

Exigences de sécurité applicables aux applications et services web

Direction des Systèmes d'Information

Émetteur

Date	Indice	Suivi de modifications	Rédaction	Relecture
16/09/2022	1	Version initiale	RSSI	RSOT
08/11/2023	2	Ajout spécificités API	RSSI	RSOT
04/01/2024	3	Ajout règle B18	RSSI	RSOT
12/02/2024	4	Nouveau nom d'entreprise	RSSI	RSOT

Sommaire

1. Exigences de sécurité	2
1.1. Chiffrement des flux et gestion des certificats	2
1.2. Développement sécurisé	3
1.3. Emails	4
1.4. Nom de domaine et hébergement	4
1.5. Lutte contre les intrusions	5
1.6. Lutte contre les attaques	5
1.7. Administration et gestion des accès	5
1.8. Traçabilité des accès et supervision	6
1.9. Sauvegardes	6
1.10. Plan de reprise informatique	6
1.11. Sécurité des interfaces de programmation d'applications (API)	6

1. Exigences de sécurité

À noter : Les règles portant la mention « EXT » sont applicables dans le cas où l'hébergement du site ou de l'application n'est pas effectué sur le système d'information de la SGP. Lorsqu'il s'agit d'un hébergement sur le système d'information géré par la SGP, la conformité à ces règles est habituellement couverte par notre infogérant et nos briques de sécurité (ex : pare-feu applicatif).

1.1. Chiffrement des flux et gestion des certificats

- **A1_EXT :** Privilégier TLS 1.3 et accepter TLS 1.2
La version TLS 1.3 doit être prise en charge et privilégiée. La version TLS 1.2 est également acceptée sous condition de suivre les exigences de ce guide.
- **A2_EXT :** Ne pas utiliser SSLv2, SSLv3, TLS 1.0 et TLS 1.1
Les versions SSLv2, SSLv3, TLS 1.0 et TLS 1.1 sont interdites.
- **A3_EXT :** Authentifier le serveur à l'échange de clé
Au cours d'un échange de clé, le serveur doit être authentifié par le client. Les alternatives anonymisées de ces échanges ou reposant sur l'utilisation de certificat brut définies dans la RFC 7250 sont proscrites.
- **A4_EXT :** Échanger les clés en assurant toujours la PFS
La propriété de confidentialité persistante doit être assurée (PFS). Il faut pour cela employer une suite cryptographique reposant sur un échange Diffie–Hellman éphémère (ECDHE ou, à défaut, DHE).
- **A5_EXT :** Utiliser SHA-2 comme fonction de hachage
Les fonctions de hachage de la famille SHA-2 doivent être utilisées.
- **A6_EXT :** Préférer l'ordre de suites du serveur
L'ordre des suites cryptographiques qui figure dans sa configuration du serveur doit prévaloir sur l'ordre des suites signalées par les clients.
- **A7_EXT :** Ne pas utiliser la compression TLS
L'utilisation du mécanisme de compression TLS est à proscrire.
- **A8_EXT :** Ne pas transmettre de données 0-RTT
Le serveur ne doit pas accepter les données 0-RTT lorsqu'il en reçoit.
- **A9_EXT :** Présenter un certificat signé avec SHA-2
La fonction de hachage utilisée pour la signature du certificat doit faire partie de la famille SHA-2.
- **A10_EXT :** Utiliser des clés de taille suffisante
Pour une protection des communications, les clés RSA doivent avoir une taille minimale de 2048 bits, et les clés ECDSA doivent avoir une taille minimale de 256 bits.
- **A11_EXT :** Réserver chaque certificat à une seule terminaison TLS
Un même certificat d'authentification ne doit pas être utilisé par plus d'une seule terminaison TLS (interdiction d'utilisation de certificat *wildcard*).
- **A12_EXT :** Transmettre une chaîne de certificats ordonnée et complète
Les chaînes de certificats transmises à l'aide des messages Certificate doivent être ordonnées et complètes.
- **A13_EXT :** Recourir à un processus sécurisé de fourniture des certificats
Pour les certificats ayant une validité supérieure à 90 jours, ceux-ci doivent être fournis par la Direction des Systèmes d'Information de la SGP qui les obtient auprès de l'autorité de certification habituelle.

En synthèse, les recommandations de l'ANSSI doivent être respectées (cf. <https://www.ssi.gouv.fr/guide/recommandations-de-securite-relatives-a-tls/>) et une note A+ doit être obtenue à la suite d'un test effectué sur <https://www.ssllabs.com/ssltest/>

1.2. Développement sécurisé

- **B1 : Mettre en œuvre HSTS**
Il est nécessaire de mettre en œuvre HSTS avec une durée de 2 an afin de limiter les risques d'attaque de type Man-In-The-Middle dus à des accès non sécurisés générés par les utilisateurs ou par un attaquant.
- **B2 : Vérifier l'échappement des contenus inclus**
Les données externes employées dans quelque partie que ce soit de la réponse envoyée au navigateur doivent avoir fait l'objet d'un « échappement » adapté au contexte d'interprétation.
- **B3 : Vérifier la conformité des données issues de sources externes**
Il est nécessaire de vérifier, chaque fois que c'est possible, que les données ont bien la forme attendue. Lorsque cela est possible, une approche par liste d'autorisations est recommandée : par exemple une donnée censée être numérique ne doit être composée que de chiffres.
- **B4 : Proscrire l'usage de la fonction *eval()***
La fonction *eval* est dédiée à la transformation de chaîne de caractères en code JavaScript. L'usage de cette fonction doit être proscrit.
- **B5 : Restreindre les contenus aux ressources fiables**
Il est nécessaire de mettre en œuvre CSP par en-têtes HTTP afin de présenter aux navigateurs une liste des sites reconnus comme présentant des ressources fiables et ainsi contribuer au principe de moindre privilège en réduisant le risque potentiel de vulnérabilité XSS.
- **B6 : Interdire des contenus inline**
Les contraintes CSP ne doivent pas présenter les mots-clés suivants : *data:*, *'unsafe-eval'* ou *'unsafe-inline'*.
- **B7 : Définir la stratégie de construction de l'en-tête Referer**
Il est nécessaire de définir une stratégie de construction de l'en-tête de requête HTTP *Referer* au travers de l'en-tête de réponse HTTP *Referrer-Policy*.
La stratégie de construction de l'en-tête *Referer* par défaut ne doit pas être conservée et l'option *unsafe-url* ne doit pas être utilisée.
- **B8 : Ne pas stocker d'informations sensibles dans les cookies**
Dans le cadre de la défense en profondeur et à l'exception des jetons de session, il est recommandé de ne pas stocker des informations sensibles dans les cookies. Leur utilisation n'est souhaitable que pour le stockage temporaire d'informations de faible volume, pour lesquelles la perte ou la divulgation sera sans conséquence.
- **B9 : Proscrire l'accès en JavaScript à un cookie de session**
Pour un cookie de session, il est nécessaire de positionner l'attribut *HttpOnly*.
- **B10 : Limiter le transit des cookies aux flux sécurisés**
Dès lors que des cookies sont nécessaires et que le site ou l'application n'est accessible qu'en HTTPS, le flag *Secure* doit être utilisé.
- **B11 : Définir une stratégie stricte d'envoi des cookies en cross-site**
Dès qu'un cookie n'a pas de raison d'être émis lors de la navigation depuis un site web extérieur, définir l'attribut *SameSite* à *Strict*. Dans le cas contraire, utiliser la valeur *Lax* si le cookie n'autorise pas d'action privilégiée via la méthode HTTP GET.
- **B12 : Définir une stratégie stricte d'envoi des cookies de session en cross-site**
Pour un cookie de session, l'attribut *SameSite* doit être défini et ne doit pas être positionné à *None*.
- **B13 : Limiter les composants logiciels tiers**
La liste des composants applicatifs tiers employés doit être limitée au strict nécessaire. Les composants non nécessaires doivent faire l'objet d'une suppression. Si leur suppression n'est pas envisageable, il est recommandé de les désactiver.
- **B14 : Maintenir à jour les composants logiciels tiers utilisés**
Les composants applicatifs tiers employés doivent être recensés et maintenus à jour. Cela impose que les composants sélectionnés pour une production soient évalués sur leur pérennité lors des phases de conception et que les vulnérabilités publiées soient suivies pour chacun d'eux.

Exigences de sécurité
applicables aux
applications et services
web

- **B15 : Migration ou isolation des systèmes obsolètes**
L'ensemble des logiciels utilisés sur le système d'information est dans une version pour laquelle l'éditeur assure le support, et tenu à jour. En cas de défaillance du support, il convient d'en étudier l'impact et de prendre les mesures adaptées.
Les systèmes obsolètes restant, gardés volontairement pour assurer le maintien en condition opérationnelle d'un processus, sont isolés.
- **B16 : Minimiser la divulgation d'informations liées aux composants techniques**
Les versions des composants techniques utilisés ne doivent pas être rendu accessibles. Tout entête ou fichier permettant d'identifier un composant technique ou sa version (ex : header « Server », changelog) ne doit pas être divulgué.
- **B17 : Gestion des empreintes de mots de passe**
Lorsqu'une application doit vérifier elle-même les mots de passe renseignés, celle-ci met en œuvre des mesures comme le hachage et le salage permettant de se prémunir contre les attaques documentées : attaques par dictionnaire, attaques par tables arc-en-ciel, attaques par force brute, etc.
- **B18 : Gestion de l'indexation par les robots**
Lorsqu'il n'est pas souhaitable d'avoir le site et son contenu indexé par des robots, une directive conforme à la RFC 9309 doit être positionnée. Cette directive est formalisée par un fichier « robots.txt » accessible à la racine du site-web et dont le contenu est le suivant :

```
User-agent: *
Disallow: /
```

En synthèse, les recommandations de l'ANSSI doivent être respectées (cf. <https://www.ssi.gouv.fr/guide/recommandations-pour-la-securisation-des-sites-web/>)

1.3. Emails

- **C1_EXT : Chiffrer les emails en transport**
Si le service procède à l'envoi d'emails, un chiffrement opportuniste doit être proposé. Celui-ci doit s'appuyer sur les protocoles à l'état de l'art (STARTTLS en mode opportuniste avec chiffrement TLS 1.2 minimum).
- **C2_EXT : Signer les emails sortants**
Si le service procède à l'envoi d'emails, une signature DKIM à l'état de l'art doit être apposée (1024 bits minimum). L'enregistrement DNS associé doit être correctement déclaré.
- **C3_EXT : Permettre le contrôle de l'authenticité des emails**
Si le service procède à l'envoi d'emails, un enregistrement SPF doit être configuré de sorte que les l'authenticité des emails puisse être vérifiée. Cet enregistrement doit être configuré en « soft fail » ou « hard fail » et complété d'un contrôle DMARC « reject » ou « quarantine ».

1.4. Nom de domaine et hébergement

- **D1_EXT : Enregistrer les noms de domaines par la DSI de la SGP**
Tout nom de domaine réservé dans le cadre de l'implémentation d'un site ou service Internet doit être réservé par les services de la DSI de la SGP.
- **D2_EXT : Bloquer l'émission de certificats illégitimes**
Un enregistrement DNS CAA doit être positionné sur le nom de domaine afin d'éviter l'émission de certificats illégitimes.
- **D3_EXT : Héberger le service sur le territoire national**
Lorsque des données sensibles sont manipulées, l'hébergement doit être effectué sur le territoire national, cette exigence concerne les différents environnements utilisés (production, préproduction...).

1.5. Lutte contre les intrusions

- E1_EXT : Restreindre les ports en écoute
Seuls les ports 80 (http) et 443 (https) peuvent être ouverts en écoute sans restriction d'origine des flux. Aucun autre port ne doit être ouvert sans restriction d'origine (ex : base de données, FTP, SSH).
- E2_EXT : Rediriger ou bloquer le trafic non chiffré
Toute tentative connexion non chiffrée (http) doit être bloquée ou automatiquement redirigée vers un protocole chiffré (https).
- E3_EXT : Mettre en place un pare-feu applicatif (WAF)
Un outil tiers permettant d'intercepter les flux avant qu'ils arrivent au serveur web doit être configuré afin de détecter et bloquer les tentatives d'intrusions les plus fréquentes (top 10 OWASP).
- E4 : Utiliser des composants faisant l'objet d'un support et exempt de vulnérabilités
L'ensemble des logiciels et matériels utilisés dans le cadre du service est dans une version pour laquelle l'éditeur assure le support. De plus ces logiciels et matériels doivent être à jour en matière de correctifs de sécurité.

1.6. Lutte contre les attaques

- F1_EXT : Réduire le risque de déni de service
Un outil ou service doit être mis en place afin de lutter contre les attaques par déni de service.
- F2_EXT : Protection contre les codes malveillants
Des logiciels de protection contre les codes malveillants, appelés communément antivirus, sont installés sur les serveurs d'interconnexion, serveurs applicatifs et postes de travail utilisés pour l'administration.

1.7. Administration et gestion des accès

- G1 : Restreindre les interfaces d'administration
Les interfaces d'administrations, qu'elles s'appuient sur le protocole HTTPS ou non doivent être accessibles qu'après une authentification à deux facteurs. Lorsque cela n'est pas possible, elles ne doivent être accessible que depuis un tunnel VPN sécurisé ou une liste d'adresses IP validées.
- G2 : Utilisation d'identifiant de connexion nominatifs
Des comptes d'administration individuels sont attribués à chaque administrateur. Les comptes natifs d'administration ne sont pas utilisés pour les actions courantes d'administration et les secrets associés ne sont accessibles qu'à un nombre très restreint de personnes.
- G3 : Révision périodique des droits d'accès
Les droits d'accès sont révisés périodiquement, au moins tous les ans. Cette révision porte sur les liens entre les comptes, les droits d'accès associés et les ressources ou les fonctionnalités qui en font l'objet.
- G4 : Confidentialité des informations d'authentification
À l'exception des mots de passe initiaux, les mots de passe ne sont connus que de leurs utilisateurs désignés. Les mots de passe ne sont pas stockés en clair, ils ne transitent pas en clair sur le réseau.
- G5_EXT : Gestion du départ d'un administrateur
En cas de départ d'un administrateur, les comptes individuels dont il disposait sont immédiatement désactivés. Les éventuels mots de passe d'administration dont il avait connaissance sont changés (exemples : mots de passe des comptes fonctionnels, comptes génériques ou comptes de service utilisés dans le cadre des fonctions de l'administrateur).
- G6 : Caractéristiques des mots de passe
Les mots de passe des comptes permettant l'accès à toute ressource informatique non publique respectent les critères précisés ci-dessous :
 - Compte utilisateur : 8 caractères minimum (avec 3 types de caractères différents parmi : majuscule, minuscule, chiffre, caractère spécial). Blocage de l'accès après 4 tentatives successives en échec.
 - Compte administrateur : 15 caractères minimum (avec 4 types de caractères différents). Blocage automatique de l'accès après 3 tentatives en échec.
 - Compte de service : 25 caractères minimum (avec 4 types de caractères différents)

- G7 : Renforcer la sécurité des identifiants de connexion
Les interfaces d'administrations ne doivent être accessibles que depuis un tunnel VPN sécurisé ou une liste d'adresses IP préalablement validées, ou protégées par une authentification à deux facteurs.
- G8 : Identification, authentification et contrôle d'accès logique
L'accès à toute ressource non publique nécessite une identification et une authentification individuelle.

1.8. Traçabilité des accès et supervision

- H1 : Gérer un historique inaltérable des accès
Un journal des accès et tentatives d'accès doit être alimenté automatiquement. Ce journal doit être stocké d'une façon à ce qu'il ne puisse être altéré et doit couvrir l'ensemble des accès (utilisateur final, administrateur, interfaces...). Il doit indiquer a moins : adresse IP d'origine, URL accédée, horodatage, code de la réponse fournie par le serveur.
- H2 : Conserver un historique des accès
Le journal des accès doit être conservé au moins 1 an.

1.9. Sauvegardes

- I1_EXT : Sauvegarder les données
L'ensemble des données et de la configuration doit être sauvegardé quotidiennement.
- I2_EXT : Protéger les sauvegardes
Les sauvegardes doivent être protégées contre les altérations (physiques ou logiques). Elles doivent être stockées sur un site distant de plusieurs kilomètres.
- I3_EXT : Conserver les sauvegardes
Les sauvegardes doivent être conservées pendant au moins 1 mois.
- I4_EXT : Superviser les sauvegardes
Un suivi de la bonne exécution des sauvegardes doit être effectué au moins une fois par mois. Celui-ci doit permettre d'identifier toute anomalie dans la gestion des sauvegardes.
- I5_EXT : Tester les sauvegardes
Un test de restauration des sauvegardes doit être effectué avant la mise en service effective du site Internet puis au moins une fois par an.

1.10. Plan de reprise informatique

- J1_EXT : Implémenter et tester un plan de reprise informatique
Un plan de reprise informatique doit être implémenté et testé. Celui-ci doit permettre de rétablir les services en 72 heures à la suite d'un sinistre majeur sur le centre de données principal.

1.11. Sécurité des interfaces de programmation d'applications (API)

- K1 : Restriction des services et méthodes API strictement nécessaires (cf. SYS-02, EXP-01, RES-13)
De façon à réduire la surface d'attaque, les services de type API doivent être activés uniquement lorsque nécessaire.

Une attention particulière doit être portée concernant :

- Les services de type API activés par défaut lors de la mise en place d'un nouveau service ou application informatique alors qu'ils ne seraient pas nécessaires.
- Les méthodes de type API activées par défaut lors de la mise en place d'un nouveau service ou application informatique alors qu'elles ne seraient pas nécessaires (ex. verbes *PUT*, *POST* ou *DELETE* lorsque qu'il est seulement nécessaire d'avoir un accès en consultation).

- **K2 : Sécurité des comptes exploitant des services API (cf. ADM-12, CAC-01 à CAC-27)**
Un compte exploitant des services API est considéré comme un compte de service, ou un compte administrateur, ou un compte utilisateur en fonction de son usage. Cette caractérisation est définie selon la « Politique de gestion des autorisations et du contrôle d'accès logique » de la SGP, celle-ci définit les règles de sécurité applicables en fonction de chaque cas¹.

Une attention particulière doit être portée concernant :
 - L'attribution des privilèges strictement nécessaire à ces comptes (en matière de périmètre de visibilité et de privilèges accordés)
 - La gestion sécurisée de leur secret en transit et en stockage (qu'il s'agisse d'un *token*, d'un mot de passe ou d'une clé privée d'un certificat)
- **K3 : Vérifier la conformité des données issues de sources externes (cf. E3, B3, VIR-01)**
Les services implémentant des API doivent vérifier que toute donnée d'entrée possède bien la forme et le format attendus afin de se prémunir contre l'exploitation de failles de sécurité et l'introduction de code informatique malveillant.

Une attention particulière doit être portée concernant :
 - Les failles de sécurités fréquentes concernant les API selon l'OWASP²
 - Les entêtes de serveurs qui exposent les versions des composants techniques utilisés tels quel la version du serveur web de l'API
- **K4 : Protection de la description des services web de type API (cf. B16, CAC-02 et ISP-07)**
Tout document détaillant le fonctionnement d'un service web de type API ou ses composants internes ou sous-jacents doit être protégé de sorte afin que seules les personnes ayant besoin d'en connaître puissent y accéder après s'être authentifié.

Une attention particulière doit être portée concernant :
 - Les fichiers de description du service API (ex. WSLD) qui seraient rendus accessibles sans authentification préalable lorsque cela n'est pas nécessaire
 - Les entêtes de serveurs qui exposent des versions des composants techniques utilisés tels quel la version du serveur web de l'API
- **K5 : Implémentation sécurisée de modes d'authentification pour l'utilisation d'API (cf. EXP-03, RES-04, CAC-09)**
Les services consommateurs et fournisseurs d'API doivent recourir des protocoles d'authentification sécurisés à l'état de l'art. L'utilisation d'une méthode permettant le rejeu de l'authentification comme l'authentification basique ou « basic auth » est tolérée uniquement lorsqu'un mécanisme de mitigation est implémenté en complément du chiffrement des flux par un protocole à l'état de l'art : liste approuvée d'adresse(s) IP, authentification mutuelle mTLS ou expiration automatique du token après 30 jours ou moins.

Une attention particulière doit être portée concernant :
 - Les restrictions applicables aux secrets des comptes utilisés en fonction de leur typologie (nombre de tentatives infructueuses avant blocage automatique, complexité des secrets, algorithme de chiffrement utilisés, etc.)
- **K6 : Toute utilisation de passerelle API de type SaaS doit faire l'objet d'une analyse de risque avant sa mise en service (cf. EXT-04, RES-04, RES-05)**
Lorsque l'infrastructure informatique de la SGP doit être interconnectée à celle d'une société externe, une analyse des risques est nécessaire.

Une attention particulière doit être portée concernant :
 - Les impacts liés à l'indisponibilité de ce service
 - Les impacts liés à l'accès illégitime aux secrets stockés
 - Les requêtes API provenant de réseaux non maîtrisés à destination d'une application SGP hébergée dans une zone réseau non prévue pour être exposée sur Internet (en l'absence d'utilisation d'un système de rupture protocolaire, pare-feu applicatif ou *API gateway*)

¹ Cf. <https://sharing.oodrive.com/share-access/sharings/jkkQHLkJ.5qz6q5qv>

² Cf. <https://owasp.org/www-project-api-security/>