classement de référence des principales attaques rencontrées sur les applications web notamment :

A01:2021 – Broken Access Control	A06:2021 – Vulnerable and Outdated Components
A02:2021 – Cryptographic Failures	A07:2021 – Identification and Authentication Failures
A03:2021 – Injection	A08:2021 – Software and Data Integrity Failures
A04:2021 – Insecure Design	A09:2021 – Security Logging and Monitoring Failures
A05:2021 – Security Misconfiguration	A10:2021 – Server-Side Request Forgery (SSRF)

Score CVSS v3 (Common Vulnerability Scoring System): système standardisé de notation des vulnérabilités établi par le FIRST (Forum of Incident Response and Security Teams). Il attribue à une vulnérabilité un score, qui est entre 0 et 10, en fonction de plusieurs critères, tels que la complexité de l'exploitation, l'impact sur la confidentialité, l'intégrité et la disponibilité des données, et d'autres facteurs. Plus le score est élevé, plus la vulnérabilité est critique (niveau faible, moyen, élevé ou critique).

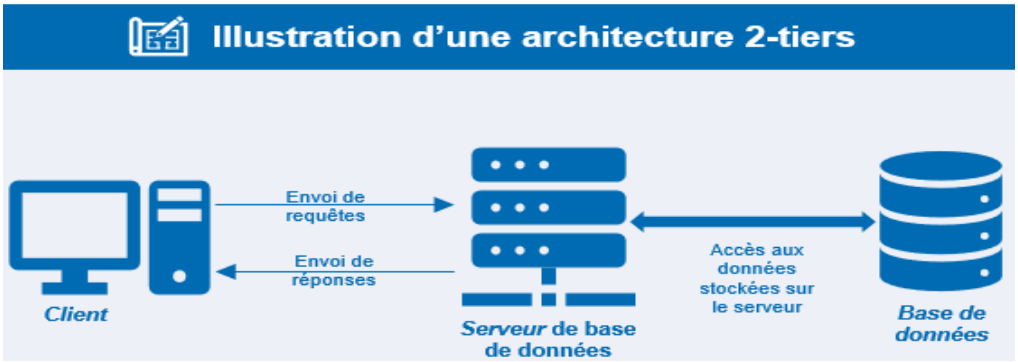
TSP (Trust Service Provider) : Un TSP est un fournisseur de services de confiance qui fournit des certificats numériques utilisés pour authentifier et sécuriser les signatures électroniques. Ces certificats sont associés à une identité spécifique (par exemple, une personne ou une organisation) et contiennent des informations telles que la clé publique de cette identité, des données d'authentification, et sont signés numériquement par l'autorité de certification. Le TSP joue un rôle essentiel en vérifiant l'identité du titulaire du certificat avant de lui délivrer un certificat de signature. Cela assure la confiance dans les signatures électroniques, car elles sont associées à des identités vérifiées.

Client – serveur : architecture désignant la séparation des tâches d'une application entre deux entités distinctes et cloisonnées. Le côté client désigne l'ensemble des composants logiciels manipulés par les utilisateurs. Le côté

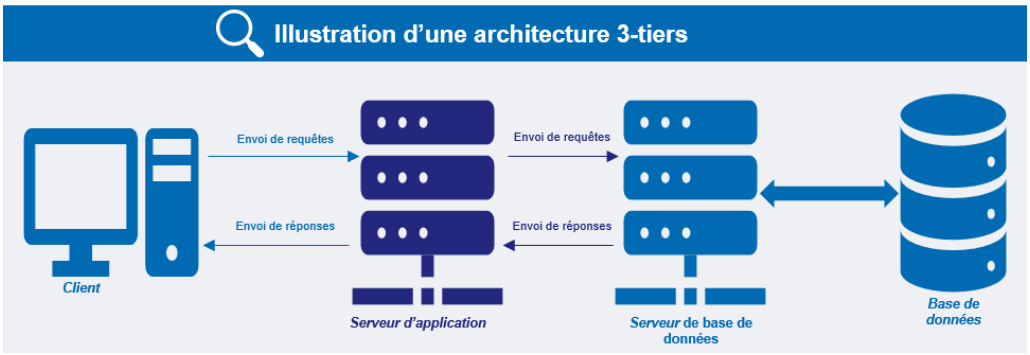
serveur désigne l'ensemble des composants logiciels responsables des traitements, c'est-à-dire l'implémentation de la logique métier, et de l'accès aux données.

Exemple générique d'application client-serveur : Système où les clients (utilisateurs) se connectent à un serveur pour demander des services ou des ressources. Le serveur fournit ces services ou ressources en réponse aux requêtes des clients. Cette architecture est utilisée dans une variété d'applications, telles que la messagerie instantanée, le partage de fichiers, les services de bases de données, etc. Les clients envoient des requêtes au serveur, qui les traite et renvoie des réponses appropriées. Les clients interagissent avec le serveur pour obtenir des informations ou effectuer des actions, ce qui permet une distribution efficace des services et des ressources.

Architecture 2-tiers : c'est une architecture qui se compose de deux principales couches : la couche client et la couche serveur. Dans cette architecture, le client réalise directement les requêtes sur le serveur de base de données.



Architecture 3-tiers : c'est une architecture qui se compose de trois principales couches : la couche client, la couche applicative et la couche données. Dans cette architecture, le client réalise les requêtes sur la couche applicative qui requête elle-même le serveur de base de données.



Journaux techniques : Journaux (logs) contenant des informations relatives au fonctionnement des SI (ex : logs des serveurs, des applications, des composants matériels, etc.) Ils comprennent généralement des détails sur les erreurs, les défaillances, les performances, les données de débogage, l'activité du réseau, etc.

Journaux métiers : Journaux (logs) contenant des informations liées aux activités, aux opérations spécifiques aux métiers, et peuvent englober des données relatives aux transactions, aux utilisateurs, aux applications, etc. qui sont essentielles dans le contexte des activités d'une entreprise.

Secret : donnée sensible et protéiforme caractérisée par son usage dans les processus d'authentification des utilisateurs et de contrôle d'accès à des ressources et/ou fonctionnalités de composants système ou applicatif. Elle peut désigner des identifiants de comptes, identifiants de session, des mots de passe, clé de chiffrement, clés d'API, jetons d'authentification, etc.

Annexe 2 : Synthèse de rappel pour les éditeurs

- Le test d'intrusion doit être effectué par un prestataire d'audit de la sécurité des systèmes d'information qualifié (PASSI) ;
- Lors de la phase de cadrage, il est recommandé de prévoir une « clause de revoyure » avec l'auditeur afin de valider les corrections effectuées dans l'hypothèse où certaines exigences ne seraient pas validées lors du test d'intrusion ;
- La prestation est à réaliser de préférence dans un environnement iso-prod (tel qu'un environnement de développement, de test, etc.) ;
- L'application testée doit correspondre à la même version majeure que celle en production ;
- L'auditeur ne doit pas être limité par les dispositifs de sécurité (WAF, sondes, etc.), ceux-ci doivent impérativement être désactivés s'ils ne font pas partie de la solution commercialisée ;
- L'éditeur doit compléter dans le formulaire du test d'intrusion l'ensemble des informations propres à son application dans l'onglet « 1 – Résultat Formulaire » ;
- Le formulaire atteste de la réalisation du test d'intrusion. Il doit être complété puis signé électroniquement par l'auditeur dans un format PDF qu'il aura généré en suivant les indications du formulaire. Ce document constitue une preuve à fournir dans le cadre de la candidature au référencement Ségur ;
- Selon le type d'application, l'éditeur doit communiquer à l'auditeur l'ensemble des informations recensées dans le tableau suivant :

Pour une application Web	Pour une application Mobile	Pour un client lourd
› **L'application**, une URL ou une adresse IP ;	› Une **APK sans pinning de certificat et sans vérification du débridage** du téléphone. Si possible, des terminaux débridés (accès complet pour déverrouiller les fonctionnalités) ;	› **Le client et des accès pour examiner les configurations** possibles de l'application ;

Pour toutes les solutions :

- **Plusieurs comptes** avec différents niveaux de privilèges (compte utilisateur, compte à privilège) ;
- La **liste des comptes génériques** pour vérifier l'exposition de ces derniers ;
- **Une matrice des flux** spécifiant les flux essentiels au fonctionnement du système ;
- Une **extraction des logs techniques** sur les tests réalisés lors du **premier jour opérationnel** afin de suivre les tentatives d'authentification, la présence de données sensibles (toute donnée à caractère personnel, qu'elle soit ou non de santé, ou participant à la sécurité du système d'information constitué par le système seul ou le système d'information auquel il participe) et le format des événements. Ces logs doivent être transmis le plus rapidement à l'auditeur.
- **L'éditeur doit informer l'auditeur** de **toutes les vulnérabilités connues**, ainsi que des **mesures de sécurité mises en œuvre** pour les contourner.

L'ensemble de ces informations doivent être transmises dès la phase de cadrage afin de faciliter le travail de l'auditeur et limiter les échanges lors de la phase de tests qui pourraient occasionner un dépassement des délais prévus pour la prestation.

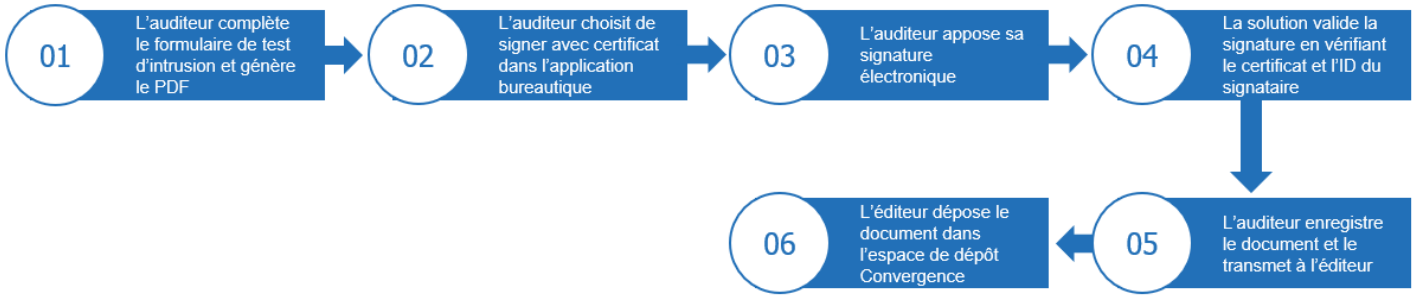
Annexe 3 : Synthèse de rappel pour les auditeurs

- Le test d'intrusion doit être effectué par un prestataire d'audit de la sécurité des systèmes d'information qualifié (PASSI). Cependant il ne s'agit pas d'un audit PASSI (la certification PASSI n'est pas requise pour l'auditeur et les conditions de réalisation du test d'intrusion ne sont celles d'un audit PASSI) ;
- Lors de la phase de cadrage, il est recommandé de prévoir une « clause de revoyure » avec l'auditeur afin de valider les corrections effectuées dans l'hypothèse où certaines exigences ne seraient pas validées lors du test d'intrusion ;
- Le formulaire atteste de la réalisation du test d'intrusion. Il doit être complété puis signé électroniquement par l'auditeur dans un format PDF qui peut être généré en suivant la macro du formulaire. La signature doit être approuvée par TSP (Trust Service Provider) via un certificat de signature ou via une plateforme de signature électronique. Ce document à transmettre à l'éditeur constitue la preuve que celui-ci devra fournir dans le cadre de la candidature au référencement Ségur ;
- Uniquement deux onglets du formulaire doivent être obligatoirement remplis par les auditeurs : l'onglet « Base commune » ainsi que l'onglet correspondant au type de l'application auditée ;
- Un formulaire doit être réalisé par type d'application ou type d'environnement ;
- Pour chaque point de contrôle, la conformité de l'application doit être évaluée et consignée dans le formulaire ;
- Les notes « N/A » et « Non » à l'évaluation de conformité de l'application nécessite impérativement une justification dans la colonne commentaire ;
- Il est essentiel de s'assurer que les informations suivantes ont bien été transmises par l'éditeur en amont de la phase de tests :

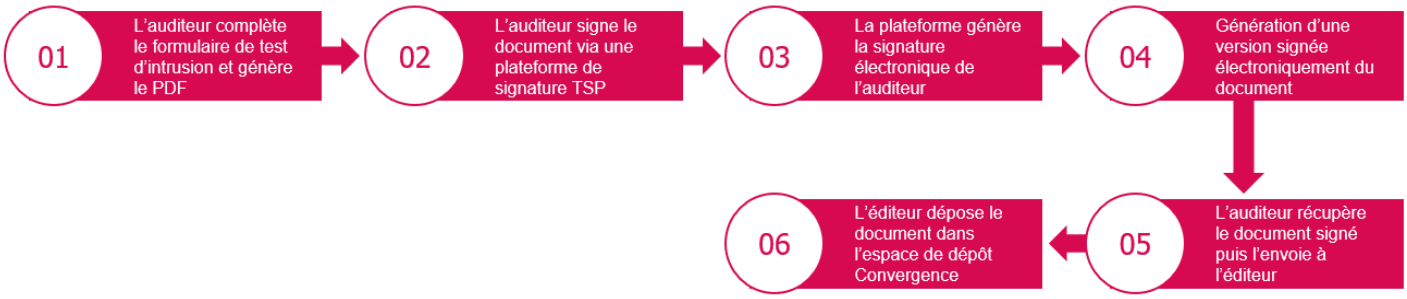
L'ensemble de ces informations doivent être transmises dès la phase de cadrage afin de faciliter le travail de l'auditeur et limiter les échanges lors de la phase de tests qui pourraient occasionner un dépassement des délais prévus pour la prestation.

Annexe 4 : Processus autour de la signature électronique

Signature électronique avec certificat délivré par un TSP



Signature électronique via une plateforme de signature



Annexe 5 : Vulnérabilités de l'OWASP et mapping des règles de sécurité du formulaire

La majorité des points à contrôler sont basés sur le référentiel du Top 10 de l'**Open Web Application Security Project - OWASP** (cf. Annexe 1 : Définition et concepts généraux). Ainsi, la description et les liens des principales vulnérabilités de l'OWASP et du formulaire du test d'intrusion se retrouvent dans le tableau ci-dessous :

Top 10 des risques de sécurité	Descriptions des vulnérabilités	ID Base commune	ID application Web	ID application mobile	ID client lourd 3Tiers ou plus	ID Client Lourd < 3Tiers
01. Contrôles d'accès défaillants	Failles de sécurité sur les droits des utilisateurs. Les attaquants peuvent exploiter ces vulnérabilités pour accéder à d'autres utilisateurs, données, ou exécuter une fonctionnalité métier en dehors des limites de l'utilisateur authentifié. **Exemple de vulnérabilités :** *CWE-35 Path Traversal, ex. CVE-2023-31179* *CWE-352 Cross-Site Request Forgery (CSRF), ex. CVE-2023-30525* *CWE-601 URL Redirection to Untrusted Site ('Open Redirect'), ex. CVE-2022-29272*	C8	W1, W8, W21, W22	M16, M18, M19	T6, T9, T10, T11, T14, T15, T16, T17	D5, D8, D9, D10, D11, NA1, NA2, NA3
02. Défaillances cryptographiques	Vulnérabilités liées au chiffrement (flux réseaux, base de données, informations sensibles, etc.) et à ses erreurs. Elles résultent d'algorithmes / de protocoles désuets ou faibles, d'une mauvaise gestion des clés de chiffrements, de l'absence de fonction de hash ou de mauvaise validation des certificats. **Exemple de vulnérabilités :** *CWE-326 Inadequate Encryption Strength, ex. CVE-2023-21443* *CWE-331 Insufficient Entropy, ex. CVE-2022-33756* *CWE-523 Unprotected Transport of Credentials, ex. CVE-2022-31805*	C16	W18	M10	X	X
03. Injection	Vulnérabilités liées à l'injection de données non fiables dans l'application pouvant mener à l'exécution de commandes. Les attaquants introduisent des données malveillantes et peuvent en conséquence, voir, modifier et supprimer des données métiers ou nécessaires au bon fonctionnement d'une application et même obtenir le contrôle du serveur. **Exemple de vulnérabilités :** *CWE-20 Improper Input Validation CVE-2023-31047* *CWE-79 Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting'), ex. CVE-2023-31807* *CWE-89 Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection'), ex. CVE-2023-31038*	C18	W23, W24	M20, M21	T15, T17, T12, T13	D9, D11, NA4, NA5

04. Conception non sécurisée	Défauts au niveau de la conception de l'application et de l'architecture logicielle (architecture en 2-tiers, accès direct à une base de données, serveur exposé, etc.) L'absence de prise en compte des risques commerciaux liés à un logiciel ou à un système en cours de développement, ainsi que l'incapacité à déterminer le niveau de sécurité requis, sont des facteurs qui peuvent contribuer à une conception non sécurisée. **Exemple de vulnérabilités :** *CWE-313 Cleartext Storage in a File or on Diskv, ex. CVE-2023-0114* *CWE-434 Unrestricted Upload of File with Dangerous Type, ex. CVE-2023-30266* *CWE-522 Insufficiently Protected Credentials, ex. CVE-2023-30776*	C14, C15	W2, W17	M3, M4, M5, M6, M7 M11	T1, T2, T5, T22	D1, D2, D15
05. Mauvaise configuration de sécurité	Failles liées à une mauvaise configuration sécurisée des serveurs, applications, base de données ou Framework, etc. Les failles les plus courantes sont le manque de durcissement, des autorisations mal configurées, des fonctions superflues tolérées ou installées, des mots de passe par défaut ou l'absence de mises à jour. **Exemple de vulnérabilités :** *CWE-260 Password in Configuration File, ex. CVE-2016-7043* *CWE-547 Use of Hard-coded, Security-relevant Constants, ex. CVE-2023-1712* *CWE-611 Improper Restriction of XML External Entity Reference, ex. CVE-2023-28680*	C9	W5, W6, W8, W9, W10, W11, W12, W13, W14, W15, W16	M1 M9, M12, M13, M14, M15, M16	T7, T8, T15, T18, T19, T21	D6, D7, D9, D12, D13, D14
06. Composants vulnérables et obsolètes	Présence de logiciels vulnérables qui ne sont plus supportés / obsolètes. Les configurations des composants ou des serveurs ne sont pas sécurisés (mise à jour des versions, présence de vulnérabilité non patchées, etc.) **Exemple de vulnérabilités:** *CWE-1035 Using Components with Known Vulnerabilities* *CWE-1104 Use of Unmaintained Third-Party Components*	C14, C15	X	M2	X	X
07. Identification et authentification de mauvaise qualité	Erreurs d'authentification en raison de problématiques de conception ou d'implémentation lors de la confirmation de l'identité, de l'authentification et de la session de l'utilisateur Les failles les plus courantes sont la présence de mot de passe faibles ou stockés en clair / faiblement chiffrés, des sessions d'utilisateurs qui ne sont pas invalidées correctement ou des identifiants de session exposés, etc. **Exemple de vulnérabilités :** *CWE-287 Improper Authentication, ex. CVE-2023-32243* *CWE-521 Weak Password Requirements, ex. CVE-2023-22451* *CWE-613 Insufficient Session Expiration, ex. CVE-2023-28003*	C2, C3, C4, C5, C6, C7	W3, W4	X	T14	D10
08. Manque d'intégrité des données et du logiciel	Vulnérabilités concernant les mises à jour logiciel, les données critiques et les pipelines CI/CD dont l'intégrité n'est pas vérifiée (plug-ins, bibliothèques ou modules issus de sources non vérifiables ou non fiables). Les fonctions de mises à jour automatiques des applications sans vérification d'intégrité font aussi partie de ces failles. **Exemple de vulnérabilités:** *CWE-353 Missing Support for Integrity Check, ex. CVE-2021-28546*	X	W19	X	X	X

	CWE-494 Download of Code Without Integrity Check, ex. CVE-2023-22635 *CWE-502 Deserialization of Untrusted Data, ex. CVE-2023-26779*					
09. Carence des systèmes de contrôle et de journalisation	Journalisation et de surveillance insuffisantes des événements telles que les traces d'audit, les alertes / erreurs générées, les journaux liés à l'authentification, etc. **Exemple de vulnérabilités :** *CWE-117 Improper Output Neutralization for Logs, ex. CVE-2020-4072* *CWE-532 Insertion of Sensitive Information into Log File, ex. CVE-2023-31207* *CWE-778 Insufficient Logging, ex. CVE-2021-32680*	C10, C11, C12, C13	X	M8	X	X
10. Falsification de requête côté serveur	Une faille SSRF se produit lorsqu'une application web récupère une ressource distante sans valider l'URL fournie par l'utilisateur. Elle permet à un attaquant de contraindre l'application à envoyer une requête élaborée à une destination inattendue, même si elle est protégée par un pare-feu, un VPN ou un autre type de liste de contrôle d'accès au réseau (ACL). **Exemple de vulnérabilité :** *CWE-918: Server-Side Request Forgery (SSRF), ex. CVE-2023-27161*	X	W7	X	T16	D10