Date : 30 septembre 202620 octobre 2026
Version : 1.01
Nombre de pages : 80

# RETEX DU CERT-FR SUR LES VIOLATIONS DE DONNEES APPLICATIVES

## POURQUOI CE RETEX ?

### Contexte opérationnel

Sur la période de la fin d'année 2025 au début 2026 de multiples incidents sur des applications contenant des données personnelles et parfois sensibles[1] ont eu lieu. Ces incidents ont résulté en la divulgation d'importantes quantités d'informations. La plupart de ces incidents sont liés à des accès à des applications web opérant de grandes bases de données.

### Objectifs du document

Dans ce retour d'expérience, le CERT-FR se concentre sur les violations de données exploitées dans les applications web. Les autres formes de stockage et de manipulation de ces données ne sont pas considérées dans le présent document. Ce retour d'expérience n'a pas vocation à traiter de façon exhaustive de la sécurité des données ou tous les scénarios de menace possibles. Il vise à donner des clés pour la prévention de telles attaques qui ont de forts impacts potentiels pour les organisations et les individus.

L'accompagnement du CERT-FR sur cette typologie d'incidents de sécurité permet aujourd'hui de dégager, de manière non exhaustive, un ensemble de constats et de recommandations associées.

[1] L'expression « donnée sensible » est entendue au sens défini par la CNIL - https://www.cnil.fr/fr/definition/donnee-sensible

# Table des matières

Pourquoi ce RETEX ? ..... 1
Contexte opérationnel ..... 1
Objectifs du document ..... 1
Synthèse managériale ..... 5
Incidentologie des violations de données personnelles en 2025 ..... 6
Constats du CERT-FR ..... 6
Impacts et risques observés ..... 7
Terminologie ..... 8
Risques considérés dans le document ..... 9
Impacts ..... 9
Scénarios ..... 9
Au-delà du document ..... 10
Principes de sécurisation ..... 11
Principes applicables à toutes les applications web gérant des données personnelles ou sensibles ..... 11
Principe spécifique aux administrations ..... 12
Contexte technique considéré dans le document ..... 13
Mise en œuvre des principes de sécurisation ..... 15
1. Appliquer une politique de gestion des données ..... 15
2. Minimiser les accès utilisateur ..... 17
3. Limiter l'exposition de l'application ..... 20
4. Dépasser le login/mot de passe ..... 27
5. Tracer les actions des utilisateurs en étant capable de les attribuer à un utilisateur et une adresse réseau ..... 33
6. Sécuriser l'environnement d'hébergement de l'application ..... 37
7. La supervision des journaux doit permettre d'y détecter les anomalies ..... 42
8. Déclencher des réactions automatiques et des alertes en cas de situation anormale ..... 44
9. La sécurité de l'application doit être vérifiée ..... 47
10. Mettre en œuvre une démarche d'homologation pour chaque application conformément au RGS et à la PSSIE ..... 51
Références ..... 53
ANNEXES ..... 54
A. Traçabilité : propager et journaliser les adresses sources de connexion ..... 55
B. Traçabilité : configurer la journalisation des adresses sources de connexion ..... 56

C. Contrôle d'accès : limiter les accès sur la base d'adresses IPs sources identifiées ..... 61

D. Contrôle d'accès : limiter les accès par géo-filtrage ..... 64

E. Contrôle d'accès : limiter les heures de connexion ..... 73

F. Limitation d'exfiltration : nombre de requêtes sur une fenêtre temporelle glissante ..... 76

G. Supervision : détection par nombre de tentatives d'accès ..... 78

**Attention :** Sauf disposition réglementaire contraire, les recommandations du présent document n'ont pas de caractère normatif. Elles sont livrées en l'état et adaptées aux menaces au jour de leur publication. Au regard de la diversité des systèmes d'information, le CERT-FR ne peut garantir que ces informations puissent être reprises sans adaptation sur les systèmes d'information cibles. Dans tous les cas, la pertinence de l'implémentation des éléments proposés par le CERT-FR doit être soumise, au préalable, à la validation de l'administrateur du système et/ou des personnes en charge de la sécurité des systèmes d'information.

Par ailleurs, plusieurs produits sont mentionnés dans le présent document. Ils le sont sur la base des configurations rencontrées par le CERT-FR dans ses activités de réponse à incident. Ces mentions ne constituent ni un jugement ni une recommandation d'un éditeur ou un produit.

# SYNTHESE MANAGERIALE

Ces dernières années, le CERT-FR est intervenu sur un nombre important de violations de données issues d'applications web.

Dans la plupart des cas, ces exfiltrations résultent d'intrusions sans exploitation de vulnérabilité technique : réutilisation d'identifiants légitimes, accès obtenus par ingénierie sociale ou, simplement, exposition des données sans contrôle d'accès.

Le plus souvent, lors des assistances aux victimes, les équipes du CERT-FR ont pu mettre en œuvre des mesures de réduction des risques de récidive. Ces mesures ne sont pas un substitut à une conception *Privacy by Design* telle que prônée par la CNIL. Elles permettent néanmoins de continuer à opérer des applications existantes, rarement prévues pour les niveaux d'hostilité rencontrés sur les systèmes d'information modernes, jusqu'à leur remplacement par des systèmes à l'état de l'art.

Le présent document scinde ces préconisations en 10 catégories :

1) Appliquer une politique de gestion des données
2) Limiter les accès utilisateurs
3) Limiter l'exposition
4) Sortir du *login*/mot de passe
5) Sécuriser l'environnement d'hébergement de l'application
6) Tracer les actions des utilisateurs
7) Superviser les journaux pour des traces d'anomalies
8) Déclencher des réactions automatiques et des alertes en cas de situation anormale
9) Vérifier la sécurité des applications
10) Pour les administrations, mettre en œuvre une démarche d'homologation pour chaque application conformément au Référentiel général de sécurité (RGS) et à la Politique de sécurité des systèmes d'information de l'Etat (PSSIE).

Les annexes fournissent des exemples de mise en œuvre de ces recommandations sur les briques technologiques que le CERT-FR a été amené à rencontrer.

Au travers de ce document, le CERT-FR souhaite non seulement montrer qu'un grand nombre de compromissions sont évitables mais aussi qu'il est possible, y compris sur des applications historiques, de drastiquement réduire les risques de compromission de données, sans attendre un éventuel remplacement.

# INCIDENTOLOGIE DES VIOLATIONS DE DONNEES PERSONNELLES EN 2025

## Constats du CERT-FR

En 2025, le CERT-FR a été informé de 196 incidents relatifs à des exfiltrations de données. En comparaison, 130 incidents de ce type avaient été traités par le CERT-FR en 2024. Le CERT-FR ne constate pas de **ciblage particulier** : tout type de secteur et d'entité peut être victime de ce type d'incident.

Dans de nombreux cas, ces exfiltrations ont été permises par des actions attaquantes présentant un **faible niveau de complexité**.

- Une simple **compromission de comptes utilisateurs** peut suffire. En mai 2025, une entité proposant des services de *e-santé* a ainsi été victime de l'extraction d'une partie de ses données – personnelles et médicales. En prenant le contrôle d'un compte utilisateur du service, l'attaquant a pu avoir accès à une API requêtant une base de données de l'entité, et ainsi en extraire le contenu.
- Également, certaines exfiltrations sont la conséquence d'une **exploitation de processus d'accès à des données peu sécurisés**. Le CERT-FR a traité en 2025 une exfiltration massive de données d'un applicatif dans lequel les accès via une API étaient insuffisamment limités. A la suite de la compromission d'un compte utilisateur, l'attaquant a consulté et extrait via l'API un ensemble de données auxquelles l'utilisateur n'avait pas légitimement accès.

Le CERT-FR a également constaté un recours récurrent aux ***infostealers*** dans les attaques menant à des violations de données personnelles d'ampleur. Un ensemble d'entreprises du secteur de l'agroalimentaire ont ainsi été victimes en février 2025 de violations de données, dues à la présence d'*infostealers* sur de nombreux postes de travail de leurs employés. Le déploiement de ce maliciel est la conséquence de l'installation par des employés d'un logiciel gratuit se présentant comme un convertisseur de documents Word en format PDF. Ce logiciel exfiltre tous les documents à convertir téléversés par les utilisateurs, tout en générant des documents dont l'ouverture provoque l'installation du maliciel, permettant ainsi sa propagation.

| Infostealer |
| --- |
| Un *infostealer* est un code malveillant conçu pour collecter des informations sur le poste de travail de la victime, notamment des identifiants enregistrés dans les navigateurs Web, des cookies de session, etc.<br>Les *infostealers* sont installés après compromission d'un poste, souvent dans le cadre de campagnes de propagation de botnets.<br><br>Ces codes malveillants sont généralement opérés par des acteurs spécialisés qui vont revendre les secrets à d'autres qui les exploiteront.<br>Les données peuvent être vendues comme des *instantanés* figés dans le temps, ou sous forme d'abonnement régulièrement rafraîchit. |

Outre ces campagnes d'attaques affectant des postes professionnels, l'infection par un *infostealer* de postes personnels de particuliers peut mener à des vols d'authentifiants. Ceux-ci sont par la suite utilisés pour se connecter illégitimement à leurs comptes sur divers applicatifs et en exfiltrer des informations, en particulier les données personnelles. Le CERT-FR a traité sur la fin d'année 2025 plusieurs exfiltrations de données issues de services publics permises par la présence d'*infostealer* sur des postes personnels d'usagers.

A noter, l'exfiltration de données peut également être rendue possible par la **compromission d'un prestataire** de la cible. L'ANSSI a traité en 2025 l'exfiltration de résultats d'analyses médicales d'un laboratoire, permise par la compromission du compte d'un prestataire destiné à gérer une application de visualisation des résultats d'examens médicaux. Ce type d'incident peut entraîner un préjudice important pour la victime finale, sans qu'elle n'ait de capacité d'action sur l'infrastructure compromise.

Ces actions malveillantes sont souvent **publiquement revendiquées**, notamment sur des forums cybercriminels ou sur la messagerie TELEGRAM, et les données publiées – en intégralité ou en partie. De telles revendications peuvent ne **pas être avérées** ou correspondre à des **republications** de données issues d'autres incidents ou provenant de **sources multiples**. Cela complexifie la levée de doute, qui est néanmoins toujours nécessaire. Les données prétendument exfiltrées doivent être qualifiées par l'opérateur du traitement de données personnelles afin d' identifier la source et évaluer l'impact de la divulgation.

## Impacts et risques observés

Préalablement à la mise en œuvre du Référentiel général sur la protection de la donnée(RGPD) seul un sous-ensemble restreint d'informations personnelles a été traité comme plus sensible par de nombreux opérateurs d'application. Les remplacements d'applications structurantes sont lents dans les organisations privées comme publiques. Les attaquants peuvent encore en 2026 trouver des applications où des mesures de sécurité, anciennes, ne sont pas proportionnées au niveau de la menace.

Par ailleurs, beaucoup d'organisations sous-estiment les risques d'usages malveillants d'identifiants légitimes, quand les attaquants récupèrent des couples de *login* et de mot de passe. Dans les incidents rencontrés par le CERT-FR, les compromissions viennent aussi bien des usagers des applications, légitimes ou non, que d'opérateurs techniques ou de sous-traitants.

La combinaison de ces deux facteurs a conduit à des compromissions de masse sur des données. Ces informations fusionnées et croisées entre elles constituent ainsi des fichiers très intrusifs.

Une fois récupérées par une personne mal intentionnée, ces données peuvent faire l'objet d'une commercialisation sur les marchés cybercriminels. Elles vont être exploitées dans le cadre de fraudes, ou encore être mises à disposition gratuitement dans le cadre d'opérations de déstabilisation.

## Terminologie

Le présent document utilise la définition de la CNIL de *Violation de données personnelles* pour évoquer un incident de sécurité causant la destruction, la perte, l'altération, la divulgation non autorisée de données à caractère personnel transmises, conservées ou traitées d'une autre manière, ou l'accès non autorisé à de telles données, de manière accidentelle ou illicite[2].

Cette définition est conforme au Règlement général sur la protection des données.

| COMMISSION NATIONALE DE L'INFORMATIQUE ET DES LIBERTES (CNIL) |
| --- |
| « Les **responsables** d'un fichier et les **sous-traitants** doivent prendre toutes les mesures nécessaires pour **assurer la sécurité** et la **confidentialité des données personnelles** qu'ils traitent :<br>• Des mesures de **sécurité physique** : sécurité des accès aux locaux ;<br>• Des mesures de **sécurité informatique** : antivirus, sécurisation des mots de passe, etc.<br>Ils doivent également veiller à ce que seuls les destinataires autorisés puissent accéder aux données. [3]» |

[2] Voir https://www.cnil.fr/fr/definition/violation-de-donnees

[3] Voir https://www.cnil.fr/fr/cnil-direct/question/les-mesures-de-securite-cest-quoi

# RISQUES CONSIDERES DANS LE DOCUMENT

Le présent document résume les leçons retenues par le CERT-FR à la suite du traitement d'incidents d'exfiltration ou d'accès à des données sensibles ou personnelles dans le contexte d'opération d'applications web.

Tous les risques possibles sur un système d'information ne sont pas pris en compte dans les recommandations du présent document. Seuls sont considérés les risques relatifs aux violations de données dans un environnement applicatif.

En particulier, les risques de divulgations de données lors de leur manipulation au sein d'un système d'information bureautique ou de développement ne sont pas considérés dans le présent document.

## Impacts

Les deux principaux types d'impacts considérés sont :

- Les divulgations unitaires de données à fort enjeu : dont la révélation peut à elle seule occasionner des préjudices importants (coordonnées de possesseurs d'objets de valeur, informations bancaires, informations de santé, intervenant d'une procédure judiciaire, informations familiales de professions à risque…) ;
- Les violations de masse de données personnelles : pouvant occasionner des dommages moins importants, ou moins immédiats mais à un grand nombre de personnes. Ces divulgations de masse viennent nourrir des campagnes de fraudes appuyées sur des informations précises et personnalisées.

## Scénarios

Les scénarios de compromission suivants sont considérés dans l'accès illégitime aux données :

- Usurpation d'identifiants : des comptes et secrets utilisateurs légitimes sont utilisés par l'attaquant. Ceux-ci peuvent être obtenus par réutilisation, voire mutation de divulgations publiques, par attaques basées sur l'ingénierie sociale, ou à la suite de la compromission d'un poste utilisateur et l'installation d'un *infostealer*.
- Accès direct aux données : une erreur de configuration expose des accès aux services ou aux données sans passer par l'application. Cet accès contourne ainsi toutes les mesures de sécurité applicative. Ces cas surviennent par exposition de ports réseau, par défaut de contrôle d'accès sur des stockages (S3 par exemple), sur le transit de sauvegardes dans des espaces non sécurisés… Ces erreurs de configuration sont particulièrement critiques dans les environnements infonuagiques où certains services de stockage sont par défaut accessibles à tout Internet.
- Accès par un collaborateur interne malveillant (*insider*) : plusieurs incidents récents ont vu des utilisateurs légitimes être la source de divulgation illégitime de données. Ils peuvent être en conflit avec l'organisation, ils peuvent agir par vénalité, mais peuvent aussi, parfois, faire l'objet de coercition de la part des attaquants. Dans certains cas, ce sont des utilisateurs légitimes ayant quitté l'organisation qui ont utilisé des comptes non révoqués après leur départ.
- Exploitation de vulnérabilité applicative : des erreurs dans l'application permettent de contourner les mesures de sécurité et d'accéder illégitimement aux informations. Dans

certains cas, ces vulnérabilités permettent de compromettre le serveur hébergeant l'application et ensuite d'extraire les données directement depuis leur stockage.

## Au-delà du document

Les scénarios de violation de données hors de l'application ne sont pas traités dans le présent document.

Néanmoins, les traitements hors application (en particulier sous-traités), les fichiers contenant des extractions partielles ou totales de bases de données, les données non effacées sur des supports mobiles constituent des sources fréquente de divulgations indésirées. Le cycle de vie des données et de leurs supports doit être étudié dans son entièreté. Les utilisateurs et les décideurs doivent être sensibilisés aux risques associés à la manipulation des données. Toutes ces politiques doivent être coordonnées avec le délégué à la protection des données (DPO) de l'organisation.

# PRINCIPES DE SECURISATION

L'expérience du CERT-FR l'amène à proposer une série de mesures qui visent à radicalement réduire les risques de violation de données personnelles depuis les applications et en particulier les risques de divulgations massives.

## Principes applicables à toutes les applications web gérant des données personnelles ou sensibles

1) Appliquer une politique de gestion des données
   a) Recenser les informations réglementées ou à fort enjeux
   b) Formaliser les flux de données
   c) Définir un plan de sécurisation
   d) Contrôler périodiquement la sécurité des informations à protéger
   e) Piloter avec la direction
2) Limiter les accès utilisateurs
   a) Appliquer un principe de moindre privilège sur les comptes utilisateurs
   b) Limiter la volumétrie accessible par session
   c) Appliquer des règles métier de limitation d'accès
3) Limiter l'exposition
   a) Distinguer les sources d'accès : usage interne, externe limité, ou public
   b) Limiter aux utilisateurs ayant strictement un besoin d'accès
   c) Ne considérer comme « interne » que le trafic venant d'un réseau appliquant votre Politique de sécurité des systèmes d'information (PSSI)
4) Dépasser le *login*/mot de passe
   a) Pour les utilisateurs « internes » : déployer de l'authentification forte
   b) Pour des utilisateurs « grand public » : privilégier des identifiants à usage unique et des *passkeys*
5) Sécuriser l'environnement d'hébergement de l'application
   Opérer dans un environnement conforme aux recommandations des guides de l'ANSSI
6) Tracer les actions des utilisateurs
   Être capable d'attribuer toute action dans l'application à un utilisateur et une adresse réseau au travers de tous les composants et sur toutes les parties de l'application
7) Superviser les journaux pour des traces d'anomalies
   Au minimum pratiquer des inspections périodiques
8) Déclencher des réactions automatiques et des alertes en cas de situation anormale
   a) Ralentir les tentatives d'authentification et déclencher des blocages temporaires
   b) Bloquer l'accès aux fonctions sensibles en cas d'ouverture de session « à risque »
9) Vérifier la sécurité des applications
   a) En amont de sa mise en production et dans la durée
   b) Analyser les risques, auditer, identifier les composants (SBOM), maintenir à jour face aux vulnérabilités

## Principe spécifique aux administrations

10) Mettre en œuvre une démarche d'homologation pour chaque application conformément au Référentiel général de sécurité[4] et à la Politique de sécurité des systèmes d'information de l'Etat[5]. En particulier, l'homologation implique de :
    a) Réaliser une analyse de risque prenant en compte la confidentialité des informations à protéger ;
    b) Sélectionner des mesures centrées sur la protection des données manipulées par l'application.

[4] Référentiel général de sécurité (RGS) : https://cyber.gouv.fr/reglementation/reglementation-identite-confiance-numerique/securite-echanges-voie-electronique/referentiel-general-de-securite/
[5] Politique de sécurité des systèmes d'information de l'Etat (PSSIE) : https://cyber.gouv.fr/reglementation/cybersecurite-systemes-dinformation/protection-du-service-numerique-de-letat/

# CONTEXTE TECHNIQUE CONSIDERE DANS LE DOCUMENT

Les recommandations suivantes proposent des pistes techniques et organisationnelles pour les mises en œuvre des principes de sécurisation énoncés précédemment.

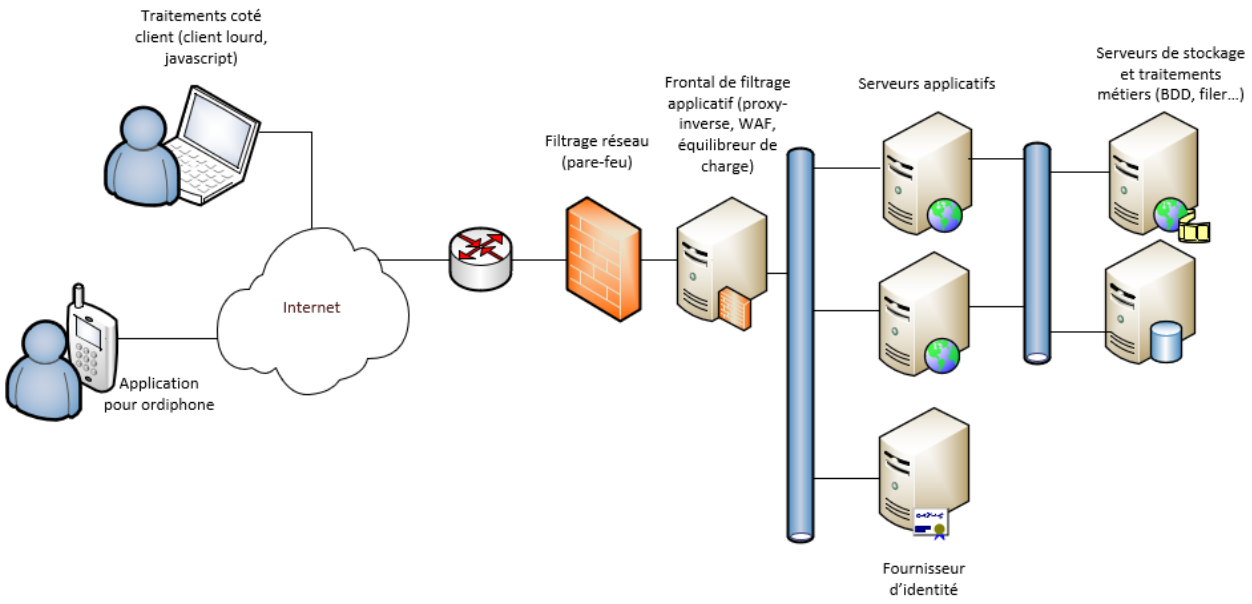



*Figure 1 - Architecture type considérée*

Les recommandations supposent une architecture d'application web moderne.

Dans ce type d'environnement, une couche de filtrage réseau (pare-feu) permet de contrôler l'exposition des services et les flux sortants. Ce filtrage est complété par une couche traitant les requêtes au niveau applicatif (proxy-inverse, WAF[6], équilibreur de charge). Derrière, un ou des serveurs applicatifs exécutent le code central de l'application et s'appuient sur des serveurs spécialisés dans le stockage ou des traitements spécifiques à un métier. Un fournisseur d'identité permet la gestion de l'authentification et des accès.

Il est notable que des points de contrôle similaires se retrouvent dans la plupart des infrastructures même virtualisée, opérée sur des services *cloud* d'*Infrastructure As A Service*, voire déployée à la demande dans des containers.

Quand une mesure de sécurité ne peut pas être appliquée à un point de l'architecture, des mesures compensatoires sont souvent possibles en amont ou en aval. En particulier, sans modifier substantiellement une application, il est courant d'introduire des fonctions de sécurité sur la couche

[6] *Web Application Firewall* (WAF) : ou pare-feu applicatif, équipement permettant le filtrage de requêtes web sur la base de leur contenu.

de filtrage applicative : traçabilité, contrôle d'accès, gestion d'identité moderne, mesures associées à TLS[7], etc.

[7] *Transport Layer Security* (TLS) est le protocole de chiffrement réseau le plus commun sur les applications web. Successeur de SSL, ce protocole permet de protéger les flux en confidentialité, en intégrité et d'authentifier le serveur, voire les deux parties de la communication.

# MISE EN ŒUVRE DES PRINCIPES DE SECURISATION

## 1. Appliquer une politique de gestion des données

Le cycle de vie des données dans l'application et son environnement d'opération doit être étudié et documenté. Des mesures appropriées doivent être mises en œuvre pour s'assurer que les données soient protégées dès qu'elles quittent leur environnement d'exploitation.

*Principes*

Recenser les informations à protéger

Conformément au RGPD[8], il convient d'identifier les données personnelles collectées et manipulées dans votre organisation et leurs supports techniques (bases de données, fichiers, sauvegardes, applications, canaux de communication). Suivant le contexte, des données non personnelles peuvent aussi être réglementées. Un travail similaire d'identification est alors nécessaire.

Formaliser les flux de données

Il convient d'identifier où l'information circule au sein des systèmes et réseaux de l'organisation, et sur ceux de tiers. Une attention particulière doit être portée aux accès provenant de clients hors de contrôle de votre organisation sur lesquels vous n'avez pas de visibilité.

Définir un plan de sécurisation

Sur la base des recensements des informations et des flux, il convient de définir des mesures de sécurité, techniques ou organisationnelles, assurant leur sécurisation tout au long de leur cycle de vie.

Contrôler périodiquement la sécurité des informations à protéger

Les mesures sélectionnées doivent être auditées périodiquement afin de vérifier leur bonne mise en œuvre et leur efficacité. Les écarts identifiés doivent faire l'objet de mesures correctives.

Piloter avec la direction

La direction de l'organisation est responsable et garante de la bonne protection des données personnelles. Elle doit être impliquée dans les choix de sécurité applicative.

| Cas d'accès direct aux données |
| --- |
| Le CERT-FR a eu connaissance de nombreux cas d'exposition involontaire de données, dus à une erreur de configuration ou à un manque de sécurisation. Une entreprise de solution de télétransmission de données médicales a ainsi exposé sur les sites qu'elle concevait des données médiales de patients, conséquence d'une erreur commise lors d'une intervention de maintenance sur un serveur. Les données, très facilement accessibles, ont pu être consultées ou téléchargées. |

*Mise en œuvre technique*

Parmi les mesures de mitigation des risques sur les données dans les applications, les suivantes sont particulièrement importantes :

[8] Règlement Général sur la Protection des Données (RGPD) : https://www.cnil.fr/fr/reglement-europeen-protection-donnees

### Limiter l'accès direct à la donnée

Aucun stockage de donnée ne devrait être accessible sans passer par l'application. Bases de données, stockages de fichiers, caches, doivent être hébergés dans des zones filtrées au niveau réseau sur la source de connexion.

### Systématiser les contrôles d'accès aux données

Tout accès aux informations protégées doit être authentifié et autorisé explicitement.

En particulier, il ne doit pas être possible d'accéder à un stockage de la donnée (serveur de fichier, stockage structuré type S3, base de données « NoSQL » ou relationnelle…) sans être identifié et autorisé par un contrôle d'accès.

### Limiter la circulation des données hors des environnements de production

En dehors des systèmes de production, les données utilisées devraient être simulées. L'utilisation d'une donnée réelle est à proscrire par défaut. Si néanmoins l'utilisation de donnée purement synthétique est impossible, la valeur de ces données doit être dégradée par des mesures de bruitage de l'information[9]. L'utilisation d'une méthode de pseudonymisation ou d'anonymisation doit faire l'objet d'un processus d'approbation spécifique.

### S'assurer que les flux réseau disposent d'une protection cryptographique adéquate

Les flux réseau doivent être protégés par des mesures cryptographiques à l'état de l'art[10] assurant le chiffrement, et l'authenticité[11]. Les clés assurant l'effectivité des fonctions cryptographiques doivent être gérées à l'état de l'art.

### Protéger les stockages de la donnée

Les supports de stockage des données (disques, bandes, clés USB) doivent être protégés contre l'extraction y compris après usage. Le cycle de vie de ces supports doit inclure soit une protection cryptographique soit des mesures destructives rendant les données irrécupérables.

Une mesure recommandée est le chiffrement systématique des disques par une clé gérée dans des composants cryptographiques matériels ou par un serveur de clé.

Ceci peut être effectué par l'usage de *LUKS* conjointement à *Clevis* pour lier le disque au TPM sous système Linux et un serveur *TANG* permet un déverrouillage à distance. Sous système d'exploitation Windows *Bitlocker* permet directement le chiffrement depuis des clés protégées par le TPM, et le service *Network-unlock* contrôle le déchiffrement à distance.

### Protéger en confidentialité et en intégrité les données dans les exports et sauvegardes

Quand la donnée quitte l'application, elle doit être chiffrée. En particulier, les sauvegardes qui par essence sont stockées hors de l'application, doivent être protégées par des protections cryptographiques à l'état de l'art. Ces fonctions, comme pour les flux, doivent être configurées et opérées à l'état de l'art.

### Appliquez une politique de conservation des données

La suppression des données qui ne sont plus requises pour le traitement limite les impacts des compromissions.

---

[9] Il existe un grand nombre de méthodes de dé-identification conservant les propriétés générales d'un corps de données : l'anonymisation rend les données non rapprochables d'une identité, la pseudonymisation masque cette identité par un alias. Voir https://www.cnil.fr/fr/technologies/lanonymisation-de-donnees-personnelles

[10] Voir https://messervices.cyber.gouv.fr/guides/mecanismes-cryptographiques

[11] Parmi les protocoles les plus courant assurant ces fonctions, il existe : TLS, IPSec, SSH, SMB chiffré

## 2. Minimiser les accès utilisateur

*Principes*

Limiter au strict nécessaire les accès des utilisateurs au niveau des données et de l'infrastructure en appliquant une vue métier sur le parcours utilisateur et les rôles attribués.

### Restreindre les accès aux données en fonction des profils métier

Le principe du moindre privilège doit être appliqué pour l'accès aux données : les utilisateurs doivent avoir des accès restreints à ceux nécessaires au bon exercice de leurs missions.

### Réguler les accès à l'application en fonction des profils métiers

Les accès des utilisateurs doivent être régulés, que ce soit en nombre de sessions actives concurrentes ou en rythme de requêtes. Cette régulation doit être différente en fonction du type d'accès au service : utilisateurs interactifs, applications tierces, clients lourds voire un agent IA.

### Restreindre les accès d'administration fonctionnelle

Les utilisateurs qui ne disposent pas de privilèges d'administration ne doivent pas pouvoir accéder aux zones ou adresses des fonctions d'administration de l'application.

### Limiter l'usage en heures non ouvrées

Dans la mesure du possible, restreindre les accès à l'application, hors des heures ouvrées de l'organisation (par exemple entre 23h et 5h). Définir des horaires en fonction des zones géographiques d'accès[12] .

### Gérer le cycle de vie des utilisateurs de l'application

La création et la modification de compte doivent suivre un processus systématique. Chaque compte doit pouvoir être rattaché à une personne ou une entité. Les comptes partagés sont à proscrire. Les besoins d'accès doivent être revus en fonction des changements de rôles du porteur du compte.

Les accès du compte sur l'application métier doivent pouvoir être désactivés, modifiés ou supprimés.

Un processus de revue périodique doit permettre de s'assurer que seuls les comptes légitimes sont actifs sur l'application.

[12] Voir *Contrôle d'accès : limiter les accès par géo-filtrage* et *Contrôle d'accès : limiter les heures de connexion* en annexe.

**Acceptabilité de la minimisation des accès**

Les accès dans un système d'information sont parfois perçus comme des privilèges, des marques de confiance. Il en découle qu'une politique de minimisation des accès peut être vécue comme une forme de défiance et de déclassement.

Le CERT-FR constate que l'usurpation d'identité par des attaquants est endémique. Dans la plupart des cas, ces usurpations d'identité se font sans complicité ou même faute significative de la victime. Mais l'usurpation d'une identité pour commettre des actes délictueux a des conséquences corrosives sur la confiance. Il est fréquent que la suspicion vienne peser durablement sur la victime de l'usurpation.

Il est donc important de considérer les restrictions d'accès comme une mesure protégeant autant le propriétaire des identifiants que l'organisation.

### *Mise en œuvre technique*

#### Gestion RBAC[13] des utilisateurs

- Des groupes métiers doivent être définis, au sein de l'application ou d'un annuaire externe sur lequel s'appuie l'application.
- A chaque groupe doit être associé l'ensemble strictement nécessaire des autorisations ou des restrictions d'accès aux données de l'application.
- Chaque utilisateur doit être associé aux groupes métiers lui correspondant. Un utilisateur qui ne fait pas partie d'un groupe spécifique ne doit pas pouvoir accéder aux données.

#### Limitation sur les sessions utilisateur

Les utilisateurs doivent avoir une limite concernant :

- Le nombre de sessions concurrentes autorisées sur l'application ;
- La durée des sessions utilisateur.

#### Limitation sur le rythme de requête

Mettre en place du *rate-limiting* adapté aux différents comptes utilisateur.

A l'exception de certains groupes spécifiques, la plupart des utilisateurs d'une application n'ont pas besoin de parcourir de grandes quantités d'entrées. A fortiori, parcourir un très grand nombre d'entrées en un temps court est complètement anormal.

L'idéal est de positionner des seuils de consultation internes à l'application. Mais faute d'une telle possibilité, il est recommandé de limiter la volumétrie sur une période donnée par session sur les équipements frontaux : pare-feu applicatif, ou serveur web.

#### Filtrage sur les accès administrateur

Configurer un filtrage pour restreindre l'accès aux pages web et accès API dédiés à l'administration fonctionnelle de l'application.

[13] *Role Based Access Control* (*RBAC*) : gestion de droits par les groupes ou les droits d'accès sont donnés à des groupes auxquels les utilisateurs sont rattachés plutôt que directement aux utilisateurs.

Ce filtrage peut être directement configuré au sein de l'application mais également mis en place au niveau des équipements de type proxy-inverse ou WAF. Il peut s'appuyer sur le compte utilisé ou sur l'adresse IP source de l'utilisateur si les administrateurs fonctionnels opèrent depuis un sous-réseau dédié.

### Restrictions horaires

Configurer si possible des règles de restriction horaires sur les comptes utilisés pour se connecter à l'application ou sur les équipements réseau en amont de l'application. Différentes méthodologies sont disponibles en annexe « *Configurer une application pour limiter les heures de connexion* ».

### Gestion des comptes inactifs

Il convient de mettre en place un mécanisme de détection et désactivation des comptes inactifs.

Un compte inutilisé pendant une période propre au métier doit être désactivé pour limiter les risques d'utilisation de comptes qui ne sont plus légitimes. Ceci peut être fait dans l'application, mais aussi au niveau des annuaires, voire au niveau d'un gestionnaire d'identité lors de l'ouverture de session.

Alternativement, un processus externe peut être mis en place.

Si l'horodatage de dernière connexion des utilisateurs n'est pas une information stockée au sein de l'application, elle peut potentiellement se déduire de l'analyse des journaux applicatifs ou de ceux de proxy-inverse / WAF en amont de l'application.

De la même façon, si la désactivation sans suppression d'un compte n'est pas possible au travers de l'application, les accès de ce dernier peuvent être filtrés via un serveur mandataire inverse ou un WAF en amont de l'application.

## 3. Limiter l'exposition de l'application

### Principes

Chaque application a une population d'usagers définie qui détermine son besoin d'exposition. Elles peuvent être à usage interne exclusivement, à usage externe restreint ou à usage public. L'exposition de l'application doit être limitée au minimum nécessaire à son bon fonctionnement. Les mesures de sécurité doivent renforcer cette limitation.

Par principe, il convient de considérer comme « interne » exclusivement le trafic originaire d'un réseau appliquant votre politique de sécurité des systèmes d'information (PSSI).

*Application à usage interne*

Les applications à usage interne à une organisation n'ont besoin d'être accédées que depuis des postes sur lesquels la politique de sécurité de l'organisation s'applique. Dans ce cas, l'accès à l'application ne doit pas être possible depuis Internet.

Il est impératif d'être sur un réseau interne à l'organisation ou transiter par un tunnel chiffré et authentifié (VPN) pour se connecter à de telles applications.

Préférez l'authentification des VPN sur la base d'une identité de machine, celle du poste utilisateur, plutôt que celle de l'utilisateur. Il est fortement conseillé de protéger cette identité par un composant de confiance : carte à puce, *token* cryptographique, TPM, composant de confiance de smartphone.

Exemple d'application : protéger par VPN une gestion de données opérée au sein d'une administration n'ayant pas à être accédé par des tiers.

*Application à usage externe limité*

Certaines applications doivent être accessibles entre plusieurs organisations, depuis des postes qui n'appliquent pas la politique de sécurité de l'organisation qui opèrent l'application.

L'accès à l'application ne doit pas être librement possible directement depuis Internet. Il ne devrait être possible de se connecter à l'application que depuis les réseaux internes ou depuis un VPN (site-à-site ou point-à-site) pour les partenaires.

Une limitation à une liste d'adresses IP sources autorisées est envisageable si les solutions de VPN n'ont pas pu être mises en place.

Ce contrôle moindre sur les sources de connexion doit être compensé par des mesures rigoureuses sur les limitations métier dans l'application.

Exemple d'application : applications sur lesquelles peuvent intervenir des membres de plusieurs organisations, par exemple dans un cadre de sous-traitance.

*Application à usage public*

Certaines applications ont un usage par essence public. Les usagers peuvent s'y connecter légitimement depuis n'importe quelle source.

L'accès depuis tout Internet à une application rend difficile la diminution du risque par le contrôle de la source.

Des mesures de réduction d'exposition sont tout de même possibles : filtrage sur la base de la source de connexion (géo-filtrage), ou de listes d'interdiction (nœud Tor, IP mal réputées …). Si bloquer sur le critère de source de connexion n'est pas possible, ce devrait être, au moins, un critère d'alerte pour la supervision.

Un filtrage applicatif devrait compléter ces mesures afin de s'assurer que ne sont exposées sur Internet que les fonctions nécessaires au grand public (par exemple, n'autoriser les accès au chemin « /admin » que depuis l'intérieur de l'organisation).

Exemple d'application : guichets en self-service ou le grand public peut maintenir ses propres informations.

## Mise en œuvre technique

### *Architecture de filtrage*

Le service d'une application ne devrait jamais être exposé directement à ses usagers.

Une architecture moderne permet de filtrer les flux à plusieurs niveaux de protocoles.

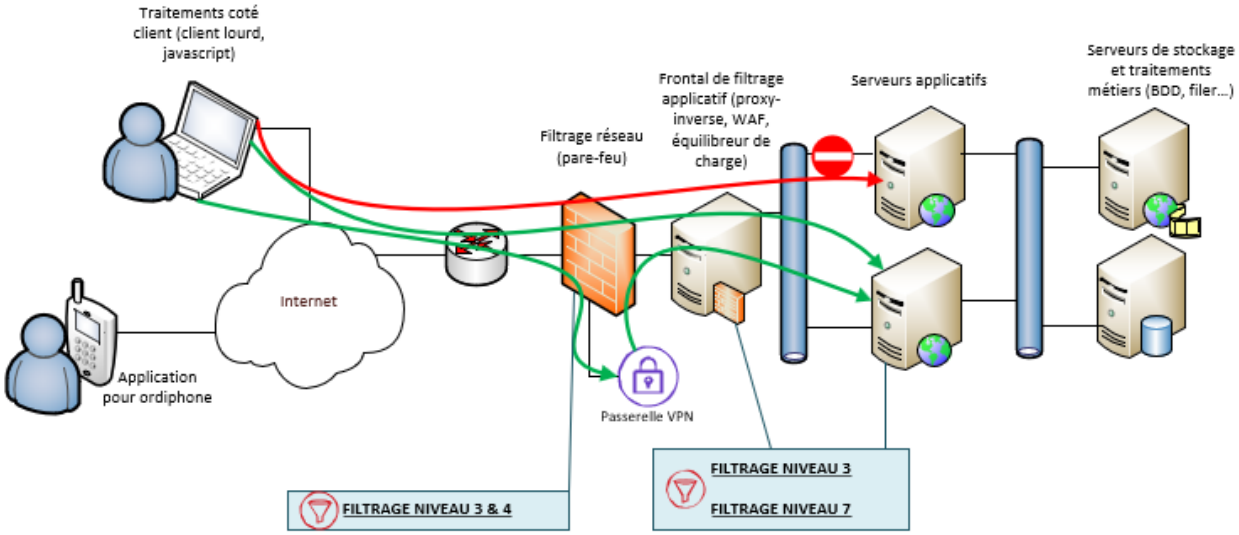



*Figure 2 - Points de filtrage autour d'une application WEB version simplifiée*

Celui-ci devrait, au moins, être derrière une fonction de filtrage réseau sur la base des adresses IP et ports (niveau 3 et 4), et une fonction de filtrage de niveau applicatif (niveau 7).

Le filtrage de niveaux 3 et 4 permet de s'assurer qu'aucun service indésirable ne soit accessible hors de la zone où s'exécute l'application. Le filtrage de niveau 7 permet d'appliquer des mesures de filtrage plus fines, sur la base des chemins ou ressources accédés.

Ces filtrages se complètent afin de fournir aux défenseurs une vue sur les tentatives de connexions anormales, mais aussi sur le trafic légitime.

Chaque composant du filtrage est une source de journaux qui doivent être collectés et gérés centralement.

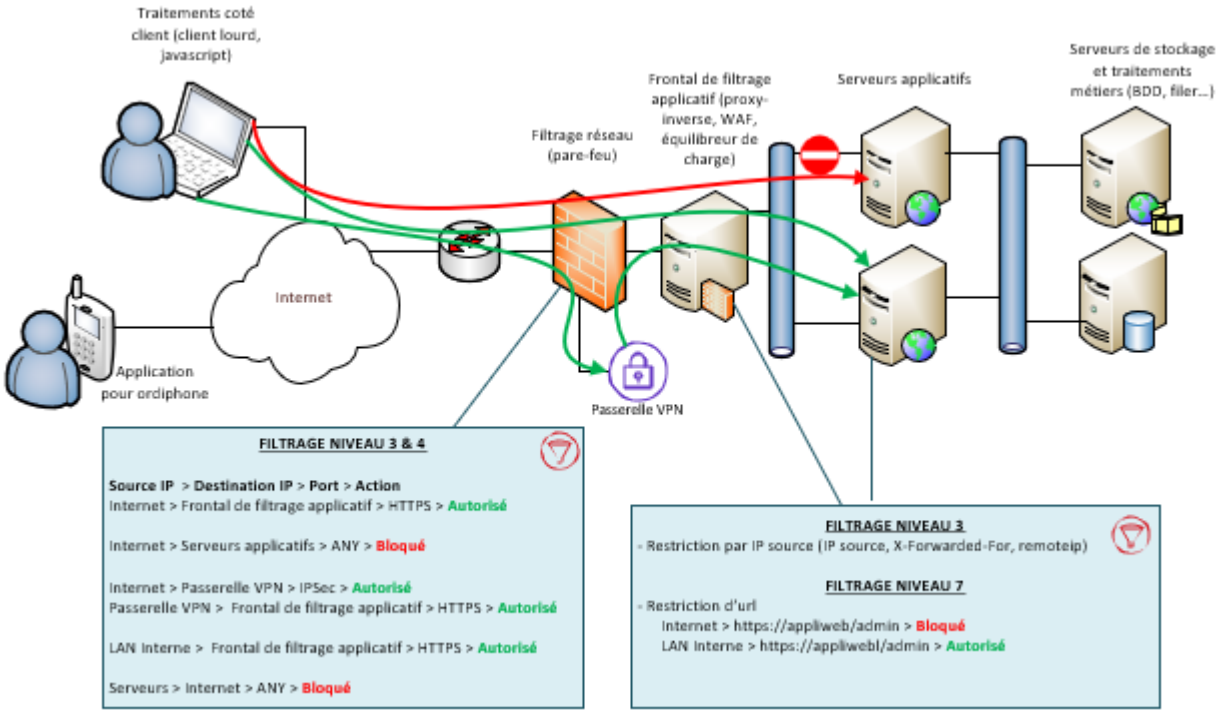



*Figure 3 - Exemple d'architecture montrant les fonctions de filtrage autour d'une application WEB*

**Préférer les distinctions de services par nom d'hôte que par chemin**

Dans les applications web, plusieurs fonctions distinctes de l'application sont accédées au travers d'Internet. Des services web peuvent être destinés à des partenaires, une zone « admin » peut servir à l'administration applicative…

Il est recommandé d'héberger les différents services sur des noms d'hôtes (éventuellement virtuels) distincts. Par exemple, il est préférable d'utiliser https://admin.example.com plutôt que https://www.example.com/admin.

Cette mesure permet d'exercer plus facilement des politiques de sécurité et de filtrage différentes pour des services distincts et minimise les risques d'exposition croisée sur les services portant le routage d'URLs (WAF, composant applicatif) et réduit les partages de données implicites dans le navigateur.

*Identifier et distinguer les flux non locaux*

Dans plusieurs incidents, le CERT-FR a pu constater l'usage d'adresses publiques en interne d'infrastructures applicatives. Cet usage ne crée pas directement un risque tant que les routages sont strictement segmentés. Mais il rend confus l'interprétation de flux observés et aggrave toute erreur de configuration des protocoles de routage.

Par ailleurs, l'usage d'adressage public en interne des organisations provoque des erreurs dans certains produits de sécurité informatique qui font des inférences sur la base des sous-réseaux IP utilisés.

Il est recommandé de respecter les plans d'adressages de la RFC 1918[14] qui spécifie les sous-réseaux IP utilisables sur des réseaux privés. Le respect de ce standard permet de bien différencier les flux locaux des flux externes.

Note : Pour changer rapidement un adressage public utilisé en interne vers un adressage privé, se contenter de changer le 1er octet en 10 est une solution rapide. Par exemple : 136.13.0.0/16 (réseau public) devient 10.13.0.0/16 (réseau privé).

*Maintenir l'information de la source de la connexion*

Il est important de configurer les applications manipulant les flux HTTP pour inclure les entêtes HTTP permettant de garder l'origine de la connexion (`X-Forwarded-for` par exemple[15]).

Il est aussi important d'intégrer dans les journaux des pare-feux, VPN et des équipements périmétriques l'adresse originale (champ `remoteip` par exemple).

C'est seulement en configurant la traçabilité de ces informations tout le long des connexions que toutes les actions peuvent être associées à une adresse IP.

*Limiter les actions à fort enjeux*

Il est recommandé de filtrer les actions à fort enjeux sur l'application depuis des réseaux non contrôlés si leurs interfaces y sont par défaut accessibles. Ce contrôle peut s'effectuer pour des URI spécifiques dans les configurations des WAF, dans les proxy inverses et autres équipements frontaux, mais aussi par un filtrage réseau de niveau sur la base des interfaces, des adresses IP et des ports.

Attention : Il faut aussi noter que les filtrages basés sur un chemin ou un paramètre HTTP sont fréquemment contournés. Activer des options de normalisation des valeurs permet de réduire les risques de contournement et est recommandé mais insuffisant pour assurer une sécurité dans la durée. Les pratiques de *virtual-patching* sur un WAF ne doivent être considérées que comme une façon de couvrir temporairement une vulnérabilité.

*Réduire l'exposition*

Les applications à usage interne ne doivent être accessibles que depuis un VPN avec authentification forte ou depuis un réseau interne. La mise en place de règle de filtrage sur le LAN du VPN Interne et pour les réseaux locaux peut se faire au niveau du pare-feu interne ainsi que les WAF, les serveurs mandataires inverses[16] et les équipements frontaux.

Les applications à usage externe limité ne doivent être accessibles que depuis les réseaux internes ayant le besoin d'accès (cf. point supra) et les réseaux des partenaires et prestataires externes spécifiques avec VPN (site-à-site et/ou point-à-site). Sinon un filtrage strict des adresses IP sources peut être mis en place en se basant sur des IP de sorties identifiées pour les partenaires et intervenants externes.

---

[14] RFC 1918 : https://www.rfc-editor.org/rfc/rfc1918

[15] Voir en annexe *Traçabilité : propager et journaliser les adresses sources de connexion*

[16] *Reverse Proxy*

Les applications à usage public ne doivent pas être accessibles depuis des réseaux qui ne sont pas considérés de « confiance ».

Il est fortement recommandé d'utiliser les listes présentes dans les pares-feux pour interdire les connexions depuis les réseaux de mauvaise réputation (domaines très récents, adresses IP associées à des activités malveillantes …).

En fonction des interactions métiers, il peut être envisagé de limiter aussi l'exposition à des adresses associées aux pays avec lesquels vous avez des interactions légitimes et interdire les autres. Il est également possible de limiter les accès depuis des sources d'anonymisation (VPN commerciaux, nœuds TOR)[17].

Au niveau des applications, il peut aussi être intéressant de compléter le filtrage sur la base de l'adresse IP par des restrictions sur la base des entêtes tels que le champ `user-agent`.

Il est intéressant de limiter le débit et le volume de données autorisé en téléchargement et en téléversement par session en fonction des usages métiers[18].

**Utilisation des valeurs contenues dans les entêtes HTTP**

Les valeurs transmises dans les entêtes HTTP sont sous le contrôle de l'usager. Donc, dans le cas d'une attaque, elles peuvent être falsifiées et le sont couramment.

Il existe des méthodes de pilotage de navigateur web non interactives qui permettent même d'utiliser un moteur Chromium ordinaire qui simule une navigation interactive alors qu'un script est aux commandes. Cette navigation est alors quasi-indistinguable d'un usager légitime.

Néanmoins, l'expérience montre que de nombreux attaquants, soit par ignorance, soit parce qu'ils n'ont pas besoin de les falsifier, laissent des valeurs marquantes dans ces entêtes.

Bloquer sur une valeur spécifique comme « user-agent » ou une typo sur un champ de cookie n'aura qu'un effet temporaire face à un attaquant décidé. En revanche, la mesure permet de détecter, et parfois bloquer des attaques dès les premières tentatives.

Ainsi, si un compte effectue une connexion valide avec un user-agent caractéristique d'une librairie du langage Python c'est un indicateur que ses identifiants sont possiblement compromis.

Ce type d'information peut d'ailleurs être associé à la gestion de session comme décrit dans la section sur la protection du cookie de session.

*Dissocier les usages*

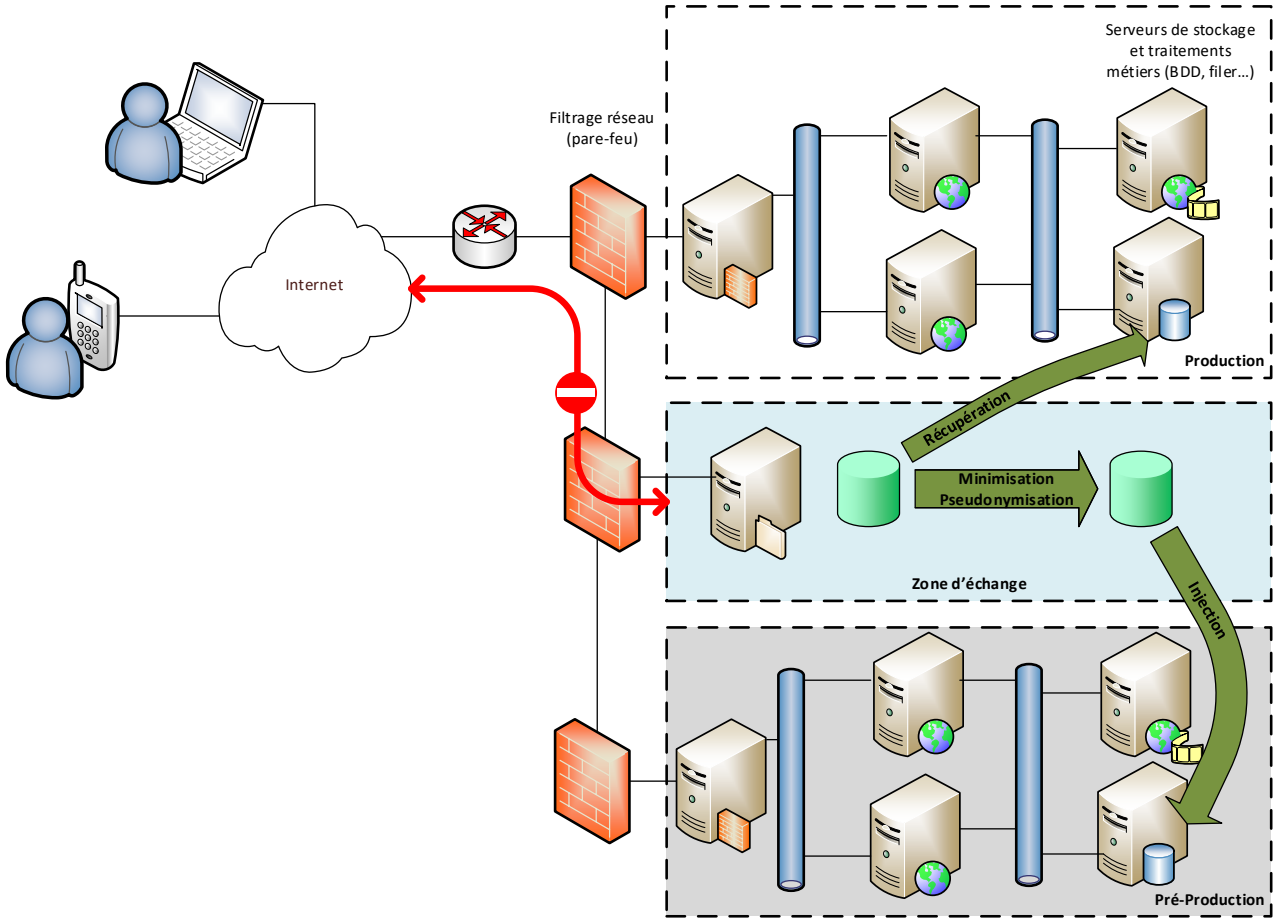
Les plateformes de préproduction, de recette et de développement doivent être strictement dissociées. Le filtrage réseau entre les différentes instances de l'application doit être strict.

[17] Voir en annexe *Contrôle d'accès : limiter les accès sur la base d'adresses IP*

[18] Voir en annexe *Limitation d'exfiltration : nombre de requêtes sur une fenêtre temporelle glissante*

Les échanges entre ces diverses plateformes doivent se faire de façon contrôlée en application de procédures spécifiques. Ceci devrait se faire au travers de zones d'échange dédiées et avec des flux initiés depuis la zone la plus privilégiée vers la moins sensible.

Par exemple, une synchronisation de donnée depuis la production vers la préproduction, devrait être initiée depuis un service sans accès Internet, s'exécutant depuis une tierce zone : en extrayant des informations de la production, et en les réinjectant dans la préproduction sans connexion directe entre les deux zones, sans flux initié depuis une zone portant l'application.




*Filtrer les flux entrants et sortants*

Les communications initiées par des applications ou les serveurs les hébergeant sont très stéréotypées et prévisibles. Les applications par défaut n'ont pas de raison d'initier des communications à destination d'adresses arbitraires sur Internet.

Il est recommandé d'instituer un filtrage strict, interdisant par défaut toutes communications sortantes et n'autorisant que celles à destination de domaines légitimes. Ceci peut être réalisé en forçant les flux à passer par un serveur mandataire filtrant[19]. Les flux sortants sont limités au strict nécessaire pour le maintien en condition de sécurité.

Un exemple de ce type de configuration est d'interdire tout flux sortant d'une DMZ applicative sauf venant du serveur mandataire[20]. Et dans la configuration du serveur mandataire de n'autoriser que

[19] S'il est courant de penser à utiliser des serveurs mandataires pour les flux HTTPS, certains comme HAProxy permettent de traiter la plupart des types de connexions TCP.

[20] Serveur proxy

les communications vers les serveurs officiels de dépôts pour Linux et de mise à jour pour Windows, et des services d'activation de licence.

*Exemple :* un serveur web n'a pas de raison évidente de se connecter vers une machine virtuelle d'hébergeur (VPS) ou des réseaux sociaux. Ces tentatives de connexion devraient être bloquées sur le serveur mandataire sortant et une alerte devrait être levée.

Par extension de la même logique, les flux entre les zones métiers devraient être limités à leur minimum et contrôlés en liste autorisée. Pour les interconnexions métiers, entre composants applicatifs, des ruptures peuvent être introduites au travers de bases de données ou d'échanges de fichiers.

**Cas des services web**

Beaucoup d'applications sont découpées en service web.
Cette architecture introduit des opportunités de contrôle et des complexités.

**Contrôle d'accès**
La consommation de données d'un service web opérant des informations non publiques doit se faire avec un contrôle d'accès à l'état de l'art.
C'est habituellement un jeton cryptographiquement signé confirmant le droit d'accès du consommateur.
Il est important de s'assurer que les informations transmises au service dans le jeton d'accès ou dans la requête permettent de valider l'identité du demandeur et de la tracer.

Dans certains incidents, le CERT-FR a rencontré des configurations où l'identité du consommateur final du service était perdue dans la transitivité entre un premier service frontal et un service final qui ne connaissait pas le consommateur. C'est un point important à vérifier.

**Exposition**
Par principe, le moins de services web possibles doivent être exposés sur Internet.

En particulier, le CERT-FR a rencontré des applications ou le code javascript de la page accédait directement à des services fournissant des données sans authentification ou contrôle d'accès. Cette configuration est à éviter.
Il est impératif de distinguer des services à usage interne de ceux exposés sur Internet. Au besoin, il est possible de développer un service exposé sur Internet qui ne soit qu'un agrégateur de services internes et mette en œuvre une sécurité appropriée.

**API Gateway**
Quand beaucoup de services sont consultés pour constituer une réponse, il est fréquent d'utiliser des *API Gateway*. Ce sont souvent les équipements qui vont constituer la frontière entre l'interne et Internet. Dans le cas où ce type d'infrastructure existe, c'est un point privilégié pour mettre en œuvre du filtrage, de la détection et valider l'exposition des services.
La plupart des *API Gateway* supportent plusieurs standards de contrôle d'accès. C'est donc aussi un point où ceux-ci peuvent être relativement facilement durcis.

## 4. Dépasser le login/mot de passe

### Principes

En 2026, l'authentification par le couple login/mot de passe est obsolète. Toute application qui repose uniquement sur ce mécanisme pour identifier ses usagers doit être considérée comme exposée à un risque fort de piratage.

Pour les applications accédant à de grandes bases de données, deux types de profils sont à distinguer. Les utilisateurs publics d'applications de masse n'ont généralement accès qu'à leurs propres informations ou à un nombre restreint d'entrées (typiquement pour mettre à jour ces informations). En revanche, les utilisateurs internes à l'organisation, ou ses partenaires, ont fréquemment des accès à une portion significative ou à la totalité des informations. Si un compte utilisateur interne est compromis, les dégâts sont potentiellement considérables en une seule fois. En revanche, sur des accès publics, des effets de masse sont atteints que par la compromission d'un grand nombre de comptes.

Une politique de sécurité réaliste vise à limiter trois familles d'attaques :

- Attaques indiscriminées sur les comptes « publics » : par énumération de mots de passe, ou utilisation de dictionnaires venus de divulgations de données antérieures.
- Attaques ciblées sur les comptes « internes » : par hameçonnage[21] et attaques dites d'adversaire au milieu[22].
- Attaques opportunistes basées sur des collectes d'identifiants volées par des *infostealers* touchant les deux profils.

**Limitations de l'authentification forte pour les applications grand public**

Beaucoup de mesures d'authentification robustes reposent sur des composants matériels et logiciels relativement récents ou haut de gamme.

Si une application s'adresse à un public très large, l'imposition de telles mesures résulte en l'exclusion d'une partie des usagers potentiels du téléservice.
Il est recommandé de prévoir des parcours alternatifs, y compris hors ligne pour gérer de tels cas.

Même quand une application doit accepter un mélange de clients « modernes » supportant des échanges sécurisés et de clients incompatibles, la mise en œuvre de mesures robustes sur les usagers compatibles est recommandée.

En effet, cette réduction de la population vulnérable permet de concentrer les mesures de détection sur les cas les moins protégés.

21 L'hameçonnage ou phishing est une technique de fraude consistant à contacter la victime en vue de la tromper pour l'inciter à communiquer des données personnelles et/ou bancaires en se faisant passer pour un tiers de confiance.
22 Les attaques d'adversaire au milieu (*adversary in the middle)* sont des attaques similaires à de l'hameçonnage ou l'attaquant joue interactivement les identifiants de la victime au moment de leur saisie sur le système cible, afin de faire valider, en temps réel les facteurs d'authentification secondaires.

## Mise en œuvre

*Protection des comptes utilisateurs « publics »*

Afin de protéger les comptes utilisateurs publics contre les attaques par énumération, un code à usage unique, demandé à chaque nouvelle session peut remplacer ou compléter le mot de passe. Ce code peut être transmis par un canal secondaire comme un message instantané, un SMS ou un mèl.

Cette mesure ne permettant pas de se protéger efficacement contre les attaques ciblées (hameçonnage, ou compromission du poste), elle devrait être complétée par une notification à l'utilisateur de toute nouvelle authentification à son nom.

**Utiliser *Webauthn* et les *passkeys***

*Webauthn* est un standard de sécurité porté par le W3C[23]. *Passkey* est le terme commercial désignant les identifiants *Webauthn* tels que proposés par la FIDO-Alliance.

Les *passkeys* sont des secrets d'identifications stockés par le navigateur et généralement protégées par un composant de sécurité de la plateforme.
Le vol des *passkeys* requiert donc de contourner le stockage par un composant de confiance ce qui est hors d'atteinte pour la plupart des attaquants. Comme pour tout second facteur connecté au poste de travail, il est possible à un attaquant maitre du poste d'usurper l'identité de l'utilisateur légitime[24]. Mais cela peut être détecté et requiert un niveau d'expertise sans commune mesure avec les attaques par réutilisation de mots de passe.

La *passkey* est créée sur un poste particulier après une authentification forte. Après initialisation, elle permet d'authentifier de nouvelles sessions avec une forte confiance et une bonne expérience utilisateur.

Voler directement les *passkeys* étant difficile, les processus d'enrôlement et de réinitialisation d'identifiants perdus restent vulnérables. Ils doivent faire l'objet d'une sécurisation forte. Leur faible fréquence d'usage permet toutefois la considération de mesures qui seraient trop intrusives pour un usage quotidien. Cette diminution du nombre d'évènement d'authentification permet aussi d'améliorer la supervision de sécurité.

En revanche, cette solution n'est supportée que sur des navigateurs récents et peu utilisables pour des sessions non interactives.
Il n'est donc pas possible de rendre une telle fonction impérative pour toute application visant un public large. Sur une grande population, des équipements anciens ne pourraient plus s'identifier.
Si *webauthn* est utilisé, un mode dégradé doit être prévu pour les usagers de navigateurs plus anciens.

*Mesures de transition pour les utilisateurs « internes »*

Une application dans laquelle des accès transverses sont possibles sur la base d'un simple couple login/mot de passe est à risque. Pour l'organisation qui l'opère, c'est le risque de se retrouver bloquée dans une situation ou la multiplication des incidents rend impossible la prise de recul et la planification à plus long terme. Dans certains cas, par exemple sur certaines infrastructures

[23] Voir https://www.w3.org/TR/webauthn/

[24] Voir par exemple https://unit42.paloaltonetworks.com/passwordless-authentication-security-risks/

infonuagiques, il est facile et rapide d'activer des seconds facteurs d'authentification faibles : généralement des codes à usage unique à saisir dans l'interface (OTP[25]).

Un tel déploiement peut être considéré, à titre transitoire, pour donner du temps au défenseur. Mais tous les facteurs d'authentification qui doivent être manuellement saisis sont vulnérables à l'hameçonnage. Les outils des attaquants sont matures et efficaces. Ces types de second facteur ne sont pas durablement efficaces contre une attaque ciblée en 2026.

Une autre solution est le déploiement de bi-clés et de certificats d'authentification TLS dans les navigateurs des postes clients à protéger. Bien que cryptographiquement forte, cette solution ne résiste pas aux *infostealer* qui sont capables d'extraire les certificats et leurs clés privées quand ils ne sont pas protégés par du matériel de sécurité (TPM, carte à puce). Cette solution souffre aussi de problèmes d'ergonomie, les erreurs d'authentification étant présentées par le navigateur comme des incidents de négociation de connexion réseau.

Si le déploiement de telles mesures ne peut pas être effectué rapidement, il convient de préférer des limitations d'accès réseau à l'application avant le déploiement de moyens d'authentification forte.

**La réinitialisation de mot de passe**

**Le problème**
Plutôt que d'attaquer l'authentification ordinaire, beaucoup d'attaquants vont réinitialiser les mots de passe à leur profit. Qu'ils contrôlent la boite de messagerie de la victime, soit ils disposent d'informations personnelles sur la victime de l'usurpation ils arrivent souvent à leurs fins.

**La solution**
Le processus de réinitialisation de mots de passe doit être considéré comme particulièrement sensible.

Pour des comptes à privilèges, comme des administrateurs applicatifs, cette opération doit être confirmée par une authentification forte ou par l'approbation d'un tiers.

Pour les comptes des usagers, le simple envoi d'un secret par email est aujourd'hui insuffisant. Il convient de compléter celui-ci d'un autre élément : envoi d'un secret sur un autre média[26], moyen d'authentification forte, validation depuis une autre identité garantie, validation par un tiers, vérification d'identité à distance[27], voire procédure avec un contact en vis-à-vis[28].

Dans tous les cas, ces réinitialisations doivent être tracées et traitées comme des évènements de sécurité.

[25] Les *One Time Password* (OTP), généralement des *Time Based One Time Password* (TOTP) sont la forme de second facteur d'authentification la plus courante. Ces codes à usage unique viennent généralement compléter le mot de passe de l'utilisateur. Ils sont souvent envoyés par SMS, ou générés par une application spécifique.

[26] Ce facteur doit être adapté aux enjeux de l'application. Suivant ceux-ci, on pourra utiliser SMS, canaux applicatifs des smarphones, voire un envoi de courrier physique.

[27] Voir https://cyber.gouv.fr/offre-de-service/solutions-certifiees-et-qualifiees/services-de-securite-evalue/solutions-en-cours-de-qualification/prestataires-pvid/

[28] Passage à un guichet avec une pièce d'identité par exemple.

*Protection des comptes « internes »*

Les accès transverses ou privilégiés doivent être contrôlés par des moyens d'authentification forte tels que définis dans le guide « ANSSI : Recommandations relatives à l'authentification multi-facteur et aux mots de passe[29] ».

Les mesures d'authentification peuvent être déployées directement dans l'application quand le mécanisme est supporté. C'est la solution idéale, et souvent ce type de support peut être ajouté par le déploiement d'une extension ou par simple reconfiguration.

Il est parfois possible de faire porter des fonctions d'authentification directement par les équipements de filtrage frontaux.

Enfin, l'état de l'art est de déporter les fonctions d'authentification depuis l'application vers un service fournisseur d'identité (IDP)[30]. Dans ce cas, l'utilisateur non authentifié est redirigé par les serveurs de filtrage applicatif vers le service d'identification avant de pouvoir accéder à l'application. Cette architecture permet de faire évoluer les moyens d'authentification indépendamment de l'application.

| **Intérêt des serveurs d'identité** |
| --- |
| Les serveurs d'identité (*Identity providers* ou IDP) sont devenus une brique commune des applications web.<br><br>Ce service extrait la gestion de l'identité et d'autorisation d'accès du code applicatif et délivre des tickets d'accès.<br>Ces tickets sont consommés par les services applicatifs afin de valider l'accès à la ressource ou au traitement demandé.<br><br>Cette architecture permet de simplifier la logique de contrôle d'accès et d'identification. Mais, surtout, elle permet de centrer dans un seul composant la vérification d'identité.<br><br>Ainsi, l'évolution des moyens d'authentification est simplifiée.<br>Sans IDP, des modifications de code applicatif substantielles peuvent être requises.<br>Avec un IDP, activer l'authentification multi-facteur ou supporter un nouveau standard n'affecte, coté serveur, que l'IDP. Dans la plupart des cas, c'est même une simple reconfiguration.<br><br>En revanche, l'IDP est une brique centrale de sécurité et doit être configuré et administré avec un haut niveau de précaution. Si l'IDP est piraté, toute la sécurité applicative est compromise. |

[29] Lien en section « Références »

[30] Un *Identity Provider (IdP)* ou fournisseur d'identité est un service, interne ou externe à l'organisation qui permet de découpler l'authentification des utilisateurs des applications elles même.

*Au-delà de l'authentification initiale.*

### Le problème du cookie de session

Dans la plupart des applications web, l'étape d'authentification permet d'ouvrir une session. Cette session est identifiée par un cookie de session. Après cette étape, toutes les demandes à l'application ne sont plus identifiées que par ce cookie.

Récupérer le cookie de session, c'est récupérer l'accès de l'utilisateur tant que la session reste valide.

### Evolution des attaques

Le vol de cookie est un problème ancien dans le développement web. Plusieurs familles de vulnérabilités historiques dans le code HTML ou Javascript permettent de récupérer l'identifiant de session. Mais, du fait du développement de l'authentification à multiples facteurs, d'autres méthodes deviennent populaires.

Le premier moyen d'attaque est l'installation d'*infostealer* sur les postes des victimes. Ces logiciels malveillants vont non seulement récupérer les clés cryptographiques, les logins et mots de passe, mais aussi tous les cookies de session stockés par les navigateurs.

Un autre exemple d'attaque est la manipulation de l'utilisateur ciblé pour aller vers un site hostile qui va se comporter en serveur mandataire vers les mires d'authentification de l'application ciblée et intercepter les cookies à la volée.

### Protections classiques du cookie de session

La plupart des mesures traditionnelles de protection des cookies de session visent à se protéger des failles web ou de l'interception réseau : marquage des cookies en `http_only` et `secure`, chiffrement strict des communications par TLS[31].

Malheureusement, ces mesures ne protègent ni des *infostealers*, ni des serveurs mandataires offensifs qui opèrent hors de l'application et au-dessus du chiffrement.

### Protections par invalidation de session

Une autre approche de cette protection est d'invalider la session et forcer l'utilisateur à se réauthentifier. L'attaquant en possession du cookie ne peut plus l'utiliser car il a expiré.

Coupler les expirations de sessions avec des mesures facilitant la reconnexion rapide comme les *passkeys* réduit grandement les capacités des attaquants.

Pour ce faire, le plus simple est d'avoir des vies de session courtes. Ainsi, la fenêtre d'utilisation de l'information volée se réduit et le temps que l'attaquant ait l'identifiant en main, la session est fermée. Néanmoins, cette mesure crée une course entre la vie des sessions et la capacité de l'attaquant à récupérer des identifiants « frais ». De plus, invalider fréquemment les sessions dégrade l'expérience utilisateur forcé de se réauthentifier.

La seconde approche vise à lier la session à ses conditions d'utilisation : un poste utilisateur précis, un navigateur précis (user-agent), un point de connexion spécifique. Ainsi un changement d'adresse IP source de connexion, ou simplement de zone géographique de l'adresse IP source de connexion, d'entête de description du navigateur est l'approche qui, dans notre expérience, offre aujourd'hui le

[31] On se référera pour le détail de mise en œuvre à la section 5.5.3 du guide ANSSI « Recommandations pour la mise en œuvre d'un site web : Maitriser les standards de sécurité coté navigateur » référencé en annexe.

meilleur ratio efficacité / effort / expérience utilisateur. Il convient dans une telle politique de ne pas oublier le cas des usagers mobiles dont l'adresse IP peut changer fréquemment.

Certaines méthodes pour réaliser ce lien vont aller plus loin en exécutant dans le navigateur des codes javascript qui vont tenter d'en extraire une empreinte. Cette méthode à l'inconvénient d'être très intrusive et de créer des signatures personnelles qui permettent de réidentifier des personnes au travers de leurs différentes identités. Cette traçabilité peut excéder les limites de la réglementation en termes de protection des données personnelles et il est recommandé d'accompagner la mise en place d'un tel dispositif d'un conseil juridique.

Toutes ces mesures n'empêcheront pas un attaquant compétent et motivé d'usurper l'identité de l'usager. Mais elles vont l'obliger à se connecter interactivement sur le poste utilisateur. Et ce changement présente plusieurs avantages pour le défenseur :

- Les usurpations d'identité à l'échelle de milliers ou même de millions d'usagers deviennent prohibitivement coûteuses pour l'attaquant.
- Les actions de l'attaquant, qui est forcé d'interagir avec le poste de travail, ont plus de chances d'être détectées et de laisser des traces forensiques.

**Un futur de la protection du cookie de session : DBSC**

Le W3C travaille sur une évolution de la gestion des *cookies* permettant de protéger cryptographiquement les identifiants de session : *Device Bound Session Credentials*[32] ou *DBSC*.

Dans ce standard, l'information de session est stockée et protégée par l'élément cryptographique du poste utilisateur[33] et par de la cryptographie asymétrique. Les *infostealers* ne peuvent alors plus extraire hors du poste le secret qui est protégé par le composant matériel.

Cette mesure viendra donc compléter la protection de l'ouverture de session sur toute la durée de sa vie.

L'usage de ce standard requiert une légère évolution de la négociation de session dans l'application[34]. Dans les architectures web modernes, c'est souvent un composant dédié le fournisseur d'identité[35] qui doit être adapté.

Sur le poste client, en mi-2026, ce standard n'est supporté que par le navigateur Google Chrome. La généralisation de ce standard dépendra de son adoption par les différents éditeurs de navigateurs.

Si l'usage de ce standard se répand, il est recommandé de l'adopter dans les applications.

[32] Voir https://www.w3.org/TR/dbsc/
[33] Sur un PC qui utilise le TPM, les *Trusted Elements* sur les téléphones mobiles, et *Secure Enclave* sur les produits Apple.
[34] Support de la cinématique de Secure Session Registration ou SSR.
[35] La formulation anglaise *Identity Provider* ou *IDP* est souvent utilisée pour désigner ce type de service.

## 5. Tracer les actions des utilisateurs en étant capable de les attribuer à un utilisateur et une adresse réseau

### Principes

Le CERT-FR a pu constater de multiples défauts de configuration de journalisation dans les infrastructures applicatives rencontrées : impossibilité de tracer l'association entre action et identité, perte des adresses sources des usagers dans les journaux, collecte de journaux à des points fournissant des informations incomplètes.

Dans de tels environnements, la détection d'incident et leur investigation sont difficiles, allongeant les délais de réaction. Un début d'incident qui aurait pu être endigué par une équipe de supervision de sécurité en quelques minutes va se transformer en investigation de plusieurs jours. Les attaquants peuvent alors profiter de ce délai pour poursuivre leurs actions.

L'attribution des actions est un point important pour permettre de comprendre les attaques et leurs impacts. Il est important de lier chacune des actions des utilisateurs ou des comptes de l'application à une adresse réseau dans des journaux en prenant en compte les accès réussis et en échecs, les actions réalisées au niveau des données type ajout, suppression, accès, tentatives …

La rétention de journalisation au niveau du concentrateur doit permettre de remonter plusieurs mois en arrière au moment d'une suspicion d'incident[36]. Il est important de mettre en place un collecteur de journaux applicatif en respectant les bonnes pratiques.

Pour une journalisation optimale, il est important d'avoir un service de temps partagé entre tous[37], et que les enregistrements soient sur un fuseau horaire harmonisé (UTC recommandé). Il convient aussi de s'assurer que les journaux contiennent des dates-heures complètes incluant l'année et la plus grande précision possible[38].

| Cas de problème de traçabilité de bout en bout opacifiant les actions attaquantes |
| --- |
| Le CERT-FR observe régulièrement que la détection d'activités attaquantes et leur compréhension sont entravées par une absence de traçabilité de bout en bout des actions utilisateurs. Le traitement d'un incident sur un périmètre regroupant plusieurs entités interconnectées a mis en évidence que ce manque de traçabilité avait nui à l'efficacité de certaines mesures de remédiation. En effet, suite à un premier incident un ensemble de compte utilisateurs compromis – permettant l'accès à des ressources de ces différentes entités – avaient été réinitialisés. Seulement un applicatif identifiait les utilisateurs d'une manière alternative, empêchant la réinitialisation des comptes d'être effective. Cela a conduit à un second incident sur cet applicatif, dont les données ont été exfiltrées *via* l'un des comptes compromis. |

[36] Dans la plupart des contextes 6 mois de rétention est un minimum. Les recommandations de l'ANSSI sur le sujet peuvent être trouvées dans le guide suivant : https://messervices.cyber.gouv.fr/guides/la-supervision-de-securite-piloter-un-projet-de-supervision#collection

[37] Le protocole de synchronisation le plus courant est NTP (https://www.rfc-editor.org/rfc/rfc5905 ) mais il est aussi possible d'utiliser des protocoles comme PTP (https://www.rfc-editor.org/rfc/rfc8173.html ). Sur les architectures virtualisées, la synchronisation de temps peut aussi être effectuée depuis l'hyperviseur, auquel cas, celui-ci doit être synchronisé sur une source fiable et partagée.

[38] Dans certains environnements, la configuration par défaut de la journalisation ne stocke pas l'année. Et certains protocoles de journalisation historiques (notablement le protocole Syslog-BSD) arrondissent les horodatages à la seconde rendant la reconstitution d'évènements difficile.

## Mise en œuvre technique

*Journaliser les accès à l'application*

L'application doit pouvoir permettre de journaliser les accès réussis comme ceux échoués.

Les entrées de journaux doivent au moins contenir les informations de date et d'heure, d'adresse source, de ressources visées, de taille d'échange, de l'état de l'échange (réussite, échecs, erreurs …), du nom du serveur ciblé et de l'utilisateur. Dans le meilleur des cas, il faudrait retenir l'ensemble des informations non confidentielles des entêtes HTTP.

**Traçabilité et horodatage**

**Le problème**
Dans leurs interventions, les agents du CERT-FR rencontrent souvent des configurations de serveurs de journaux UNIX qui rendent difficile le traçage des actions des attaquants.

**La précision des horodatages**
Ces configurations ont souvent une précision limitée à la seconde et stockent des horodatages sans date. Dans un environnement de production moderne, c'est insuffisant pour retracer des actions adverses. Il est recommandé de préférer un format de flux de journaux conforme au Syslog IETF[39] et des stockages d'horodatage stockant les informations précises à la milliseconde[40].

**La synchronisation des horodatages**
Un autre problème courant est la non synchronisation entre composants de la chaîne de traitement des requêtes.
Si tous les équipements n'utilisent pas la même heure ou si les zones horaires sont décalées, rapprocher les événements entre équipements devient très complexe.

Deux solutions sont impérativement à mettre en œuvre :

- Configurer une source de temps commune sur tous les équipements émettant des journaux ou des événements (et idéalement sur tous les équipements).
- Choisir une zone de temps commune et l'appliquer sur tous les équipements (GMT est souvent un bon choix).

Pour aller plus avant sur la supervision de sécurité, il est recommandé de se référer aux guides ANSSI sur le sujet[41].

La configuration de format de journaux peut généralement être effectuée dans le serveur web frontal. Les modules applicatifs disposent souvent de leur propre configuration de traçabilité[42] qui peut être ajustée pour obtenir la visibilité et le format désirés.

---

[39] Syslog IETF – RFC 5424

[40] Sauf besoin spécifique, Il est recommandé de stocker les horodatages conformément à la norme ISO 8601.

[41] Piloter un projet de supervision https://messervices.cyber.gouv.fr/guides/la-supervision-de-securite-piloter-un-projet-de-supervision, Recommandations de sécurité pour l'architecture d'un système de journalisation https://messervices.cyber.gouv.fr/guides/recommandations-de-securite-pour-larchitecture-dun-systeme-de-journalisation

[42] Par exemple, avec Log4J en Java, Log4Python en Python, LogRus en Go, ou dans les différentes librairies de télémétrie applicative.

### Journaliser les actions sur les données

L'application doit pouvoir enregistrer les actions qu'elle réalise sur les données type consultation, ajout, modification et suppression de données.

Il est important de pouvoir lier l'identité de l'usager, les adresses sources des connexions et les actions effectuées dans l'application. Ce lien peut être directement fait par l'inclusion des adresses dans les évènements applicatifs. Parfois, un identifiant de session permet de faire le lien entre les informations réseau et celles de l'application. Il doit alors être journalisé.

Il convient de définir un format de journaux standardisé pour que les informations du compte, d'horodatage, d'action, de l'état des échanges, du temps et de la taille soient correctement enregistrées et prises en compte dans la chaîne de supervision.

Une réflexion doit aussi être menée sur la traçabilité de l'accès aux données hors application :

- Accès aux fichiers de stockage ;
- Accès aux API de stockage ;
- Accès au travers de bases de données.

Les traces de ces accès, hors application, doivent aussi permettre de retracer la source réseau des actions, l'identité utilisée et le type d'action de manière horodatée.

#### *Mettre en place un puits de collecte d'événements*

Afin d'éviter l'effacement ou l'altération des données de journalisation par un attaquant, il convient de centraliser les journaux dans un collecteur dédié et protégé à l'état de l'art.

Beaucoup d'incidents étant détectés tardivement, ou encore signalés lorsque des informations sont rendues publiques, la capacité à rechercher les traces d'un incident dans le passé est cruciale. Le temps de rétention des données doit couvrir plusieurs mois. La durée précise de rétention dépend des réglementations et engagements contractuels applicables au système d'information et doit être choisie en s'appuyant sur une expertise juridique[43].

La capacité de stockage des journaux doit être proportionnée à la durée de rétention visée. Le cycle de vie des journaux doit faire l'objet d'une procédure, de préférence mise en œuvre de manière automatisée.

Il existe différentes manières de collecter les journaux :

- En utilisant les journaux UNIX par la configuration d'un client / serveur `syslog`[44] ;
- En utilisant l'infrastructure native en environnement Microsoft (*Windows Events Forwading*) ;
- En déployant un outil de collecte tiers[45] ;
- En programmant une procédure de collecte récurrente, par script ou même manuellement.

[43] Dans la plupart des cas, 6 mois est une durée de rétention minimale.

[44] `Rsyslog` étant en 2026 le plus courant dans les environnements Linux.

[45] Tels que Splunk-Forwarder ou FileBeat

**Prise en compte des équipements mandataires inverses dans les journaux d'applications web**

**Le problème**

La plupart des applications web sont opérées derrière des équipements frontaux : proxy inverses, équilibreurs de charge, WAF… Ces équipements terminent la connexion HTTPS avant l'application. Il en résulte que, sur le plan réseau, la connexion vers l'application semble toujours provenir de l'équipement frontal. Par défaut, les journaux applicatifs vont perdre l'information d'adresse IP source de connexion de l'usager.

Malheureusement, une telle configuration est fréquemment rencontrée dans les réponses à incident du CERT-FR et complique énormément la détection et l'analyse des attaques.

**La solution**

Les équipements frontaux peuvent être configurés pour inclure l'information d'adresse source dans les entêtes de requête HTTP. Ces entêtes ont fréquemment un champ `X-Forwarded-For`, `Forwarded` ou encore `X-Real-IP`, mais d'autres extensions propriétaires sont parfois rencontrées. Bien que par défaut, la plupart des services web ne journalisent pas ces champs, il est généralement relativement facile de changer la configuration ou le code de l'application pour conserver l'information.

Par ailleurs, lors de la prise en compte des entêtes d'origine, il est important de s'assurer que la source de l'information provient bien des serveurs frontaux légitimes.

Cette information doit être tracée dans les journaux des serveurs web. En outre, les journaux applicatifs doivent permettre de rapprocher une session avec les ? les actions d'un utilisateur de cette information d'adresse d'origine.

Des exemples de configuration pour les logiciels les plus courants sont fournis en annexe.

# 6. Sécuriser l'environnement d'hébergement de l'application

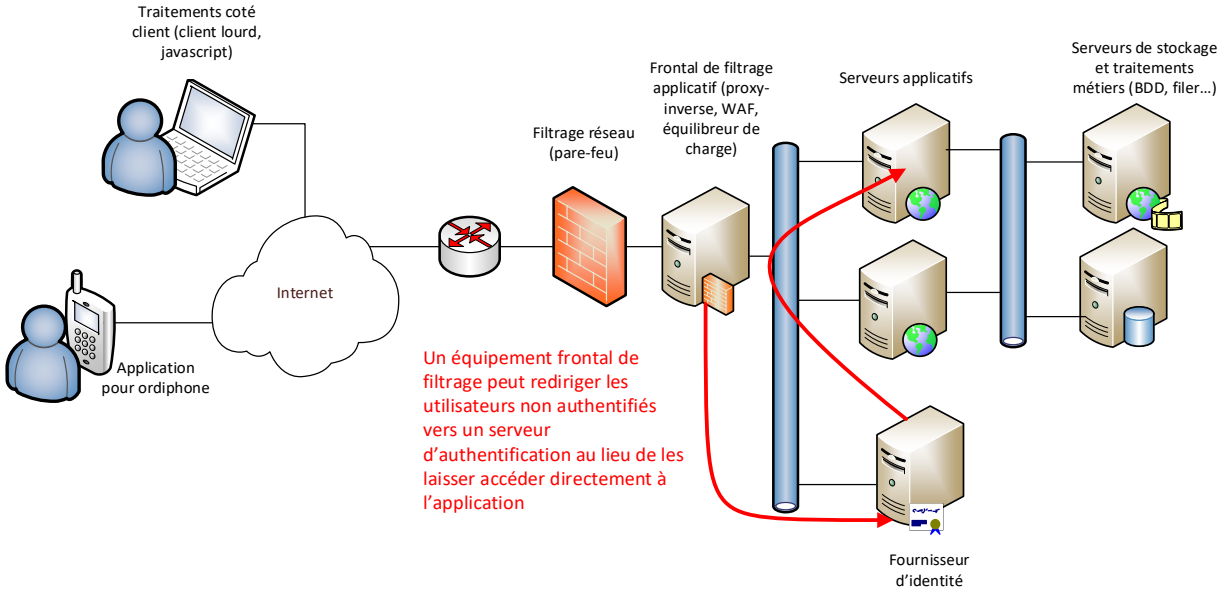



*Figure 4 - Architecture utilisant un IDP pour une application Web*

Les divulgations de données n'ont pas lieu que directement via l'application elle-même. Souvent, des accès à des services critiques restent exposés à Internet ou à des réseaux bureautiques. Ces interfaces permettent alors d'accéder aux données en contournant toutes les mesures de sécurité de l'application.

## Principes

### *Réduire l'exposition de la zone d'hébergement*

Aucune interface d'administration ne doit être accessible directement depuis Internet. Cette restriction s'applique quel que soit le type de cette interface : SNMP, SSH, RDP, web, ou protocoles propriétaires. Il faut en particulier penser à protéger les interfaces des éléments d'infrastructure : hyperviseurs (VCenter, ESX, Promox, HyperV…), accès administrateurs des pare-feux, des HAProxy et des frontaux…

### *Maintenir le niveau de sécurité de l'hébergement*

Le CERT-FR rencontre encore fréquemment des infrastructures dont les équipements ne sont pas à jour. Il est, bien sûr, important d'appliquer les correctifs de sécurité rapidement de façon générale.

Les équipements de sécurité et de réseau exposés sur Internet doivent être mis à jour systématiquement et immédiatement. Les exploitations de masse sont extrêmement rapides après la publication de code d'exploitation de faille. La compromission des équipements de bordure vulnérable donne aux attaquants un moyen efficace de contournement des mesures de sécurité, et souvent aussi des accès privilégiés sur toute l'infrastructure.

Il est important de mettre en place du maintien en condition de sécurité et opérationnelle. Celui-ci doit inclure avec un suivi régulier des vulnérabilités et une application à minima mensuelle des correctifs de sécurité sur les parties les plus exposées : pare-feu, HAproxy, reverse-proxy et frontaux. Il ne faut pas oublier, dans les mises à jour, les bibliothèques de composants qui peuvent être présents en plus du système et de l'application principale.

*Architecture en n-tiers*

Une segmentation de niveau 2[46] associée à des sous réseaux IP distincts est une mesure très efficace pour limiter les possibilités de latéralisation des attaquants.

Il est donc recommandé de mettre en place un modèle n-tiers distinguant les équipements frontaux, les blocs applicatifs et le stockage de données dans des zones dédiées.

Entre ces zones, il est recommandé d'appliquer un filtrage réseau strict n'autorisant que les flux légitimes.

*Pratiques d'administration*

La gestion des applications et des composants informatiques d'applications doit être réalisée depuis une zone dédiée aux postes d'administration.

Il faut interdire les accès issus des zones d'applications vers la zone des postes d'administration.

Les accès aux interfaces d'administration ne doivent être possibles que depuis la zone des postes d'administration. Les utilisateurs non privilégiés ne doivent pas pouvoir accéder aux interfaces d'administration.

Les accès internet depuis ces postes d'administration ne doivent pas être possibles sauf cas des postes nomades d'administration. Dans ce cas, ces postes ne doivent pouvoir établir sur Internet d'autres communications que le tunnel VPN d'administration.

Dans le cas où les postes d'administration doivent accéder à Internet, par exemple pour gérer un service *cloud*, le concept de réseau de postes d'administration reste valide. Ces postes ne devraient sortir sur Internet que depuis un sous-réseau ou une adresse IP précis qui soient explicitement autorisés dans les services en ligne. Tout accès privilégié venant d'une autre adresse est à considérer comme un incident de sécurité. Par ailleurs, cette sortie se fera de préférence au travers d'un relais applicatif sortant (typiquement un proxy web) qui sera configuré en liste d'autorisation limitée aux services strictement nécessaires.

*Journaliser et superviser les évènements sur les infrastructures des applications*

Les systèmes d'exploitation des infrastructures applicatives doivent être configurés pour générer des événements conformément aux bonnes pratiques de journalisation[47]. Ces traces doivent être centralisées. En particulier, les connexions locales ou à distance ainsi que les actions locales doivent être tracées. Pour ce faire, il est souvent suffisant de configurer les mécanismes natifs du système : journaux des événements Windows et service *Windows Event Collection* (*WEC*) ou *Syslog* UNIX qui fournissent un grand nombre de fonctions de sélection des événements et d'export vers des puits de journaux.

Sur un système Windows, la traçabilité peut utilement être augmentée par le déploiement de l'outil Sysmon[48]. Sur un système Linux, le service *auditd* permet d'augmenter la visibilité sur les actions locales et est recommandé.

Au-delà du système d'exploitation, les alertes de sécurité générées par les mesures de sécurité doivent être centralisées et supervisées. Il est recommandé de configurer et collecter les journaux

---

[46] Séparation physique ou VLAN

[47] Voir les guide « Piloter un projet de supervision » et « sécuriser la journalisation dans un environnement Microsoft Active Directory » référencés en annexe.

[48] Pour Sysmon voir : https://learn.microsoft.com/en-us/sysinternals/downloads/sysmon

des pare-feu système, services de contrôle d'intégrité, politiques d'exécution, modules de sécurité, EDR, antivirus, anti-rootkit… La mise en œuvre de moyen de sécurité sans supervision des alertes n'est pas recommandée.

Il faut mettre en place une supervision de sécurité pour les accès aux supports à l'application.

## Mise en œuvre technique

### *Mesures à court terme*

Les mesures à court terme visent à sortir d'une zone de danger imminent. Ces mesures ne permettent généralement pas une sécurisation de long terme dans des conditions acceptables.

#### Interdire les flux d'Internet vers les consoles d'administration

Les consoles d'administration, textuelles ou web ne doivent pas être exposées aux flux venant d'Internet.

Une mesure immédiate est de bloquer, au niveau du pare-feu, toute connexion TCP/UDP vers les ports d'administration (ex. : SSH, SNMP, HTTPS, ports d'administration et de déverminage de serveurs applicatifs, connexions à des bases de données) provenant d'adresses IP publiques.

Si certains services ne sont distingués de la production que par un chemin (par exemple « /admin »), il est recommandé de bloquer ceux-ci sur un pare-feu applicatif ou un serveur mandataire inverse.

A terme, les besoins d'administration devraient être servis au travers d'un réseau dédié aux postes d'administration.

#### Supprimer les services exposés inutilement

Beaucoup de services non utilisés sont fréquemment actifs sur les infrastructures de production. Ils constituent une surface de code significative qui peut être exploitée. De plus, même quand ils ne permettent pas d'établir des accès à distance, ces services permettent souvent des actions d'exploration.

Il est donc important de désactiver les services d'administration et de supervision qui ne sont pas utilisés dans les applications et dans le socle système. Par exemple, SNMP est actif par défaut sur beaucoup d'équipements d'infrastructure et peut être désactivé s'il n'est pas utilisé.

En environnement UNIX, le CERT-FR a souvent rencontré des services destinés à usage local (une machine) qui, pourtant, sont configurés pour écouter sur toutes les interfaces réseau du système. Dans la plupart des cas, il est possible de reconfigurer ces services pour n'écouter que sur l'interface locale. Ce changement est recommandé pour minimiser l'exposition de service même en intra-DMZ.

De même, si une interface est dédiée aux actions d'administration, les services d'administration à distance devraient n'écouter que sur cette interface.

<u>Exemple :</u> Changer un paramètre « `listen * :8888` » en « `listen 127.0.0.1 :8888` ».

Si l'option d'écouter sur une socket UNIX est disponible, elle est à préférer.

#### Appliquer les correctifs de sécurité sur toute la pile applicative

Le CERT-FR a souvent rencontré des systèmes où les délais d'application de correctifs de sécurité étaient longs. Même quand elles ne le sont pas avant publication de correctifs, les failles sont activement et rapidement exploitées depuis Internet.

Il est recommandé d'effectuer une revue des vulnérabilités en commençant par les équipements exposés. Il est impératif de corriger au plus vite les équipements vulnérables exposés sur Internet.

Un balayage de port sur les sous-réseaux IP de l'organisation depuis Internet peut permettre de s'assurer de l'absence d'oublis.

Dans les incidents traités par le CERT-FR où des vulnérabilités ont été exploitées, certaines le sont des mois après la sortie de correctif. Ceci est particulièrement important sur des failles restées exposées à l'intérieur du système d'information.

Il est important de ne pas se limiter aux corrections des équipements de bordures, mais de s'assurer de l'application en profondeur des mises à jour, même sur un rythme plus lent, à tous les composants d'infrastructure et d'application. Bibliothèques middleware et systèmes d'exploitation des serveurs internes, même s'ils ne reçoivent pas de trafic direct, sont à tenir à jour.

Même si dans un temps court, tout ne peut pas être mis à jour, chaque vulnérabilité couverte diminue l'exposition au risque. Si une vulnérabilité ne peut pas être corrigée, par exemple dans une application ancienne, des mesures palliatives doivent être considérées : supervision renforcée autour du composant, filtrage des requêtes réseau… Mais la correction de la vulnérabilité ou le remplacement du composant vulnérable doivent être planifiés, les mesures palliatives étant toutes faillibles à terme.

#### *À moyen terme : renforcement structuré*

Il est crucial d'aligner l'architecture réseau sur les bonnes pratiques d'hébergement en séparant les blocs applicatifs dans des zones réseau distinctes.

Dans les expériences du CERT-FR, une architecture à trois couches est souvent suffisante :

- Equipements frontaux : placé en DMZ, filtré par un pare-feu restrictif.
- Middleware : isolé dans un réseau interne, accessible uniquement depuis les équipements frontaux.
- Stockage et Base de données : zone dédiée, aucun accès direct depuis Internet ni depuis le LAN utilisateur.

Chaque grand bloc applicatif devrait disposer de bulle avec ces trois couches, et les flux entre ces blocs applicatifs devraient être restreints au minimum nécessaire.

### Instaurer des postes d'administration sécurisés

La compromission d'un poste d'administrateur est catastrophique. Quand un *infostealer* contamine un poste administrateur, ce sont potentiellement tous les secrets de l'infrastructure qui sont volés. La plupart des contaminations se font par un usage bureautique : traitement du courrier électronique, navigation Internet. Il convient de dédier des postes d'usage spécifique à l'administration qui ne servent jamais à des usages bureautiques.

- **Dédier des postes de travail spécifiques à l'administration**. Ces postes doivent faire l'objet d'un durcissement renforcé. En particulier, leurs disques doivent impérativement être chiffrés, les connexions à distance vers les postes doivent être impossibles et seuls les administrateurs doivent pouvoir y ouvrir des sessions.
- **Connecter ces postes à un réseau de postes d'administration** avec filtrage réseau strict : aucun flux ne doit pouvoir entrer vers le réseau d'administration. Si des accès distants sont nécessaires, utiliser un VPN dédié, d'accès basé sur une identité cryptographique du poste.
- **Interdire la navigation web sur ces postes** : sauf pour rejoindre les interfaces d'administration. Cette politique peut être implémentée sur les serveurs mandataires

sortant, sur la base du sous-réseau des postes d'administration. Si absolument nécessaire, par exemple pour accéder à des services « cloud » indispensables à l'administration, ces postes pourront accéder au web via un serveur mandataire (proxy) en listant explicitement les services autorisés et en interdisant tout autre.

Pour les mesures de plus long terme, il faut se référer au guide d'administration sécurisé de l'ANSSI[49].

[49] Voir le guide « recommandations relatives à l'administrations sécurisée des SI » cité en annexe

## 7. La supervision des journaux doit permettre d'y détecter les anomalies

### Principes

Des alertes de sécurité doivent être générées automatiquement lorsque des anomalies se produisent. Pour ce faire, les journaux des applications doivent être intégrés dans la supervision de sécurité et des règles de détection métier doivent s'y appliquer.

Le but de règles de détection métier est de caractériser un comportement qui sort de la normalité : une consultation d'un grand nombre de fiches, des consultations trop rapides pour un être humain, la modification ou la consultation de données qui sont sur plusieurs zones de responsabilité différentes, des utilisations à des heures inhabituelles…

Il est courant de réaliser des règles de détection par corrélation de plusieurs évènements et de plusieurs critères. Des règles peuvent se baser sur le comportement d'un compte, et combiner des critères techniques comme une adresse IP, les informations d'une session ; ainsi que des critères métiers comme le groupe auquel le compte appartient ou un rattachement géographique.

L'utilisation d'une application métier répond à des processus et des fonctions métiers spécifiques. Leur utilisation est cadrée par des besoins métiers. Les comportements qui s'écartent de la normalité sont parfois légitimes, mais peuvent aussi être des signes d'un incident en cours.

Tout élément anormal détecté doit faire l'objet d'une alerte automatisée. L'alerte doit être qualifiée pour en valider le caractère alarmant ou son innocuité. Cette qualification des alertes peut être assez longue. Dans certains cas, afin de limiter les dommages avant qualification, des actions automatiques peuvent être mises en œuvre.

Le triage des alertes et leur qualification doit faire l'objet de procédure. Cette procédure doit permettre une amélioration continue des règles afin d'affiner la capacité de détection en excluant les faux positifs.

Afin que les règles métier soient pertinentes, il est important pour les équipes de sécurité qu'une discussion avec le métier soit continue pour améliorer les cas d'usage et les seuils associés.

### Mise en œuvre technique

*Evènements de base*

Des situations telles que :

- Connexion en voyage impossible : qui est détectable par des connexions différentes dans un temps court de sources réseau géographiquement éloignées ;
- Enumération de comptes : caractérisée par des accès à de très nombreuses pages avec un mécanisme d'énumération sur un champ identifiant ;
- Taille de réponse anormale : qui dévie significativement de la moyenne et la médiane des échanges d'un compte (par exemple de +/- 33%) ;
- Tentatives ou connexions depuis des zones non conformes : par exemple, un compte qui tente une connexion depuis un réseau Tor ou un VPN alors qu'il arrivait toujours par une adresse IP résidentielle ;
- Tentatives de connexion en heures non ouvrables : alors que l'application n'est accessible qu'en heures de bureau, pourrait indiquer une compromission du compte ;

- Un compte qui dévie : quand un compte associé à une agence, une zone géographique, un club, un revendeur qui accède subitement à des données d'un autre rattachement ;

sont relativement universelles et devraient pouvoir être détectées sur la supervision de sécurité de l'application.

Les règles les plus pointues seront spécifiques aux fonctions de l'application. Mais elles requièrent une coopération dans la durée entre les équipes sécurité et les métiers pour leur mise en place qui s'improvise mal dans l'urgence.

*Problème des formats d'événements*

Dans cette activité de supervision, les sources d'évènements étant généralement diverses, une harmonisation est nécessaire : synchronisation sur les mêmes bases de temps, standardisation des formats de journaux. Dans ses interventions, le CERT-FR a pu observer que beaucoup de journaux applicatifs sont peu standardisés en rendant l'exploitation difficile.

Néanmoins, cette standardisation de format peut souvent se faire par configuration des différentes couches des équipements frontaux jusqu'aux librairies applicatives. Des exemples sont fournis en annexe pour certains cas rencontrés.

Parfois, une passe de transformation est requise entre les journaux bruts et ceux exploitables pour la supervision de sécurité. Ceci peut être implémenté sous forme de règles de transformation ou un simple script.

*Gestion des règles de détection*

Il est aussi recommandé pour la mise en œuvre de règles de détection de s'appuyer sur un format standard qui soit indépendant de la solution de supervision de sécurité utilisée. Un tel choix permet de bénéficier de nombreux exemples disponibles en sources ouvertes, mais aussi de trouver plus facilement des compétences pour accompagner l'écriture des règles.

Le format Sigma fait partie de ces standards pour lequel un exemple est fourni en annexe.

## 8. Déclencher des réactions automatiques et des alertes en cas de situation anormale

### Principes

La qualification des alertes de sécurité par une équipe humaine peut demander plusieurs heures de vérifications. De plus, la plupart des organisations n'opèrent pas d'équipes de supervision de sécurité en heures non ouvrées.

Le CERT-FR constate que les attaquants, en revanche, opèrent sur des horaires souvent décalés, utilisent des outils d'automatisation et peuvent atteindre leurs objectifs en quelques minutes après l'acquisition d'un accès.

Ce problème est rendu plus aigu par la disproportion entre la bande passante et la taille des informations à protéger. Les informations personnelles de millions de personnes peuvent ne représenter que quelques giga-octets d'informations qui sont exfiltrés en une poignée de minutes.

Pour couvrir ce décalage entre la vitesse de réaction humaine et celle de l'attaque, les défenseurs disposent de plusieurs outils :

- La défense en profondeur ralentit les actions adverses et augmente le décalage dans le temps entre les premières actions détectables et la réaction ;
- Les réactions automatisées peuvent entraver les capacités de l'adversaire avant qu'il n'atteigne ses objectifs finaux.

Par ailleurs, le CERT-FR a pu observer des attaquants utilisant des sources de connexions multiples à grande échelle : botnets, serveurs mandataires ouverts. Sont alors observées des centaines, voire des milliers, de sources différentes effectuant des actions similaires sur une application.

Qualifier et bloquer manuellement de telles listes n'est pas humainement possible. En revanche, l'automatisation de la défense permet de s'adapter à l'automatisation de l'attaque.

| Cas de tentatives successives avec process d'automatisation |
| --- |
| Dans un cas d'exfiltration de données d'un applicatif, les investigations ont permis d'identifier que l'attaquant avait dans un premier temps consulté ponctuellement les données à partir du compte utilisateur compromis. Quelques jours plus tard, des tests de requêtes automatisées ont été conduits par l'attaquant. Ces requêtes ont par la suite permis une exfiltration massive de données, événement qui a permis la détection de l'incident par la victime.<br><br>Les accès illégitimes précédents sont passés inaperçus, car ils étaient conduits à partir d'un compte disposant des droits adéquats et leur volumétrie était faible. |

## Mise en œuvre technique

*Réaction automatisée*

Il existe des orchestrateurs de réaction automatique de sécurité (SOAR) dédiés à la réponse rapide aux événements de sécurité. Ces outils permettent à partir d'alerte et d'une procédure précise de qualification et/ou d'autorisation de réaliser des actions d'endiguement automatique telles que désactiver un compte, mettre en place une règle dans un pare-feu.

Si une solution centralisée est plus facile à gérer, son déploiement peut être long et coûteux. Dans l'urgence, il n'est pas indispensable de déployer de telles solutions pour automatiser certaines réactions. Les seuils par session peuvent être gérés sur les équipements frontaux. Les anomalies d'ouverture de session peuvent se bloquer ou se ralentir dans les configurations des gestionnaires d'identité (IDP). Certaines applications permettent d'intégrer directement des règles de gestion dans l'application.

Quelle que soit la solution retenue, il est important que les actions automatisées soient tracées et visibles par la supervision de sécurité et applicative. La mise en œuvre doit en être coordonnée avec les groupes de support applicatif. En effet, les éventuels faux positifs risquent de créer des demandes de support et l'assistance doit pouvoir en accompagner l'opération.

*Mettre en place des seuils métiers d'alerte*

Il faut une discussion avec les métiers pour définir les seuils de détection d'anomalies dans le cadre des procédures, processus et fonction métier de l'entité. Les habitudes de travail des utilisateurs peuvent être aussi une source d'information intéressante.

Un ou une secrétaire médicale qui traite en moyenne 40 dossiers patients par jour, peut servir de base de référence pour décider d'un seuil maximum d'accès par utilisateur à des dossiers patients. Cette métrique devra être mesurée et comparée pour permettre d'ajuster au besoin les seuils. Par exemple, une alerte pourrait être levée à partir d'une consultation de plus de 50 dossiers patients. Le compte pourrait être désactivé si cette dernière consulte plus de 75 dossiers patients.

Face à des attaquants travaillant sur des durées longues, parfois des semaines, il peut être utile de prévoir des seuils cumulatifs sur de telles durée. Par exemple, on pourra lever une alerte sur un pic de consultation, mais aussi sur une consultation mensuelle inhabituelle.

Faute d'avoir prévu de telles mesures dans la conception de l'application, le pare-feu applicatif est souvent l'endroit où les actions peuvent être analysées et bloquées. Des seuils de ralentissement ou blocage basés sur la quantité de données par session ou par source de connexion sont faciles à mettre en œuvre sur tous les équipements frontaux[50]. En revanche, des règles plus fines, par exemple par type d'accès, ou corrélant des groupes utilisateurs avec des objets accédés, ne peuvent être déportés sur des frontaux que si la logique applicative est reflétée dans la structure des requêtes, par exemple dans les chemins d'accès. Si l'analyse des requêtes dans les frontaux est trop complexe, ceux-ci peuvent causer des ralentissements significatifs, voire devenir des causes de dénis de service.

Attention : Les choix de technologie d'implémentation de l'application sont déterminants dans la capacité à mettre en place des règles fines devant l'application. Une application mettant en œuvre

[50] Il convient néanmoins de se méfier des effets des NATs qui masquent de nombreux utilisateurs derrière une seule adresse réseau.

une logique REST est facile à gérer alors que des techniques consistant à encoder les requêtes dans des paramètres opaques ne le sont que rarement pour un effort acceptable.

*Mettre en place des procédures d'endiguement*

Les mesures d'endiguement d'une compromission métier peuvent être mises en place avec une procédure (gestion humaine) ou automatique (utilisation d'outils et options automatisées). Cette dernière est dépendante des briques technologiques utilisées ainsi que des capacités de l'application. Il est recommandé de l'anticiper lors de la création de l'application pour permettre à minima de réaliser des actions de verrouillage ou désactivation de compte ou de bannissement d'une adresse IP source.

Pour les procédures d'endiguement, il est recommandé de se référer au travail réalisé par l'InterCERT-France sur les fiches réflexes[51]. Elles constituent une base pour les travaux de procédures de qualification et endiguement métier.

L'automatisation des actions peut être effectuée à différents points de l'architecture : pare-feux, proxy-inverses, gestionnaire d'identité ou de l'application. Les annexes fournissent des exemples pour les différents niveaux. Les actions d'endiguement les plus courantes sont :

- Désactiver et/ou réinitialiser un compte,
- Interdire l'accès depuis une source vers une ressource,
- Révoquer des certificats ou des jetons ou des sessions réseaux,
- Limiter la bande passante,
- Réduire l'exposition.

**Problème des seuils évalués au niveau réseau**

Il est souvent plus simple de mettre en œuvre des seuils sur des équipements frontaux.
Ces seuils sont alors, basés sur la session TCP ou TLS. On peut ainsi limiter la taille des échanges afin d'éviter les exfiltrations de masse.

Des attaquants craignant ce type de détection peuvent les contourner en se déconnectant fréquemment et en se reconnectant depuis une autre adresse IP. C'est souvent fait en utilisant un VPN.

Bien que préférable à une absence totale de supervision, cette solution ne permet ni d'évaluer exactement ce qui est consulté, ni d'attribuer ces consultations à un compte utilisateur.

Il est donc très préférable de traiter les détections d'usage abusifs au plus près de l'application. Ceci permet, au minimum de s'appuyer sur l'identité utilisée mais aussi de développer des règles métier plus précises.

[51] Voir https://www.intercert-france.fr/publications/fiches-reflexes/ pour le secteur de la santé, le CERT-Santé met à disposition une déclinaison plus spécifique des mêmes fiches : https://cyberveille.esante.gouv.fr/reagir-un-acte-de-cybermalveillance-fiches-reflexes

## 9. La sécurité de l'application doit être vérifiée

### Principes

Aujourd'hui il est acquis qu'une application doit être auditée pour sa sécurité avant sa mise en production.

Mais, la sécurité d'une application évolue avec le temps. En effet, les configurations peuvent évoluer, de nouvelles vulnérabilités être découvertes dans des composants logiciels, des évolutions d'architecture peuvent exposer excessivement des services initialement privés.

Afin de garantir la sécurité d'une application dans la durée, il convient de revérifier périodiquement sa sécurité. Ce processus, nommé maintien en condition de sécurité (MCS), vise à soutenir dans la durée le niveau de sécurité d'une application.

#### *Réaliser un inventaire et le maintenir à jour*

L'inventaire des applications, des librairies et des composants de l'application doit permettre d'identifier les composants qui y sont inclus. Les configurations évoluant avec le temps, il convient de mettre en place un processus récurrent s'assurant que cet inventaire soit tenu à jour.

Si une SBOM[52] existe elle pourra servir de référence à la veille sur les vulnérabilités en particulier sur les produits complexes qui peuvent intégrer de nombreux composants tiers.

#### *Audit de sécurité*

Dans de nombreux incidents traités par le CERT-FR, des failles exploitées auraient pu être découvertes par des tests d'intrusion ou des audits de code.

L'audit est le premier moyen de vérification d'une application, qu'il soit initial lors de la mise en production, ou reproduit périodiquement. Le niveau d'audit approprié à une application dépend de sa criticité. L'audit peut être piloté par le processus d'homologation si le système est homologué. Sinon, l'analyse de risque de l'application permet de l'orienter autour des évènements redoutés.

#### *Mesurer l'exposition de l'application régulièrement*

Le filtrage des flux d'une infrastructure évolue dans la durée et peut diverger significativement des configurations documentées. Le CERT-FR a pu constater dans plusieurs cas que des ports ouverts temporairement, pour déverminer des configurations, ou simplement par erreur, peuvent rester durablement exposés et être exploités par les attaquants. Ainsi, mires d'administration, bases de données et ports de contrôle de serveurs d'application peuvent devenir accessibles depuis Internet. Ces cas sont aussi fréquents dans les infrastructures infonuagiques ou la gestion des flux nécessite une approche différente de l'habituelle gestion de pare-feu.

Les balayages d'exposition sont un moyen de vérification des applications. Suivant leur mise en œuvre, ils peuvent simplement vérifier les ports et services exposés. De plus, ils peuvent être liés à des bases de données de vulnérabilités. Au minimum, l'exposition à Internet et les expositions de services internes doivent être vérifiées mensuellement et lors de changements d'architecture.

---

[52] *Software Bill Of Materials* (SBOM) : liste des composants inclus dans un logiciel exprimée dans un format technique tel que SPDX ou CycloneDX afin d'en permettre un traitement automatisé.

*Veille en vulnérabilité*

La connaissance des vulnérabilités dans les logiciels évolue en permanence. Des failles peuvent être identifiées dans le logiciel principal de l'application. Mais toutes les applications modernes incluent un grand nombre de composants et de librairies qui chacun contiennent des failles.

La veille sur les vulnérabilités est une autre approche de vérification de la sécurité. Elle va viser à identifier les failles connues pour les corriger.

L'inventaire des composants logiciels doit être comparé aux publications de failles sur les sites spécialisés[53]. Toutes les vulnérabilités ne font pas l'objet d'enregistrement explicite. Il convient aussi de vérifier périodiquement les sites des éditeurs des produits déployés sur le système d'information.

Les vulnérabilités découvertes, doivent être qualifiées[54] et un plan de gestion doit être mis en œuvre : corrections de court terme, corrections de long terme, mesures palliatives. En termes d'évaluation de criticité d'une vulnérabilité applicative, il peut être intéressant de démarrer de la partie la plus exposée vers la zone interne. Par exemple, un équipement périmétrique ou un frontend exposé est plus à risque que la base de données non exposée qui est derrière le frontend et le middleware.

Attention : Le CERT-FR a plusieurs fois pu constater que des mesures temporaires de protection contre l'exploitation d'une vulnérabilité étaient pérennisées sans jamais corriger la faille. Cette pratique est dangereuse, car les attaquants tendent à développer des chemins d'exploitation alternatifs qui finissent par contourner les mesures temporaires. Il est recommandé de revoir périodiquement l'efficacité de mesures temporaires vis-à-vis de l'attaque[55] jusqu'à la correction durable de la faille.

## Mise en œuvre technique

*Réaliser un inventaire des composants des plateformes applicatives et le maintenir à jour*

Il existe de nombreux outils pour faire des inventaires[56]. Il est important de constituer et maintenir un inventaire des applications métiers contenant, au moins, les éléments suivants :

- Système d'exploitation et la version,
- Les applications et leur version,
- Les équipements et leur version,
- Les librairies mises en œuvre dans les applications.

Il est recommandé de maintenir à jour automatiquement l'inventaire avec des outils spécifiques.

Il convient de sauvegarder et d'extraire régulièrement cet inventaire pour qu'il ne soit pas chiffré ou altéré ou indisponible le jour d'un incident ou d'une crise. Au pire, un maintien manuel lors des changements sur l'application peut être effectué. Cela demande des procédures et un processus strict de gestion. Ces procédures, leur documentation et leurs outils devraient aussi être sauvegardés et extraits hors du système d'information.

---

[53] Plusieurs sources de veille sur les vulnérabilités existent : notamment CERT-FR, EUVD, NVD

[54] En particulier, l'applicabilité de la faille, mais aussi sa simplicité d'exploitation, son exposition, et les impacts potentiels de son exploitation permettent de prioriser les mesures correctives.

[55] Par exemple en vérifiant les évolutions de bulletins sur le site du CERT-FR ou de l'éditeur du logiciel.

[56] Pour faire la découverte de composants, il est possible de s'appuyer sur des outils spécialisés tels que GLPI, OCS Inventory, mais aussi les EDR, les solutions d'automatisation d'administration (ansible, puppet, saltstack…), ou les outils de vérification de vulnérabilités.

*Audit de sécurité automatisé*

Il existe des outils qui permettent d'auditer automatiquement des applications au niveau du code source, ou par container. Il est recommandé d'inclure de tels outils dans les forges logicielles[57]

Chaque outil a des spécificités : analyse de code et de bonnes pratiques de développement, scan de vulnérabilités connues dans un container ou une application web… Il revient de prendre l'outil le plus adapté à l' application concernée. Il est préférable de pouvoir mettre en place des pratiques de revue de codes et de configuration mensuellement et lors de changement majeur.

Pour les applications sensibles, les audits qualifiés PASSI tant sur l'architecture que sur les vulnérabilités sont une solution pour avoir un avis externe et qualifié.

Le durcissement et l'audit des socles systèmes, des annuaires et de leur configuration sont des sujets à part entière. Il convient de se référer aux guides de l'ANSSI sur le sujet, ou aux profils CIS[58].

**Services d'audit du CERT-FR**

Certains services du CERT-FR sont ouverts aux organismes étatiques et aux entités régulées.
Ils permettent de vérifier la sécurité des environnements d'exploitation des applications comme leur exposition.
Pour y accéder, il faut s'inscrire sur ClubSSI et de sélectionner l'outil approprié dans la section *ClubSSI - Audits*.

**SILENE**

SILENE est un outil de gestion de l'exposition. Il est disponible sur le site permettant de faire partie des scans mensuels de l'agence et d'avoir des remontées régulièrement.

**ADS[59]**

ADS permet l'audit du niveau de sécurité des annuaires Active Directory est disponible sur le site du Club SSI. L'analyse se base sur une extraction de configuration par l'outil ORADAD disponible en logiciel libre en ligne[60].
L'outil permet la collecte d'information sur les contrôleurs de domaine.

*Veille en vulnérabilité*

Si un inventaire a été constitué et maintenu, il est recommandé d'automatiser la majorité de la veille en vulnérabilités.

Les outils de gestion de vulnérabilité sont une famille de produits à part entière. Les outils de gestion de configuration de sécurité vont rafraîchir périodiquement la liste des composants logiciels depuis

[57] L'intégration de vérification de sécurité dans les ateliers logiciels est particulièrement critique dans les infrastructures à déploiement continu (CI/CD).

[58] Center For Internet Security https://www.cisecurity.org/cis-benchmarks

[59] Voir https://messervices.cyber.gouv.fr/services/ads.html

[60] L'outil ORADAD: https://github.com/ANSSI-FR/ORADAD/releases

les systèmes[61]. A l'inverse, les scanners de vulnérabilités vont combiner les balayages réseau avec les bases de vulnérabilités[62].

Dans les cas les plus importants, des équipes de CTI pourront réaliser cette veille en parallèle d'une veille centrée sur l'exposition.

*Mesurer l'exposition de l'application régulièrement*

Vérifier régulièrement l'exposition réseau des applications et infrastructures vis-à-vis d'Internet, mais aussi des autres zones du système d'information est très important.

Cette vérification peut être effectuée par une procédure manuelle ou par scripts. Il est néanmoins courant d'utiliser un logiciel de scan de vulnérabilités pour cartographier les services exposés.

Une alternative, pour l'exposition Internet, est de s'appuyer sur des services tiers (outils d'EASM[63]) qui peuvent aussi lever des alertes en cas de changement d'exposition[64].

Les prestataires PRIS incluent souvent des équipes CTI dans leurs prestations qualifiées. Ils peuvent généralement fournir des services de supervision d'exposition et de vulnérabilités[65].

---

[61] Comme les outils de l'écosystème OpenSCAP

[62] Comme OpenVAS ou Nessus

[63] *Exposure Asset Security Management* (EASM) : domaine des outils destinés à superviser

[64] Tels que les services : Onyphe, Censy ou Shodan.

[65] Ces services peuvent être présentés comme de l'EASM ou du *Continuous Threat Exposure Management* (CTEM)

## 10. Mettre en œuvre une démarche d'homologation pour chaque application conformément au RGS[66] et à la PSSIE[67].

### Principes

L'homologation est une démarche de gestion des risques d'un système d'information.

L'homologation de sécurité est obligatoire pour l'ensemble des entités publiques (ou « autorités administratives ») au titre du Référentiel Général de Sécurité (RGS) visant au renforcement de la confiance des usagers dans les services publics numériques, et du décret 2022-513 du 8 avril 2022 relatif à la sécurité des systèmes d'information et de communication de l'État et de ses établissements publics qui étend le processus d'homologation à l'ensemble des systèmes d'information de l'Etat.

La décision d'homologation atteste que les mesures de sécurité mises en œuvre, ainsi que les efforts additionnels planifiés, sont jugés suffisants pour faire face aux risques cyber identifiés contre lesquels une entité choisit de se protéger.

Homologuer un système d'information permet de :

- S'assurer que les risques de sécurité liés à l'utilisation du système d'information ont été identifiés et que les mesures nécessaires pour les traiter sont ou vont être mises en œuvre dans un délai raisonnable ;
- S'assurer que les réglementations en vigueur sont correctement appliquées ;
- Valider le plan d'actions à appliquer au système d'information afin de renforcer sa sécurité ;
- Responsabiliser et engager l'autorité qui accepte son emploi et toutes les parties prenantes associées.

La démarche à mettre en œuvre pour homologuer un système d'information est détaillée dans le guide ANSSI « L'homologation de sécurité des systèmes d'information »[68].

Vis-à-vis des risques de violation de données personnelles, la démarche d'homologation permet de mettre en œuvre un niveau de sécurité proportionné aux enjeux, mais aussi d'assurer le maintien de ce niveau dans la durée.

En particulier, l'analyse de risque devrait intégrer les données personnelles comme un actif sensible par essence.

*Réaliser une analyse de risque prenant en compte la confidentialité des informations à protéger*

La démarche d'homologation doit inclure une analyse de risque prenant en compte la protection des informations sensibles. Lorsque les informations sont liées à un traitement de données personnelles qui sont susceptibles d'engendrer un risque élevé pour les droits et libertés des

[66] Référentiel Général de Sécurité (RGS) : https://cyber.gouv.fr/reglementation/reglementation-identite-confiance-numerique/securite-echanges-voie-electronique/referentiel-general-de-securite/documents-referentiel-general-de-securite/

[67] Politique de Sécurité des Systèmes d'Information de l'Etat (PSSIE) : https://www.legifrance.gouv.fr/circulaire/id/38641

[68] Lien en section « Références »

personnes concernées, l'analyse du risque doit être couplée à l'analyse d'impact relative à la protection des données (AIPD) [69].

*Sélectionner des mesures centrées sur la protection des données manipulées par l'application*

Les mesures retenues permettant de traiter les risques doivent minimiser l'impact d'un incident de sécurité lié aux données manipulées par l'application.

*Prendre en compte les scénarios du présent document*

Les scénarios de risques considérés dans le présent document devraient faire partie de toute analyse de risque portant sur une application traitant de données personnelles ou sensibles :

- Accès aux données via l'application sur la base d'identifiants usurpés,
- Accès direct aux données contournant les mesures applicatives,
- Usage illégitime d'un accès interne légitime,
- Exploitation de vulnérabilités applicatives pour contourner les mesures de sécurité applicatives.

## Mise en œuvre

La mise en œuvre des mesures spécifiques de sécurisation identifiées dans l'analyse de risques se fait normalement suivant le processus d'homologation.

Néanmoins, les mesures techniques listées dans le présent document peuvent fournir des pistes d'implémentation dans la réduction des risques.

[69] L'analyse d'impact relative à la protection des données (AIPD) : https://www.cnil.fr/fr/RGPD-analyse-impact-protection-des-donnees-aipd

# REFERENCES


- CNIL – Violation massives de données en 2024 - https://www.cnil.fr/fr/violations-massives-de-donnees-en-2024-quels-sont-les-principaux-enseignements-mesures-a-prendre
- CNIL - Guide pratique RGPD : la sécurité des données personnelles - https://www.cnil.fr/sites/default/files/2024-03/cnil_guide_securite_personnelle_2024.pdf
- CERT-FR –Exfiltration de données du secteur social : retour d'expérience du CERT-FR - https://cert.cyber.gouv.fr/uploads/CERTFR-2024-CTI-009.pdf
- ANSSI - L'Essentiel sur les fuites de données - https://messervices.cyber.gouv.fr/documents-guides/anssi_essentiels_fuites_donn%C3%A9es_1.0.pdf
- ANSSI - Recommandations relatives à l'authentification multi-facteur et aux mots de passe - https://messervices.cyber.gouv.fr/guides/recommandations-relatives-lauthentification-multifacteur-et-aux-mots-de-passe
- ANSSI - Recommandations pour la mise en œuvre d'un site web : Maitriser les standards de sécurité coté navigateur - https://messervices.cyber.gouv.fr/documents-guides/anssi-guide-recommandations_mise_en_oeuvre_site_web_maitriser_standards_securite_cote_navigateur-v2.0.pdf
- ANSSI - Recommandations relatives à l'administration sécurisée des systèmes d'information - https://messervices.cyber.gouv.fr/guides/recommandations-relatives-ladministration-securisee-des-si
- ANSSI - Sécuriser la journalisation dans un environnement Microsoft Active Directory - https://messervices.cyber.gouv.fr/guides/securiser-la-journalisation-dans-un-environnement-microsoft-active-directory
- ANSSI - Piloter un projet de supervision - https://messervices.cyber.gouv.fr/guides/la-supervision-de-securite-piloter-un-projet-de-supervision
- ANSSI - L'homologation de sécurité des systèmes d'information - https://cyber.gouv.fr/securisation/homologation-de-securite/
- InterCERT-France – Fiches réflexe sur incident - https://www.intercert-france.fr/publications/fiches-reflexes/

# ANNEXES

Les annexes de ce document fournissent des exemples de configuration liés aux recommandations qui précèdent. Ces configurations doivent faire l'objet d'une adaptation au cas particulier de votre système d'information avant tout déploiement.

Les logiciels et systèmes cités ne le sont que parce que le CERT-FR les a rencontrés sur des incidents. Ces choix ne constituent pas une recommandation de produit.

Par ailleurs, ces configurations ne constituent pas des recommandations d'architecture, mais juste des indications sur comment mettre en œuvre les principes énoncés dans le document dans les cas rencontrés récemment par le CERT-FR.

## A. Traçabilité : propager et journaliser les adresses sources de connexion

Les équipements frontaux, WAF, proxy-inverses reçoivent les connexions depuis les postes clients de l'application. Afin de ne pas perdre en traçabilité ils doivent propager cette adresse dans les entêtes HTTP vers le serveur applicatif.

Point d'attention : La normalisation ou la mise en place de journalisation spécifique doit se faire en coordination avec les équipes de supervision afin de ne pas rendre cette dernière inopérante par l'ajout de champs, la modification du format …

### Transmettre l'IP source avec Nginx en proxy-inverse

**Exemple de fichier nginx.conf pour un reverse proxy positionnant ou mettant à jour le champ X-Forwarded-For**

```
…
server {
…
  location / {
    proxy_pass http://your_app_server;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header Host $host;
  }
}
```

*Référence :*

- https://nginx.org/en/docs/http/ngx_http_realip_module.html

### Transmettre l'IP source avec Apache HTTPD 2.4 et mod_proxy_http

Quand `mod_proxy_http` est activé dans Apache, il ajoute automatiquement les entêtes `X-Forwarded-For` (adresse IP du client) `X-Forwarded-Host` (L'hôte d'origine demandé par le client dans l'en-tête de requête HTTP Host) `X-Forwarded-Server`.

Optionnellement, si plusieurs serveurs mandataires sont chaînés, il est recommandé de spécifier les adresses sources de confiance avec la directive `RemoteIPTrustedProxy`.

*Référence :*

- https://httpd.apache.org/docs/2.4/mod/mod_proxy.html

### Transmettre l'IP source avec Apache HTTPD 2.4 et mod_log_forensic

Pour la journalisation avec Apache, il existe le module `mod_log_forensic`. L'utilisation de ce module permet d'avoir un suivi strict des sessions avec un `id forensic` unique permettant de suivre l'ensemble d'un flux. De plus, le format du module utilise un séparateur de type "|" entre les champs permettant de strictement normaliser les champs et leur emplacement dans le journal. Le chemin standard des journaux de type `mod_log_forensic` sont dans `/var/log/httpd/forensic_log`

L'intérêt du module `mod_log_forensic` n'est pas juste un suivi amélioré des sessions. Ce module rend plus facile l'intégration dans une chaîne de collecte et d'analyse de journaux, entre autres en générant des événements plus faciles à traiter informatiquement.

*Référence :*

- https://httpd.apache.org/docs/current/mod/mod_log_forensic.html

### Transmettre l'IP source avec HAProxy

Ajouter l'option `forwardfor` dans les sections `defaults`, `frontend`, ou `backend` fait automatiquement ajouter le champ `X-Forwarded-For` dans les entêtes http retransmises.

```
backend webservers
  …
  option forwardfor
  …
}
```

*Référence :*

- https://www.haproxy.com/documentation/haproxy-configuration-tutorials/proxying-essentials/client-ip-preservation/add-x-forward-for-header/

### Transmettre l'IP source avec Caddy

Par défaut, Caddy configuré en proxy inverse va positionner les champs `X-Forwarded-For`, `X-Forwarded-Proto` et `X-Forwarded-Host` des requêtes retransmises. Si ces champs réexistaient ils seront écrasés.

Si plusieurs serveurs frontaux sont chaînés, il est recommandé de positionner l'option `trusted_proxies` afin que les champs soient incrémentés des valeurs du saut précédent.

*Référence :*

- https://caddyserver.com/docs/caddyfile/directives/reverse_proxy

## B. Traçabilité : configurer la journalisation des adresses sources de connexion

La configuration du format de journalisation permet de spécifier les informations à stocker. Il est nécessaire de configurer ces formats afin d'y récupérer l'information de l'adresse IP du client initial de la requête

Attention : Si les journaux sont centralisés, il faut s'assurer de la compatibilité du format spécifié avec les chaînes de supervision de production et de sécurité.

### Configurer Nginx pour journaliser les adresses sources derrière un proxy-inverse

**Exemple de fichier *nginx.conf* pour un serveur web traçant le champ *X-Forwarded-For***

```
http {
  set_real_ip_from 192.0.2.0/24;        # exemple de sous-réseau des proxy
  real_ip_header X-Forwarded-For;
```

```
  real_ip_recursive on;                # support pour des couches multiples de proxy
}
…
```

*Référence :*

- https://nginx.org/en/docs/http/ngx_http_realip_module.html

## Apache HTTPD 2.4 pour journaliser les adresses sources derrière un proxy inverse

Il faut au préalable activer le module optionnel *remoteip* (`a2enmod remoteip` sous distribution Debian et ses dérivés).

**Exemple de configuration *apache2.conf* pour le serveur web traçant le champ *X-Forwarded-For***

```
…
RemoteIPTrustedProxy 192.0.2.0/24        # Sous reseau des serveurs proxy
RemoteIPHeader X-Forwarded-For
LogFormat "%a %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined
…
```

*Référence :*

- https://httpd.apache.org/docs/current/mod/mod_remoteip.html

En l'absence du module `remoteip`, il est aussi possible de tracer directement le contenu de la valeur d'entête en modifiant le format de fichier de journalisation. En revanche, les applications sous-jacentes devront elles-mêmes extraire l'entête si elles doivent la traiter.

**Exemple de configuration apache2.conf pour tracer le champ *X-Forwarded-For***

```
…
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined
LogFormat "%{X-Forwarded-For}i %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-
Agent}i\"" proxy
SetEnvIf X-Forwarded-For "^.*\..*\..*\..*" forwarded
CustomLog "logs/access_log" combined env=!forwarded
CustomLog "logs/access_log" proxy env=forwarded
….
```

## Configurer HAProxy pour journaliser l'adresse IP source derrière un proxy-inverse

**Exemple de configuration du format des journaux avec HAProxy**

```
…
log-format
"%ci|%cp|[%tr]|%ft|%b/%s|%TR/%Tw/%Tc/%Tr/%Ta|%ST|%B|%CC|%CS|%tsc|%ac/%fc/%bc/%sc/%rc|%s
q/%bq|%hr|%hs|%{+Q}r"
…
```

*Référence :*

- https://www.haproxy.com/documentation/haproxy-enterprise/administration/logs/

## Exemple avec Internet Information Server (IIS)

*Configuration pour les versions historiques de IIS (<8)*

**Configuration pour le site par défaut d'un serveur IIS 7 ou plus récent via Powershell**

```
Import-Module WebAdministration

Add-WebConfigurationProperty
    -pspath 'MACHINE/WEBROOT/APPHOST'
    -filter "system.applicationHost/sites/siteDefaults/logFile/customFields"
    -name "."
    -value @{logFieldName='X-Forwarded-For';
             sourceName='X-Forwarded-For';
             sourceType='RequestHeader'}
```

*Configuration pour IIS 8.5 et suivants via Powershell*

A partir de la version 8.5 de Internet Information Server les requêtes contenant X-Forwarded-For peuvent être directement journalisées depuis le serveur web.

```
Set-ItemProperty IIS:\Sites\siteName -name logfile.customFields.collection -value
 @{logFieldName='X-Forwarded-For';sourceType='RequestHeader';sourceName='X-Forwarded-
For'}
ou
%systemroot%\system32\inetsrv\APPCMD.EXE set config -
section:system.applicationHost/sites /+"[name='Default Web
Site'].logFile.customFields.[logFieldName='X-Forwarded-For',sourceName='X-Forwarded-
For',sourceType='RequestHeader']" /commit:apphost
```

*Références :*

- https://techcommunity.microsoft.com/blog/iis-support-blog/how-to-use-x-forwarded-for-header-to-log-actual-client-ip-address/873115
- https://learn.microsoft.com/fr-fr/iis/get-started/whats-new-in-iis-85/enhanced-logging-for-iis85

## Exploiter l'adresse IP source au travers d'un proxy inverse avec Python Flask

**Exemple de code permettant de récupérer l'adresse IP source originelle comme adresse source de la requête.**

```
from flask import Flask
from werkzeug.middleware.proxy_fix import ProxyFix

app = Flask(__name__)

# Les paramètres à Configure ProxyFix définissent la profondeur de proxy maximale
autorisée
```

```
app.wsgi_app = ProxyFix(
    app.wsgi_app,
    x_for=1,          # profondeur max pour X-Forwarded-For
    x_proto=1,        # profondeur max pour X-Forwarded-Proto
    x_host=1          # profondeur max pour X-Forwarded-Host
)

@app.route('/')
def index():
    # request.remote_addr retourne à present l'IP de X-Forwarded-For
    return f"Client IP: {request.remote_addr}"

if __name__ == '__main__':
    app.run()
```

### Exploiter l'adresse IP source au travers d'un proxy-inverse en Go

Pour un programme programmé en Go, il est recommandé de s'appuyer sur un module tiers tel que `go-forwarded` pour récupérer l'adresse correcte.

**Exemple simplifié d'usage du module *go-forwarded***

```
package main

import (
       "fmt"
       "github.com/stanvit/go-forwarded"
       "net/http"
)

func simpleHandler(w http.ResponseWriter, r *http.Request) {
       w.Header().Set("Content-Type", "text/plain")
       w.WriteHeader(http.StatusOK)
       w.Write([]byte(fmt.Sprintf("Requesting IP is %v\nHTTPS: %t\n", r.RemoteAddr,
r.TLS != nil)))
}

func main() {
       wrapper, _ := forwarded.New(
              "192.0.2.0/24, // réseau des proxy de confiance
              false, false,
              "X-Forwarded-For", "X-Forwarded-Protocol")// entêtes à utiliser
       handler := wrapper.Handler(http.HandlerFunc(simpleHandler))
       http.Handle("/", handler)
       http.ListenAndServe(":8082", nil)
}
…
```

*Référence :*

- https://pkg.go.dev/github.com/stanvit/go-forwarded

## Exemple en PHP

```
…

// retourne l'adresse IP d'origine de requête
// telle que transmise par X-Forwarded-For
function get_real_client_ip() {
    $trusted_proxies = ['192.0.2.16']; // proxy de confiance
    $proxy_header = 'HTTP_X_FORWARDED_FOR';

    if (isset($_SERVER['REMOTE_ADDR'])
        && in_array($_SERVER['REMOTE_ADDR'], $trusted_proxies)) {
        if ( !empty($_SERVER[$proxy_header]) ) {
            $proxy_list = explode(',', $_SERVER[$proxy_header]);
            $client_ip = trim($proxy_list[0]);
            // suppose que la première IP de la liste soit bien celle du client
            if (filter_var($client_ip, FILTER_VALIDATE_IP)) {
               return $client_ip;
            }
        }
     }
     return $_SERVER['REMOTE_ADDR'] ?? null;
}
```

*Référence :*

- https://stackoverflow.com/questions/11452938/how-to-use-http-x-forwarded-for-properly

## C. Contrôle d'accès : limiter les accès sur la base d'adresses IPs sources identifiées

### Filtrer sur la base de l'adresse IP avec Apache 2.4

Les modules de contrôle d'accès Apache peuvent être utilisés pour restreindre les plages d'adresse autorisées à interroger un VirtualHost ou un répertoire particulier. Les modules les plus importants pour cette partie sont : `mod_authz_core` et `mod_authz_host`.

**Exemple pour n'autoriser que les IP venant des réseaux internes :**

```
<RequireAll>
   Require all denied
   Require ip 10
   Require ip 172.16.0.0/255.240.0.0
   Require ip 192.168.0.0/16
</RequireAll>
```

*Références :*

- La documentation Apache2 sur l'authentification est disponible ici : https://httpd.apache.org/docs/2.4/howto/auth.html
- Un document avec des exemples plus complets est disponible sur le site d'apache : https://httpd.apache.org/docs/2.4/howto/access.html
- La directive à mettre en place est de type `Require` qui est documenté à la page suivante : https://httpd.apache.org/docs/2.4/mod/mod_authz_host.html#requiredirectives
-

### Filtrer sur la base de l'adresse IP avec Nginx

Dans Nginx les restrictions sur les adresses IP sources se configurent au niveau d'une entrée de configuration de type *server*.

**Exemple pour autoriser uniquement depuis des IP venant des réseaux internes :**

```
Server {
            listen 443 ;
            allow 10.0.0.0/8 ;
            allow 172.16.0.0/12 ;
            allow 192.168.0.0/16 ;
            deny all ;
     }
```

*Référence :*

- https://docs.nginx.com/nginx/admin-guide/security-controls/controlling-access-proxied-tcp/

## Filtrer sur la base de l'adresse IP avec HaProxy

HAPRoxy permet d'utiliser des ACL couplées à `http-request deny if` … pour gérer de nombreux cas de filtrage des accès.

**Exemple pour autoriser uniquement les IP d'une liste spécifique avec http-request :**

```
{
…
http-request deny if {path -i -m beg /wp-admin} !{src -f /etc/hapee-2.8/autorisees_IP.lst}
…
}
Autorisees_IP.lst
10.0.55.1/32
10.0.55.2/32
192.168.55.0/24
```

**Exemple pour autoriser uniquement les IP d'une ou plusieurs listes spécifiques avec les acl et http-request :**

```
{
…
Acl is-admin-ip src -f /etc/haproxy/admin-ip.acl
Acl is-admin-interne-ip src -f /etc/haproxy/admin-interne-ip.acl

http-request deny if ! is-admin-ip ! is-admin-interne-ip
…
}
Admin-ip.acl :
95.95.95.95/32
93.93.93.93/32
Admin-interne-ip.acl :
10.0.55.10/32
10.0.55.20/32
```

```
192.168.55.0/24
```

Mise en place d'une ACL dans le frontend pour bloquer une liste d'adresse IP de nœud Tor par exemple.

```
frontend www…
http-request drop if { src -f /etc/haproxy/lst/tor-exit-node.lst}
…
```

*Référence*

- https://www.haproxy.com/documentation/haproxy-configuration-tutorials/proxying-essentials/custom-rules/acls/

## D. Contrôle d'accès : limiter les accès par géo-filtrage

Il est préférable de mettre en place cette limitation au niveau des pare-feux externes.

Il existe une configuration au niveau de Nginx pour réaliser ce filtrage.

### Configuration avec Nginx

Nginx fonctionne avec des bases de GeoIP[70]. Il faut dans un premier temps les télécharger.

**Exemple d'installation du module sur Linux Debian ou dérivé (comme Ubuntu)**

```
Apt-get install nginx-plus-module-geoip2
```

**Chargement dans la configuration Nginx**

```
Load_module modules/ngx_http_geoip2_module.so
Load_module modules/ngx_stream_geoip2_module.so
```

**Exemple de filtrage sur la base de la zone géographique**

```
http ou stream
{…
Geoip2 GeoIP2/GeoLite2-Country.mmdb {
      Map $geoip2_data_continent_code continent code $allowed_country {
            Default no ;
            EU yes ;
…
Server {
Listen …
If ($allowed_country = no) {
      Return 403 ;
}
}
…
}
```

*Référence :*
https://docs.nginx.com/nginx/admin-guide/web-server/http-connect-proxy/

[70] Bases de géolocalisation telles que GeoIP2 et GeoLite2 de MaxMind

## Configuration avec HAProxy

Il existe 2 types de géofiltrage pour Haproxy Digital Element avec NetAcuity et MaxMind pour HAProxy Enterprise.

Ces solutions sont commerciales et n'ont pas pu faire l'objet de tests approfondis.

*Référence :*

- https://www.haproxy.com/documentation/haproxy-enterprise/enterprise-modules/geolocation/

*Géoiltrage sur la base de NetAcuity*

**Installation du module pour la version 3.3r1.**

```
Sudo apt install hapee-3.3r1-b-netacuity
```

**Configuration dans la zone « global »**

```
global
  ...
  module-load hapee-lb-netacuity.so
  netacuity-load 04 /etc/hapee-3.3/netacuity
  netacuity-cache-size 2000000
Recharger la configuration
sudo systemctl reload hapee-3.3-lb
Exemple de configuration d'un frontend avec intégration des données de géolocalisation
dans le header http.
IPv4
frontend www
  bind :80
  mode http
  http-request add-header X-NetAcuity-IPv4-1 %[src,netacuity-lookup-ipv4("src-
ip","pulse-area-codes","pulse-city","pulse-city-code","pulse-city-conf","pulse-conn-
speed","pulse-conn-type","pulse-continent-code")]
  http-request add-header X-NetAcuity-IPv4-2 %[src,netacuity-lookup-ipv4("pulse-
country","pulse-country-code","pulse-country-conf","pulse-gmt-offset","pulse-in-
dst","pulse-internal-code","pulse-latitude","pulse-longitude")]
  http-request add-header X-NetAcuity-IPv4-3 %[src,netacuity-lookup-ipv4("pulse-metro-
code","pulse-postal-code","pulse-postal-conf","pulse-region","pulse-region-
code","pulse-region-conf","pulse-timezone-name","pulse-two-letter-country")]
```

Référence :

- https://www.haproxy.com/documentation/haproxy-enterprise/enterprise-modules/geolocation/digitalelement/

*IPv6*

```
frontend www
  bind :80
  mode http
  http-request add-header X-NetAcuity-IPv6-1 %[src,netacuity-lookup-ipv6("src-ip","pulse-area-codes","pulse-city","pulse-city-code","pulse-city-conf","pulse-conn-speed","pulse-conn-type","pulse-continent-code")]
  http-request add-header X-NetAcuity-IPv6-2 %[src,netacuity-lookup-ipv6("pulse-country","pulse-country-code","pulse-country-conf","pulse-gmt-offset","pulse-in-dst","pulse-internal-code","pulse-latitude","pulse-longitude")]
  http-request add-header X-NetAcuity-IPv6-3 %[src,netacuity-lookup-ipv6("pulse-metro-code","pulse-postal-code","pulse-postal-conf","pulse-region","pulse-region-code","pulse-region-conf","pulse-timezone-name","pulse-two-letter-country")]
```

Activer cette fonction permet le filtrage au niveau de l'application en fonction des header X-NetAcuity-IPv{4|6}-{1,2,3}.

*MaxMind*

Le préalable est de récupérer des bases de géolocalisation auprès de Maxmind. Il faut aller sur le site de l'éditeur pour télécharger les bases de données `GeoIP2-City.mmdb et GeoIP2-Country.mmdb` dans `/etc/hapee-3.3/`

Attention : Ces bases sont soumises à des limitations d'usage qu'il convient de respecter.

**Installation du module pour la version 3.3r1**

```
sudo apt install hapee-3.3r1-lb-maxmind
```

**Configuration dans la section « *global* »**

```
global
  ...
  module-load hapee-lb-maxmind.so
  maxmind-load mlock_max 512000000  CITY /etc/hapee-3.3/GeoIP2-City.mmdb  COUNTRY /etc/hapee-3.3/GeoIP2-Country.mmdb  ISP /etc/hapee-3.3/GeoIP2-ISP.mmdb
  maxmind-cache-size 200000
Recharger la configuration
sudo systemctl reload hapee-3.3-lb
```

**Exemple de configuration d'un frontend avec intégration des données de géolocalisation dans l'entête HTTP.**

```
frontend www
  bind :80
  mode http
  http-request add-header X-CityName  %[src,maxmind-lookup("CITY","city","names","en")]
  http-request add-header X-ISOCode   %[src,maxmind-
lookup("COUNTRY","country","iso_code")]
  http-request add-header X-ASN       %[src,maxmind-
lookup("ISP","autonomous_system_number")]
  http-request add-header X-ASNOrg    %[src,maxmind-
lookup("ISP","autonomous_system_organization")]
```

**Mise en place d'une ACL dans le frontend pour limiter à la France et la Belgique par exemple**

```
frontend www
  bind :80
  mode http
  http-request set-var(txn.iso_code_country) src,maxmind-
lookup("COUNTRY","country","iso_code")
  http-request add-header X-ISOCode   %[var(txn.iso_code_country)]
  acl is_fr %[var(txn.iso_code_country)] -m str -i FR
  acl is_be %[var(txn.iso_code_country)] -m str -i BE
  http-request deny if !is_fr !is_be
```

**Mise en place d'une ACL dans le frontend pour bloquer un numéro d'AS spécifique par exemple**

```
frontend www
  bind :80
  mode http
  http-request set-var(txn.isp_asn) src,maxmind-
lookup("ISP","autonomous_system_number")
  http-request add-header X-ASN       %[var(txn.isp_asn)]
  acl block_asn1 %[var(txn.isp_asn)] -m str -i 44477
  acl block_asn2 %[var(txn.isp_asn)] -m str -i 210644
  acl block_asn_ZSRV %[var(txn.isp_asn)] -m str -i 197414
  http-request deny if block_asn1 || block_asn2 || block_asn_ZSRV
```

Référence :

- https://www.haproxy.com/documentation/haproxy-enterprise/enterprise-modules/geolocation/maxmind/

## Exemple de configuration d'un pare-feu StormShield

La documentation en lien explique la mise en place d'un filtrage des requêtes en fonction de critères géographiques sur leur adresse source par : pays, régions ou continents sur des équipements StormShield.

Cette fonction permet de limiter les accès à des zones précises géographiques ou de réduire les sources dites *à risques* (nœud de sortie Tor, VPN connus…)

La réciproque est possible en mettant en place un filtrage des destinations en fonction de critères géographiques, en suivant la documentation en référence.

*Références :*

- Source / https://documentation.stormshield.com/SNS/v5/fr/Content/User_Configuration_Manual_SNS_v5/Filtering_and_NAT/Filtering_tab_filter_table_source.htm
- Destination / https://documentation.stormshield.com/SNS/v5/fr/Content/User_Configuration_Manual_SNS_v5/Filtering_and_NAT/Filtering_tab_filter_table_destination.htm

## Exemple de configuration de pare-feu Fortinet

**Vérification que la base géographique des adresses IP est présente :**

```
# diagnose autoupdate versions | grep -A 6 "IP Geography DB"
```

**NOTE :** Une licence valide permet la mise à jour régulière de la base de données.

Création d'un objet d'adresse IP géographique

1. Aller dans Politique et Objets > Adresses
2. Créer une nouvelle adresse
3. Dans la catégorie choisir adresse et entrer son nom
4. Dans la partie type choisir géographique
5. Dans la partie région / pays choisir un pays par exemple
6. Mettre en place les règles de filtrage
7. Entrer en commentaire l'explication de la règle Autorise pays France / Interdit pays Russie
8. Cliquer sur OK

**Test de commande avec diagnose**

```
diagnose firewall ipgeo [country-list | ip-list | ip2country | override | copyright-notice]
```

*Références :*

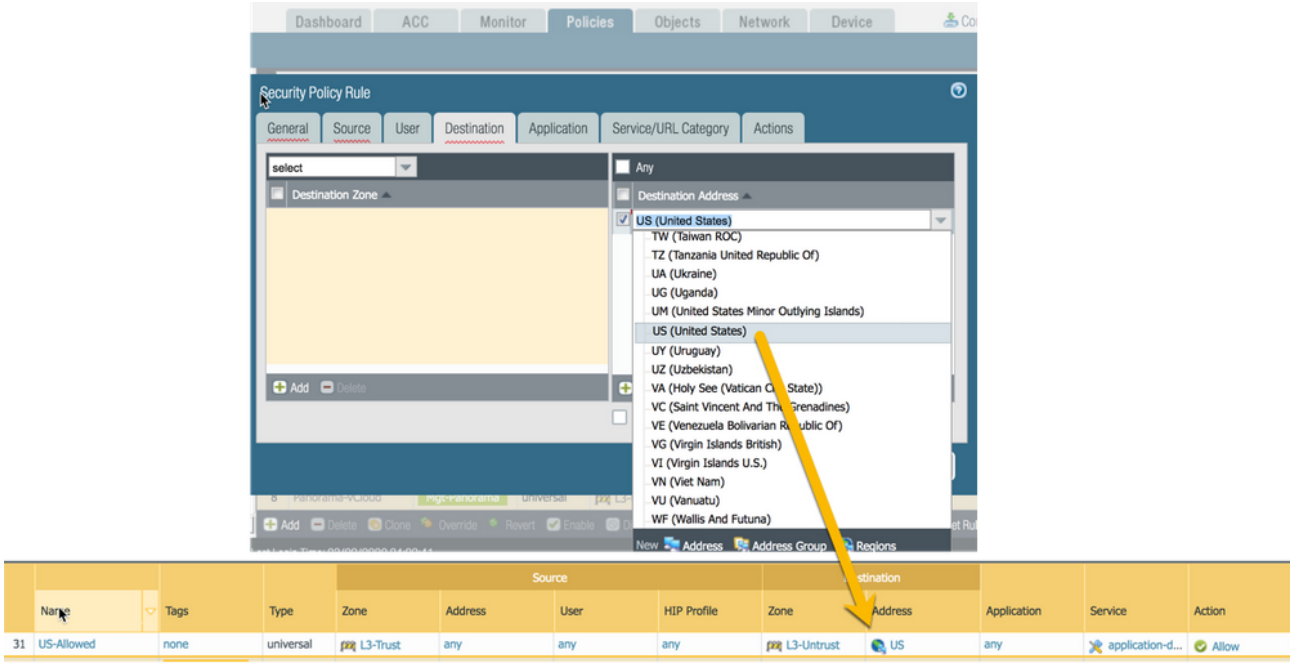
- https://community.fortinet.com/t5/FortiAuthenticator/Technical-Tip-Best-practices-on-hardening-FortiAuthenticator/ta-p/274443
- https://community.fortinet.com/t5/FortiGate/Technical-Tip-Restricting-SSL-VPN-connectivity-from-certain/ta-p/191997
- https://docs.fortinet.com/document/fortigate/7.4.1/administration-guide/286826/geography-based-addresses

## Exemple de configuration de pare-feu Palo Alto

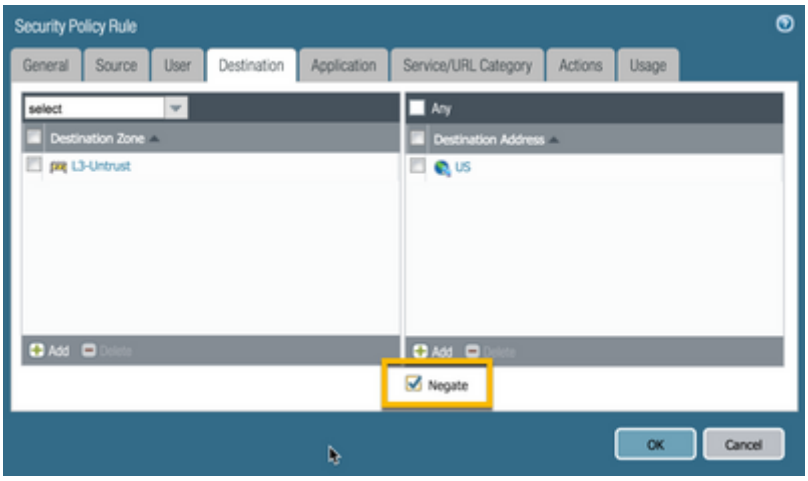
La mise en place de règle de blocage géographique est disponible directement dans les créations des règles.

Dans Security Policy Rule, au niveau de Source ou Destination

Sélectionner la région ou le pays pour autoriser ou interdire les accès.




Les mêmes fonctions permettent de bloquer des catégories telles que les proxies anonymisants.




*Références :*

- https://live.paloaltonetworks.com/t5/community-blogs/geolocation-and-geoblocking/ba-p/315433
- https://knowledgebase.paloaltonetworks.com/KCSArticleDetail?id=kA10g000000ClJWCA0#:~:text=This%20document%20describes%20how%20to%20create%20a%20security,the%20region%27s%20name%20mentioned%20on%20the%20security%20policy.
- https://knowledgebase.paloaltonetworks.com/KCSArticleDetail?id=kA10g000000ClbHCAS
- https://knowledgebase.paloaltonetworks.com/KCSArticleDetail?id=kA10g000000ClRnCAK
- https://knowledgebase.paloaltonetworks.com/KCSArticleDetail?id=kA10g000000ClVYCA0
- https://knowledgebase.paloaltonetworks.com/KCSArticleDetail?id=kA10g000000ClFOCA0

## Exemple de configuration d'un pare-feu OpnSense

Dans un premier temps, il faut créer un compte auprès de Maxmind et récupérer la clef de licence.

Enfin récupérer la base directement sur Maxmind avec la clef de licence en créant un lien :

```
https://download.maxmind.com/app/geoip_download?edition_id=GeoLite2-Country-CSV&license_key=<LICENSE_KEY>&suffix=zip
```

Aller dans les Alias et sélectionner GeoIP puis copier-coller le lien créé avec la clef de licence.

Il faut créer un Alias après en utilisant les options GeoIP à partir de la base MaxMind ou IPInfo, par exemple country = France.

Créer une règle d'autorisation pour l'alias GeoIP-France au niveau du champ source pour autoriser les connexions depuis des adresses IP localisées en France.

Dans le même esprit des règles de blocage (DROP) peuvent être créées en s'appuyant sur les bases filtrage de Spamhaus.

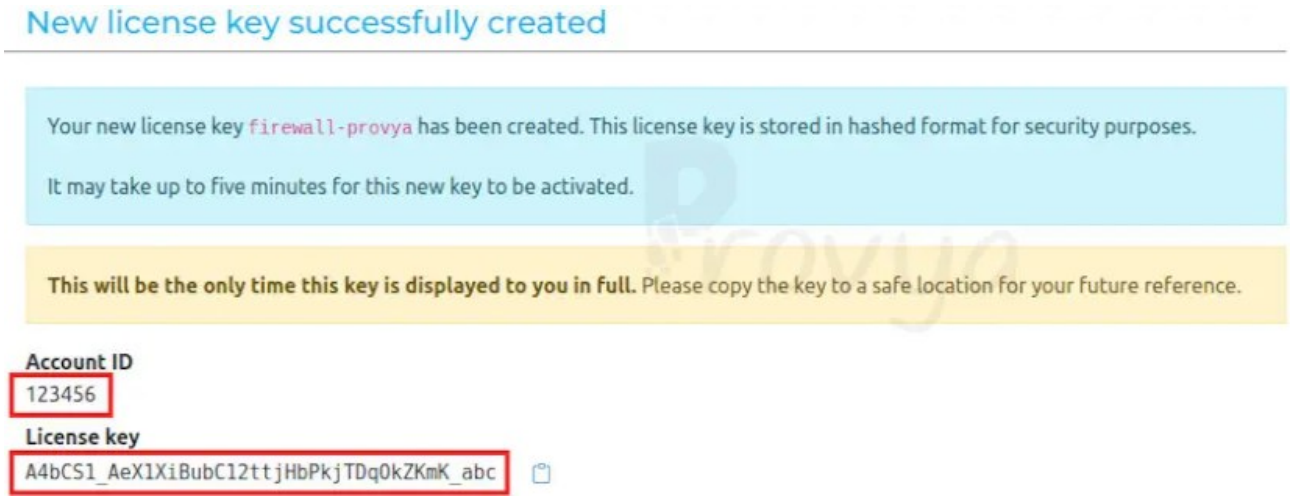
*Références :*

- https://docs.opnsense.org/manual/how-tos/drop.html
- https://docs.opnsense.org/manual/how-tos/maxmind_geo_ip.html
- https://docs.opnsense.org/manual/aliases.html

## Exemple de configuration d'un pare-feu PfSense

*Utilisation du package pfBlockerNG*

Dans un premier temps, il faut créer un compte auprès du service Maxmind et récupérer la clef de licence ainsi que le numéro de compte.

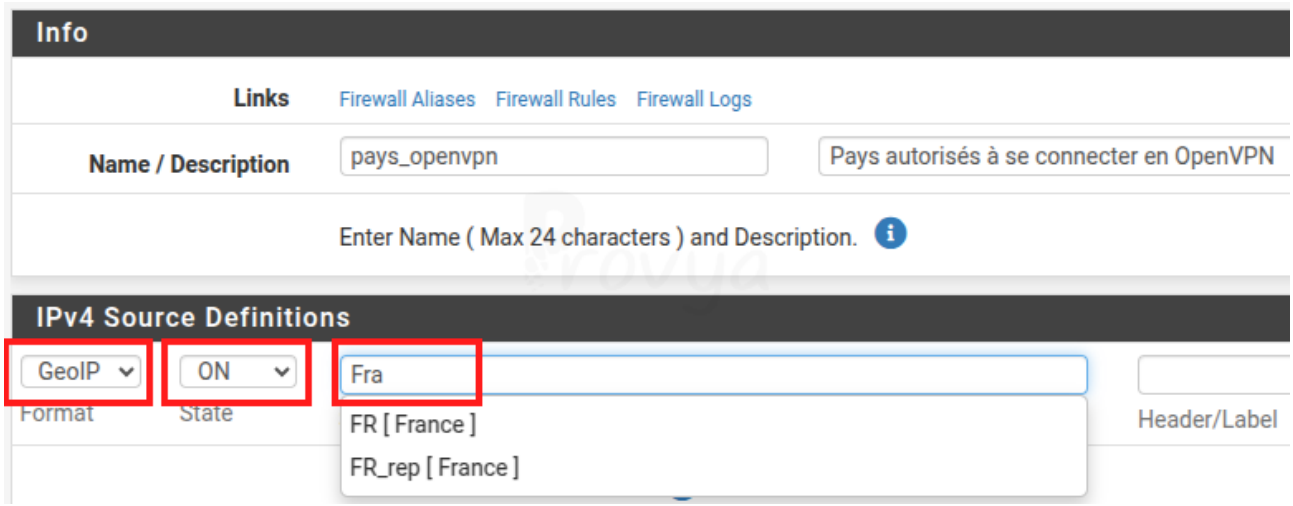



Mettre à jour l'identifiant et la clef de licence dans pfBlockerNG.

Puis configurer la mise à jour de la base dans la partie *Update* de pfBlockerNG et la lancer pour récupérer la base. On peut alors trouver dans pfBlockerNG dans la section *IP* puis *GeoIP*, la liste des continents et des actions possibles (Deny, permit, …) sur les flux entrants ou sortants. Cette approche est rapide et simple, mais couvre directement par continent et non finement.

Pour plus de flexibilité et finesse, il est préférable de créer des alias avec un liste de pays par exemple.

Aller dans pfBlockerNG / IP / IPv4 puis ajouter un alias avec un format GeoIP, Statut On et France :




Quand l'alias est configuré, il apparaît dans Firewall / Aliases / URLs avec une URL en 127.0.0.1.

Il devient alors possible de créer des règles de filtrage utilisant cet alias pour filtrer des accès.

*Références :*

- https://docs.netgate.com/pfsense/en/latest/packages/pfblocker.html

- https://provya.net/?d=2025/03/11/10/05/00-pfsense-filtrage-geographique-avec-pfblockerng

## Exemple avec F5 BigIP

Les options XFF et Trust XFF dans la partie Security Policy doivent être appliquée en plus de la géolocalisation pour une meilleure performance.

Deux modes de filtrage sont supportés :

- Interdit par défaut et une liste autorisée
- Autorisée par défaut et une liste d'interdit

Par exemple, il existe une liste de proxies anonymisants, ***anonymous proxies***, à mettre dans la liste `Disallow Access` ou la liste **N/A** pour les localisations qui ne sont pas assignées ou ne respectant pas la RFC1918.

Aller dans Securité > Application de sécurité > Politiques de sécurité > Liste politiques

Sur l'une des règles d'accès par exemple aller dans la *Configuration générale,* puis *Configuration avancée,* et sélectionner *GeoIP*.

Cet écran permet d'autoriser ou d'interdire des zones géographiques en lien avec la règle sélectionnée.

Sauvegarder puis appliquer la politique.

*Références :*

- https://my.f5.com/manage/s/article/K79414542
- https://my.f5.com/manage/s/article/K15042
- https://my.f5.com/manage/s/article/K11176
- https://my.f5.com/manage/s/article/K12866

## E. Contrôle d'accès : limiter les heures de connexion

Dans certain cas, il peut être intéressant de limiter les accès en dehors des heures ouvrées à tout ou partie des utilisateurs d'une application.

Différents points de l'architecture sont propices à l'application de cette restriction :

- Au niveau des pare-feux internes en amont de votre application
- Au niveau de la configuration de votre application directement
- Au niveau de la configuration de votre gestionnaire d'identité comme Active Directory par exemple

### Configuration des horaires de connexion dans Active Directory :

C'est l'attribut `logonHours` qui permet de gérer les heures de connexions autorisées ou interdites. Il est possible de modifier cet attribut :

- Par compte spécifique
- Par UO de comptes
- Par GPO
- Par script Powershell

*Référence :*

- https://www.it-connect.fr/chapitres/active-directory-powershell-horaire-acces/

Si votre VPN ou certains accès sont dépendants de l'Active Directory alors la limitation horaire permet de réduire directement les accès des utilisateurs visés par des changements de `logonhours`.

*Référence :*

- https://blog.soteria.io/times-up-enabling-logon-hours-via-conditional-access-policies-a3e4e9428dbc
- https://activedirectorypro.com/how-to-set-logon-hours-for-active-directory-users/
- https://learn.microsoft.com/en-us/answers/questions/2200181/how-do-i-set-active-logon-hours-for-a-domain-in-gr

Une configuration EntraID permet d'appliquer des mesures équivalentes.

### Configuration de règles en fonction des horaires sur pare-feu :

La plupart des pare-feux modernes permettent de positionner des règles de filtrage actives à des tranches horaires spécifiques. Utiliser ces règles pour restreindre des accès à un VPN ou à une l'interface web d'une application aux heures d'usage légitime aide les défenseurs.

L'avantage de cette solution est qu'elle ne change rien aux configurations. Elle limite le temps d'exposition à des périodes plus propices à une réponse rapide ou une analyse par les équipes IT et de sécurité en heures ouvrées. Elle permet même en cas d'absence de configuration dans l'application ou le fournisseur d'identité de malgré tout limiter les accès à l'application.

**Point d'attention :** Une règle pour un VPN-SSL peut être créée pour les utilisateurs et configurée pour n'être active que de 6h à 22h par exemple. Une vérification au niveau des tunnels actifs devrait alors

être réalisée. Il pourrait être judicieux de créer des tunnels d'une durée de vie limitée en durée, et ne pouvant plus être ouverts après une certaine heure. Ce point des sessions actives s'applique pour tous types de connexion Web, SSL, IPSec…

*Références*

- Fortinel Fortigage : https://community.fortinet.com/t5/FortiGate/Technical-Tip-How-to-apply-a-schedule-to-a-firewall-policy/ta-p/193508
- Palo-Alto NGFW : https://docs.paloaltonetworks.com/network-security/security-policy/administration/objects/schedules
- Stormshield : https://documentation.stormshield.com/SNS/v5/fr/Content/User_Configuration_Manual_SNS_v5/Network_objects/Different_types_of_objects_-_Time_object.htm
- PfSense : https://docs.netgate.com/pfsense/en/latest/firewall/time-based-rules.html
- OpnSense : https://docs.opnsense.org/manual/firewall.html#id1 (rechercher le terme "Schedule")
- F5 BigIP : https://techdocs.f5.com/kb/en-us/products/big-ip-afm/manuals/product/big-ip-network-firewall-policies-and-implementations-14-1-0/04.html

## Exemple d'utilisation de cron et de ufw sous système Linux

**Attention :** Le pare-feu Linux peut être géré par différents outils : UFW, firewalld, iptables, nftables. Il est recommandé de standardiser l'outil gérant les configurations de filtrage et ne pas les mélanger.

Les exemples ci-dessous sont donnés avec l'outil UFW.

**Commandes d'ouverture de flux vers 80/443**

```
Ufw allow 80/tcp

Ufw allow 443/tcp
```

**Commandes de blocage des flux TCP sur les ports 80/443**

```
Ufw drop 80/tcp
Ufw drop 443/tcp
```

**Commandes de suppression de règles d'autorisation**

```
Ufw delete allow 80/tcp
Ufw delete allow 443/tcp
```

**Commandes de suppression de règle de drop**

```
Ufw delete drop 80/tcp
Ufw delete drop 443/tcp
```

Si l'ouverture est à 6h00 UTC, il faut supprimer les règles de drop et mettre en place celles d'autorisation.

Si la fermeture est à 21h00 UTC, il faut supprimer les règles d'autorisation et mettre en place celles de drop.

**Format cron pour du lundi au vendredi de 6h à 21h avec fermeture le week-end :**

```
0 6 * * 1-5 ouverture-ufw-Web.sh
0 21 * * 1-5 fermeture-ufw-Web.sh
```

Ne pas oublier de faire un log de l'exécution pour vérifier la bonne mise en place de la règle.

**Note :** Il faut s'assurer que l'option `timezone` soit homogène sur tous les équipements de la zone technique considérée.

**Point d'attention** : Sil des accès de maintenance doivent avoir lieu hors des heures ouvertes (sauvegardes, mises à jour…) il faut s'assurer que la configuration ne les empêche pas. Il est recommandé de gérer des exclusions au moyen de règles d'autorisation pour des adresses IP spécifique (exemple : `ufw allow from 192.168.66.5 to any port 80` permet d'autoriser les connexions depuis 192.168.66.5 vers le port 80 indépendamment des autres filtrages).

## F. Limitation d'exfiltration : nombre de requêtes sur une fenêtre temporelle glissante

Si les comportements d'utilisateurs sont réguliers, il est conseillé de limiter le nombre de requêtes par session. Même si l'attaquant détecte la présence de la mesure, il se voit obligé de multiplier les sessions pour exfiltrer de l'information ce qui se détecte.

### Limiter le nombre de requêtes sur fenêtre temporelle sur HAProxy

Le serveur HAProxy permet d'effectuer des limitations par tranche temporelle.

*Références*

- https://oneuptime.com/blog/post/2026-03-20-haproxy-rate-limiting-ipv4/view
- https://www.haproxy.com/blog/four-examples-of-haproxy-rate-limiting

La configuration se fait dans la `balise frontend http_front` du fichier de configuration HAProxy.

**Création d'une table stockant les requêtes par adresse IP sur 10s pendant 60s par configuration dans le**

```
…
stick-table type ip size 100k expire 60s store htt_req_rate(10s)
…
```

**Mise en place du suivi de consommation par IP et de la limitation.**

```
…
http-request track-sc0 src
…
Deny page Haproxy too many request (429) pour un nb de req > 10 / 10s soit environ 1/s
http-request deny deny_status 429 if { sc_http_req_rate(0) gt 10 }
….
```

### Limiter le nombre de requêtes sur une fenêtre temporelle avec Nginx

Avec le serveur web Nginx cette possibilité existe aussi.

*Références*

- https://oneuptime.com/blog/post/2026-03-20-nginx-rate-limiting-prevent-abuse/view
- https://blog.nginx.org/blog/rate-limiting-nginx

**Configuration Nginx**

```
http { … }

limit_req_zone $binary_remote_addr zone=website_limit:10m rate=10r/s

Config site server { ... }
```

```
location / {

1limit_req zone=website_limit burst=15 nodelay | delay=10;
…
```

## G. Supervision : détection par nombre de tentatives d'accès

Dans la plupart des cas une détection par seuil est un moyen de lever des alertes et / ou mettre en place des mécanismes automatiques de protection désactivation de compte, bannissement d'une adresse IP, etc. Ce mécanisme est applicable aussi au métier. Le format le plus utilisé pour les règles de détection est Sigma. Ce format permet de passer d'une règle dans un format abstrait (Sigma) vers des signatures spécifiques à un outil (SIEM, XDR, EDR, etc.).

Exemple de règle Sigma par seuil avec l'exemple du cas d'un dossier sensible. Ceci est un exemple de déclinaison au format de Sigma d'une règle définie par les métiers pour détecter des anomalies. Il doit être contextualisé à votre entité, les technologies employées et les moyens de détection en place.

```
title: Dossier sensible accès
name: Dossier_sensible_access
description: Détecter le dépassement d'un seuil métier d'accès à des dossiers sensibles
par jour
logsource:
product: apache
      service: dossier_sensible
detection:
      selection:
            userIdentity.type: user_secretaire
      filter:
            eventType: access_dossier_sensible
falsepositives:
Revue annuelle des dossiers sensibles quantitatifs
level: medium
---
title: Multiple accès dossiers sensibles exfiltration
correlation:
      type: event_count
      rules:
Dossier_sensible_access
      Group-by:
user_secretaire
timespan: 24h
condition:
      gte: 50
```

La conversion vers un SIEM, tel que Splunk, s'effectue avec la commande suivante :

```
sigma convert -t splunk -p splunk_dossier_sensible \
    rules/dossier_sensible_access_exfiltration.yml
```

La règle dans l'outil Splunk pourrait être :

```
source="Apache:dossier_sensible" EventCode = "access_dossier_sensible"
```

```
| bin _time span=24h
| stats count as event_count by _time user_secretaire
| search event_count >= 50
```

*Référence :*

- https://sigmahq.io/