Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE

PREMIER MINISTRE

Secrétariat général
de la défense
et de la sécurité nationale

*Agence nationale de la sécurité
des systèmes d'information*

Paris, le 9 octobre 2014

N° DAT-NT-19/ANSSI/SDE/NP

Nombre de pages du document
(y compris cette page) : 32

Note technique

Recommandations de sécurité concernant l'analyse des flux HTTPS

Public visé:

Développeur	✓
Administrateur	✓
RSSI	✓
DSI	✓
Utilisateur	

Informations

Avertissement

Ce document rédigé par l'ANSSI présente les « **Recommandations de sécurité concernant l'analyse des flux HTTPS** ». Il est téléchargeable sur le site www.ssi.gouv.fr. Il constitue une production originale de l'ANSSI. Il est à ce titre placé sous le régime de la « Licence ouverte » publiée par la mission Etalab (`www.etalab.gouv.fr`). Il est par conséquent diffusable sans restriction.

Ces recommandations sont livrées en l'état et adaptées aux menaces au jour de leur publication. Au regard de la diversité des systèmes d'information, l'ANSSI 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 l'ANSSI 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.

Personnes ayant contribué à la rédaction de ce document:

Contributeurs	Rédigé par	Approuvé par	Date
LRP, LAM, BAI, MRR	BSS	SDE	9 octobre 2014

Évolutions du document :

Version	Date	Nature des modifications
1.0	1er octobre 2014	Version initiale
1.1	9 octobre 2014	Corrections mineures

Pour toute remarque:

Contact	Adresse	@mél	Téléphone
Bureau Communication de l'ANSSI	51 bd de La Tour-Maubourg 75700 Paris Cedex 07 SP	communication@ssi.gouv.fr	01 71 75 84 04

Table des matières

1 Préambule

Le protocole HTTPS correspond à la déclinaison sécurisée de HTTP encapsulé à l'aide d'un protocole de niveau inférieur nommé TLS [1] (anciennement SSL [2]). Ce protocole est conçu pour protéger en confidentialité et en intégrité des communications de *bout en bout* (entre un client et un serveur). Il apporte également des fonctions d'authentification du serveur, mais aussi optionnellement du client.

La protection de *bout en bout* qu'apporte TLS est a priori incompatible avec d'autres exigences de sécurité complémentaires visant à inspecter le contenu des échanges. L'analyse d'un contenu (web par exemple) sécurisé à l'aide de TLS, peut toutefois se justifier afin de s'assurer que les données provenant d'un réseau non maîtrisé (Internet par exemple) ne représentent pas une menace pour le système d'information interne. Les architectures visant à déchiffrer les flux TLS, pour permettre leur analyse, « tordent » donc le modèle pour lequel ce protocole est conçu.

Pour pouvoir mettre en œuvre le déchiffrement de flux TLS de façon maîtrisée, il est nécessaire de disposer, entre autres, d'un niveau de connaissance suffisant dans les deux domaines spécifiques et évolutifs que sont les IGC [3] et la cryptographie. **Quel que soit le contexte, la mise en place de mécanismes de déchiffrement HTTPS présente des risques dans la mesure où cette opération entraîne la rupture d'un canal sécurisé et expose des données en clair au niveau de l'équipement en charge de l'opération. Lorsqu'un tel déchiffrement est nécessaire, sa mise en œuvre doit s'accompagner de beaucoup de précautions et se faire uniquement après validation de la direction des systèmes d'information voire d'une autorité de niveau supérieur.**

Cette note présente donc les recommandations d'ordre technique à suivre lorsque l'analyse des flux HTTPS échangés entre un système d'information maîtrisé et des réseaux externes est indispensable. Deux scénarios sont présentés. Le premier, en théorie plus rare, détaille le cas où les flux HTTPS sont déchiffrés après avoir été initiés par des clients présents sur le système d'information en direction des sites web externes. Le second, plus fréquent, présente le cas où des clients externes souhaitent se connecter à l'aide du protocole HTTPS à des sites web hébergés au sein d'un système d'information maîtrisé. Ce document n'a pas pour objectif de décrire à nouveau en détail le fonctionnement du protocole TLS ; la publication intitulée « SSL/TLS : état des lieux et recommandations [4] » disponible sur le site de l'ANSSI présente ce protocole et les problématiques associées. Par contre, certains aspects juridiques relatifs au déchiffrement de flux HTTPS sont abordés à la fin de ce document.

1. *Transport Layer Security.*
2. *Secure Socket Layer.*
3. Infrastructure de Gestion de Clés.
4. http://www.ssi.gouv.fr/IMG/pdf/SSL_TLS_etat_des_lieux_et_recommandations.pdf.

2 Généralités

2.1 Rappels sur TLS

Voici un résumé de la séquence permettant l'établissement d'un tunnel TLS. Ce schéma illustre un cas classique [5] ; il a pour principal objectif de faire apparaître les éléments nécessaires à la compréhension de ce document.

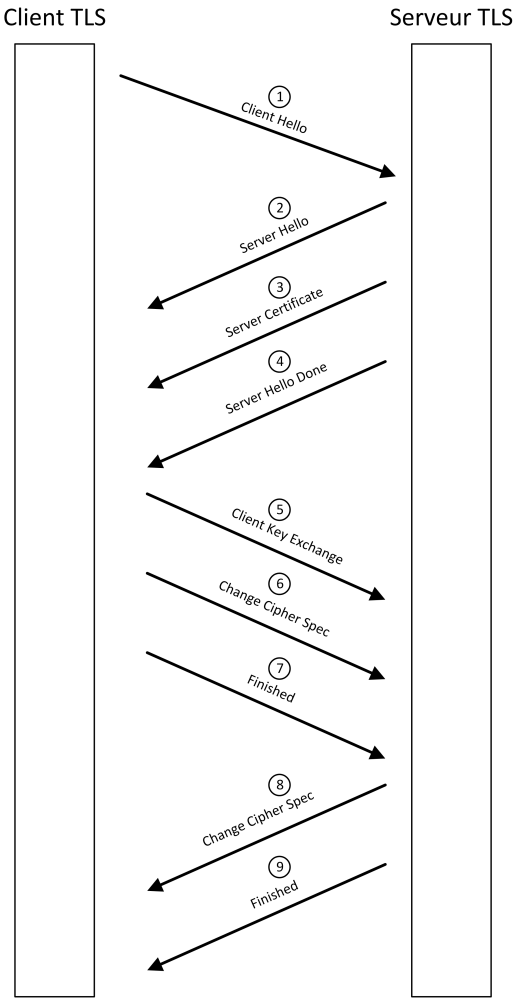

FIGURE 1 – Établissement d'une session TLS

Détail des échanges :

1. le client initie une requête en direction du serveur en envoyant un message de type *Client Hello*.
   Ce message contient en particulier les éléments suivants :
   – les suites cryptographiques (*ciphersuite*) supportées par le client ;
   – la version la plus élevée de SSL/TLS qu'il supporte ;
   – les algorithmes de compression qu'il supporte ;
   – optionnellement, les informations relatives aux extensions qu'il utilise [6].
2. le serveur répond par un message de type *Server Hello*.
   Ce message contient en particulier les éléments suivants :

5. C'est-à-dire sans authentification du client, et en utilisant un mécanisme d'échange de clés par chiffrement *RSA*.
6. Par exemple *SNI* (*Server Name Indication*).

- la suite cryptographique sélectionnée par le serveur parmi celles qu'il supporte et celles proposées par le client ;
- l'algorithme de compression sélectionné par le serveur parmi ceux qu'il supporte et ceux proposés par le client ;
- les informations relatives aux extensions utilisées par le client (acceptation ou non de ces extensions).

3. le serveur envoie ensuite un message de type *Server Certificate* au client en lui fournissant son certificat au format *X.509* (contenant sa clé publique) pour que celui-ci puisse l'authentifier ;
4. le serveur envoie un message de type *Server Hello Done* pour signifier au client qu'il a terminé cette première séquence et qu'il est en attente d'une réponse de sa part ;
5. après avoir validé le certificat du serveur, le client répond par un message de type *Client Key Exchange* ; celui-ci contient le *Pre-master Secret* chiffré à l'aide de la clé publique du serveur. Ce secret sera utilisé par les deux parties pour générer le *Master Secret* qui permet d'obtenir par dérivation les clés de session utilisées pour sécuriser les données échangées après l'établissement du tunnel TLS ;
6. le client poursuit par l'envoi d'un message de type *Change Cipher Spec* chiffré à l'aide de l'algorithme de chiffrement symétrique négocié précédemment. Ce message indique que les données que le client transmettra par la suite au serveur seront également chiffrées ;
7. le client termine par un message de type *Finished* ;
8. le serveur répond par l'envoi d'un message de type *Change Cipher Spec* chiffré à l'aide de l'algorithme de chiffrement symétrique négocié précédemment. Ce message indique que les données que le serveur transmettra par la suite au client seront également chiffrées ;
9. le serveur termine par un message de type *Finished*.

À la suite de cette séquence, le trafic échangé entre le client et le serveur sera protégé à l'aide du protocole TLS configuré en fonction des paramètres définis au cours de la séquence de négociation.

Attention : La séquence d'établissement présentée pose problème car la confidentialité des clés de session utilisées pour protéger les communications TLS dépend de la clé privée du serveur TLS. En effet, si cette dernière est compromise par un attaquant, qui aurait également capturé le trafic TLS, il sera en mesure d'obtenir le *Pre-master Secret*, le *Master Secret* ainsi que l'ensemble des clés de session.
Pour éviter ce problème, le mécanisme d'échange de clés utilisé lors de l'établissement de la session TLS doit permettre la *PFS* (*Perfect Forward Secrecy*). Cette propriété permet de générer des clés de session sans que la confidentialité de celles-ci ne dépende de la clé privée du serveur, cette dernière n'est alors utilisée que pour authentifier le serveur vis-à-vis des clients. Dans cette configuration, la compromission de la clé privée du serveur ne permet donc pas le déchiffrement de sessions TLS enregistrées au préalable, mais la clé peut néanmoins être utilisée pour usurper l'identité du serveur (attaque active dite de « l'homme du milieu ») dans le but de déchiffrer le trafic futur. Les suites cryptographiques qui permettent la *PFS* reposent sur un échange de clés de type Diffie-Hellman éphémère (DHE ou ECDHE figure dans le nom de la suite au format *IANA*[7]) et utilisent un autre algorithme de signature (RSA par exemple) pour réaliser l'authentification des parties.

7. *Internet Assigned Numbers Authority.*

2.2 Implémentation de TLS

Les applicatifs TLS (clients et serveurs) sont généralement développés à partir de bibliothèques TLS existantes. Certaines implémentations comme OpenSSL et NSS sont libres alors que d'autres sont propriétaires. La bibliothèque Schannel, par exemple, est développée par Microsoft. Lorsqu'une vulnérabilité est détectée dans une bibliothèque, celle-ci affecte l'ensemble des applications basées sur une des versions vulnérables du composant. La présence d'une vulnérabilité au sein d'une bibliothèque TLS peut ainsi avoir des conséquences extrêmement importantes en termes de sécurité. Cela peut conduire à l'affaiblissement du niveau de protection qu'apporte TLS (divulgation d'informations, perte d'intégrité, usurpation d'identité).

R1

Il est impératif d'appliquer rapidement les correctifs de sécurité associés à une bibliothèque ou à un applicatif TLS dès lors qu'ils corrigent des vulnérabilités jugées importantes.

2.3 Référentiel Général de Sécurité

Le Référentiel Général de Sécurité (RGS), disponible sur le site de l'ANSSI [8] définit les règles à respecter concernant la sécurisation des échanges électroniques entre les usagers et les autorités administratives et entre autorités administratives. Au-delà de son périmètre d'applicabilité strict, le RGS fournit des recommandations et des métriques qui font référence et qui sont utilisables dans un contexte plus large. Deux annexes composant le RGS sont particulièrement pertinentes lorsqu'il s'agit de mettre en œuvre TLS :

- l'annexe B1 [9] précise les règles et les recommandations à respecter lorsque des mécanismes cryptographiques sont employés ;
- l'annexe A4 [10] rassemble les règles relatives aux formats de certificats.

2.4 Protection des clés privées

Une mise en œuvre sécurisée de TLS requiert l'emploi d'au moins un bi-clé côté serveur ; celui-ci est composée d'un couple certificat/clé privée. Le niveau de sécurité d'un canal TLS dépend donc en partie des mesures de protection qui sont appliquées à la clé privée par l'équipement qui l'héberge. Si cette clé était compromise par une personne mal intentionnée, celle-ci pourrait être en mesure de déchiffrer des échanges enregistrés avant le vol de la clé (si la *PFS* n'est pas activée) ou d'usurper l'identité du serveur légitime après le vol. Il est donc primordial de mettre en place des mécanismes de protection logiciels ou matériels au niveau des équipements qui hébergent les bi-clés.

R2

Les clés privées associées aux certificats doivent être correctement protégées.

La solution la plus sécurisée consiste à stocker les données cryptographiques dans un composant matériel de type *HSM* [11] pouvant dialoguer de façon sécurisée (à l'aide de l'API PKCS#11 par exemple) avec l'équipement qui termine les tunnels TLS.

8. http://www.ssi.gouv.fr/fr/reglementation-ssi/referentiel-general-de-securite/.
9. http://www.ssi.gouv.fr/IMG/pdf/RGS_v-2-0_B1.pdf.
10. http://www.ssi.gouv.fr/IMG/pdf/RGS_v-2-0_A4.pdf.
11. *Hardware Security Module*. La liste des *HSM* certifiés par l'ANSSI est disponible à l'adresse : http://www.ssi.gouv.fr/fr/produits-et-prestataires/.

2.5 Validité des certificats

L'acceptation de certificats invalides (date de validité échue, certificat révoqué, etc.) est un problème de sécurité ouvrant la voie à des attaques pouvant compromettre la sécurité des communications.

R3

> Quel que soit le contexte de mise en œuvre de TLS, des mécanismes de vérification automatiques doivent être mis en place afin de s'assurer que les certificats employés sont bien valides.

Voici un aperçu des deux principaux mécanismes de vérification automatiques de révocation :

- *CRL*[12] : une *CRL* est un fichier contenant la liste des certificats révoqués par une autorité de certification (ou AC). Le maintien en ligne d'une *CRL* à jour fait partie des fonctions que doit assurer une IGC. L'emplacement de la *CRL* associée à une AC doit être renseigné dans le champ *CRLDP*[13] de chaque certificat émis par cette AC ;
- *OCSP*[14] : le protocole *OCSP* fonctionne en mode client-serveur. Il permet à un client de vérifier en ligne la validité d'un certificat en interrogeant des serveurs *OCSP* (appelés « répondeurs »). Si une IGC met à disposition un service *OCSP*, l'emplacement des répondeurs associés doit être renseigné dans le champ *AIA*[15] de chaque certificat émis par l'AC.

D'autres solutions plus récentes existent, elles visent essentiellement à améliorer l'efficacité de ces deux mécanismes, c'est le cas notamment de *OCSP Stapling*[16] ou de *CRLSets* (initiative de Google).

12. *Certificate Revocation List* ou « liste de révocation de certificats ».
13. *CRL Distribution Point* ou « point de distribution de la *CRL* ».
14. *Online Certificate Status Protocol* ou « protocole de vérification en ligne de certificat ».
15. *Authority Information Access* ou « accès aux informations de l'autorité ».
16. Littéralement « agrafage *OCSP* ».

3 Traitement des flux HTTPS sortants

Cette section a pour objectif de présenter les possibilités de traitement des flux HTTPS dont les requêtes sont initiées à partir de clients se trouvant sur un système d'information maîtrisé et qui sont destinées à établir un canal sécurisé avec des serveurs web externes (situés sur Internet par exemple).

3.1 Architecture

L'usage de serveurs mandataires (ou proxy) HTTP est une bonne pratique d'architecture permettant de s'assurer que l'ensemble des clients d'un système d'information passent par un point de contrôle unique pour pouvoir accéder à des sites web hébergés sur d'autres réseaux. C'est généralement au niveau de ce type d'équipement que sont analysés les flux HTTPS sortants, si la solution de proxy employée dispose des fonctionnalités permettant l'analyse de ce type de flux. Un proxy web peut être positionné de plusieurs façons vis-à-vis de ses clients ; la description des architectures les plus répandues dépasse le cadre de ce document.

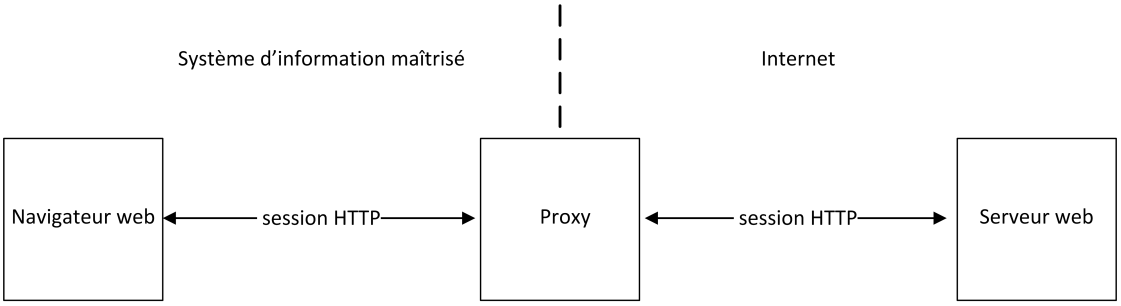

FIGURE 2 – Schéma général d'un proxy HTTP

3.2 Traitement des flux HTTPS par déchiffrement

Ce paragaphe détaille le cas où le proxy est en mesure de disposer du trafic en clair échangé entre le client et le serveur web cible. Cela est possible lorsque le proxy peut « duper » le client en interceptant la connexion TLS qu'il initie en direction du serveur web cible. Le proxy doit pour cela intégrer un serveur TLS pour pouvoir être le point de terminaison des sessions HTTPS. Le proxy joue ensuite le rôle de client vis-à-vis du serveur cible avec lequel il établit un autre tunnel TLS pour sécuriser les échanges qui transitent sur Internet. Ce double rôle permet ainsi au proxy de disposer du trafic HTTP non chiffré entre les deux tunnels TLS établis.

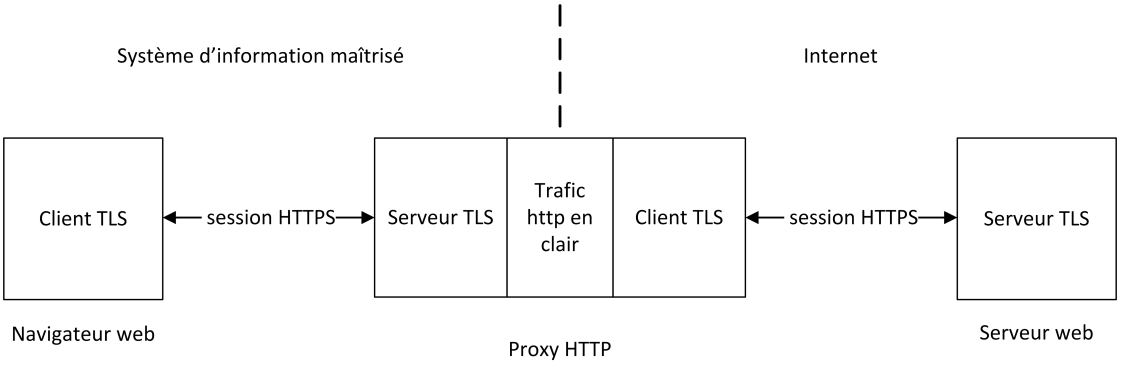

FIGURE 3 – Traitement des flux HTTPS sortants par déchiffrement

Le proxy jouant le rôle de serveur TLS pour les clients, il doit s'authentifier auprès de ces derniers en leur présentant un certificat valide lorsqu'ils initient une connexion HTTPS. Le serveur web cible devra de la même façon s'authentifier auprès du proxy en lui présentant son propre certificat. Une fois les deux tunnels TLS établis, le proxy relaie au serveur cible les demandes qu'il reçoit de ses clients. Le déchiffrement HTTPS déporte ainsi au niveau du client TLS intégré au proxy les vérifications des éléments transmis par le serveur externe (certificat, paramètres TLS, etc.) nécessaires à l'établissement du tunnel TLS.

3.2.1 Les enjeux du déchiffrement

Avant de mettre en place des mécanismes de déchiffrement au niveau d'un proxy web, il est nécessaire de bien comprendre les avantages, les inconvénients et les problématiques que cela induit.

La possibilité de disposer du trafic HTTP en clair au niveau d'un proxy web procure plusieurs avantages :

- il est possible d'analyser le trafic HTTP afin de protéger le client de menaces émanant du serveur web cible : contenus inappropriés, fichiers malveillants, etc. ;
- il est possible de contrôler le contenu des données échangées entre le client et le serveur afin de s'assurer que les flux HTTPS ne sont pas utilisés pour faire sortir du système d'information des données confidentielles. L'analyse doit être réalisée en limitant autant que possible l'exposition des données à caractère personnel des clients (se reporter au §3.2.2.3) ;
- il est possible d'appliquer la même politique de journalisation que celle mise en œuvre pour les flux HTTP non sécurisés. La journalisation doit être réalisée en accord avec le respect de la vie privée des clients (se reporter au §3.2.2.3) ;
- le proxy a la possibilité de mettre en cache du contenu qu'il peut resservir à plusieurs clients qui souhaitent accéder au même serveur cible.

Cependant, le déchiffrement présente plusieurs inconvénients :

- des données normalement chiffrées sont présentes en clair au niveau du proxy. Si ce dernier est compromis, des informations sensibles peuvent être exposées ;
- l'authentification du client à l'aide d'un certificat n'est plus possible auprès d'un site web qui requerrait ce mode d'authentification. En effet, le proxy étant placé en coupure, le client ne dialogue pas directement en TLS avec le site web ; il ne reçoit donc pas les demandes d'authentification par certificat formulées par ce dernier. Les sites qui requièrent une authentification par certificat doivent donc être placés dans une liste blanche pour laquelle le déchiffrement n'est pas effectué ;
- le niveau de sécurité du tunnel TLS établi sur Internet avec le serveur cible ne dépend plus du navigateur web du client. Celui-ci n'est donc pas en mesure de connaître les risques qu'il prend (validité du certificat serveur, suite cryptographique employée, version de TLS, utilisation de la *PFS*, etc.). La sécurisation des tunnels TLS établis avec le monde extérieur repose uniquement sur les possibilités offertes par le proxy en tant que client, celui-ci étant potentiellement plus laxiste au niveau TLS que les navigateurs web les plus récents ;
- une AC interne doit être employée pour générer les certificats que le proxy présente à ses clients (se reporter au §3.2.1.1).

En résumé, si le déchiffrement des flux HTTPS permet un meilleur contrôle des données échangées entre un système d'information et le monde extérieur, ce processus complexifie l'architecture d'accès à Internet et déporte la sécurisation du canal de communication avec l'extérieur sur le proxy. Ce type d'équipement devient ainsi *très critique*. Sa mise en œuvre doit donc être réalisée en respectant les recommandations mentionnées dans la suite de ce document.

3.2.1.1 Génération des certificats

Les certificats présentés par le proxy pour s'authentifier auprès de ses clients sont particuliers. En effet, ils sont générés spécifiquement pour les besoins de déchiffrement et doivent être valides vis-à-vis des clients pour que l'opération soit transparente. Même si ces certificats respectent les règles présentées ci-dessous ainsi que les recommandations édictées par le RGS, leur existence même est incompatible avec ce référentiel dans la mesure où ils sont générés pour usurper l'identité de sites web appartenant à des tiers.

Plusieurs contraintes doivent être respectées lors de la génération de ces certificats.

R4

> Seule une AC non publique (dont la confiance n'est pas reconnue au delà du système d'information) doit être utilisée pour signer les certificats que le proxy présente à ses clients.

Si cette règle n'est pas respectée, cela peut conduire à la génération de certificats valides publiquement (hors du système d'information) mais qui ne sont pas légitimes vis-à-vis des sites web auxquels ils sont associés. Le vol des clés privées associées à ces certificats pourrait permettre à une personne mal intentionnée de déchiffrer le trafic HTTPS de clients présents sur Internet sans que ceux-ci ne puissent s'en apercevoir (puisque leur navigateur validerait le certificat sans provoquer d'erreur). Afin de détecter ces certificats « illégitimes », les versions récentes des navigateurs (les plus répandus) implémentent des fonctionnalités spécifiques permettant des vérifications avancées. Chrome et Firefox intégrent le *Certificate Pinning*[17]. Internet Explorer dispose de *SmartScreen Filter*[18]. Ces fonctionnalités permettent de vérifier que le certificat présenté par un site web est signé par l'AC publique qu'il a déclarée comme étant celle légitime pour signer son certificat. Si l'AC qui a signé le certificat présenté au client par le site web ne correspond pas à celle dont le navigateur a connaissance pour ce site web, le navigateur peut alerter l'utilisateur, voire lui interdire l'accès au site (possibilité d'attaque active dite de « l'homme du milieu »). L'alerte peut même être remontée automatiquement à la société qui édite le navigateur afin que celle-ci soit informée du mauvais usage d'une AC dont la confiance est reconnue publiquement.

Les fonctionnalités de *Certificate Pinning*/*SmartScreen Filter* n'interdisent cependant pas l'usage d'AC internes (dont la confiance n'est pas reconnue publiquement) pour signer des certificats associés à des sites web publics. Cette pratique est considérée comme un cas d'usage « légitime » de déchiffrement des flux HTTPS : en effet, la configuration des postes clients est obligatoire pour que le déchiffrement puisse s'effectuer sans que les navigateurs web ne lèvent d'alerte de sécurité.

À noter qu'il existe d'autres mécanismes permettant à certains navigateurs de vérifier la légitimité des certificats serveurs, c'est par exemple le cas du projet *Certificate Transparency*[19] initié par Google.

R5

> La chaîne de confiance associée au certificat de l'AC interne qui a signé les certificats présentés par le proxy à ses clients doit être placée dans le magasin des autorités de confiance utilisé par le navigateur des clients. La confiance accordée à cette chaîne par le navigateur doit être limitée à l'authentification des sites web (si l'AC interne n'est utilisée que pour signer des certificats serveurs).

17. Littéralement : « épinglage de certificats ». Cette fonctionnalité est décrite dans une RFC (brouillon) : http://tools.ietf.org/html/draft-ietf-websec-key-pinning.
18. C'est à partir de la version 11 d'Internet Explorer que *SmartScreen Filter* vérifie les informations présentes dans les certificats. Pour plus d'informations : http://blogs.technet.com/b/pki/ (publication du 21 février 2014).
19. http://www.certificate-transparency.org.

La présence de cette chaîne dans ce magasin va permettre au client de vérifier que le certificat que lui présente le proxy est signé par une AC dont la confiance est garantie au préalable. Si le navigateur n'est pas en mesure d'effectuer cette vérification, celui-ci affichera au client une alerte de sécurité lorsqu'il tentera de se connecter au site web cible. Le message indiquera à l'utilisateur que le certificat présenté est signé par une AC inconnue. Si celui-ci choisit malgré tout d'accorder sa confiance à cette autorité et se connecte au site web il prend un risque important (déchiffrement du trafic par un tiers à son insu).

Pour assurer un fonctionnement transparent et sans alerte, la chaîne de confiance associée à l'AC interne doit donc être placée au préalable dans le magasin de certificats par une personne disposant des droits d'administration sur l'application cliente ou le système.

Cas des *EV Certificates*[20] : L'utilisation d'une AC interne pour signer les certificats de sites web publics pose problème au niveau du navigateur du client lorsque celui-ci accède à des sites qui disposent normalement de certificats de type *Extented Validation Certificate* (ou *EV Certificate*). Pour rappel, ce type de certificat, d'un niveau de confiance élevé, peut être délivré par une AC uniquement si elle respecte une procédure précise[21]. Une entité qui fait une demande d'*EV Certificate* à une AC autorisée à en délivrer doit par exemple apporter la preuve de son existence physique et légale. Lorsqu'un navigateur accède à un site qui lui présente un *EV Certificate* sa barre d'adresse se teinte en vert ; celle-ci peut également afficher (en fonction des navigateurs) des informations relatives à l'entité qui détient l'*EV certificate*. Cela permet d'indiquer à l'utilisateur qu'il accède à un site web qui dispose d'un niveau de confiance important ; c'est-à-dire qui présente un certificat qui a été signé par une AC spécifique qui dispose du droit de signer des *EV Certificates*. Les versions récentes des navigateurs sont en mesure d'effectuer cette vérification car ils embarquent directement dans leur code source, de façon statique, les informations relatives aux AC publiques qui ont l'autorisation d'émettre des *EV Certificates*[22]. Lorsqu'un proxy déchiffre les flux HTTPS, il présente à l'utilisateur un certificat signé par une AC interne qui n'est pas reconnue par le navigateur comme étant une autorité en mesure de délivrer des *EV Certificate*. Le certificat présenté est bien valide, le navigateur n'affiche pas d'alerte de sécurité, mais il ne teinte pas sa barre d'adresse en vert. L'utilisateur est ainsi privé d'indications visuelles prévues à l'origine pour renforcer le crédit qu'il accorde à certains sites web.

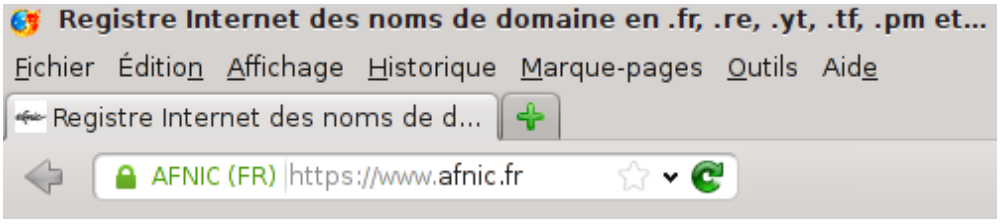

FIGURE 4 – Barre d'adresse du navigateur Firefox lorsqu'un *EV Certificate* lui est présenté.

20. *Extented Validation Certificate* : littéralement « certificat à validation étendue ».
21. Procédure intitulée « EV SSL Certificate Guidelines » publiée par le *CA/Browser Forum* : https://cabforum.org/extended-validation/.
22. Internet Explorer est le seul navigateur qui permet de passer outre la liste des autorités reconnues publiquement aptes à délivrer des *EV Certificate*. Il permet à l'administrateur d'ajouter des AC internes comme étant légitimes pour émettre des *EV Certificate*. Pour plus d'informations : http://technet.microsoft.com/en-us/library/dd759060.aspx.

3.2.1.2 Impact sur les performances

Les opérations cryptographiques réalisées par le proxy pour déchiffrer le trafic HTTPS sont coûteuses en ressources. L'équipement qui porte la fonction proxy doit donc être dimensionné correctement pour pouvoir supporter les tunnels TLS de l'ensemble des clients pour lesquels le trafic HTTPS est déchiffré (en partie ou en totalité).

3.2.2 Bonnes pratiques

3.2.2.1 Génération des certificats

Voici quelques recommandations concernant la génération des certificats présentés par le proxy à ses clients :

R6

> Une AC intermédiaire interne (sous-AC) doit être dédiée à la signature des certificats présentés par le proxy à ses clients.

R7

> Si la solution de proxy intégre nativement une AC (pré-configurée en amont par l'éditeur ou initialisée à l'installation), celle-ci ne doit pas être utilisée pour signer de certificats. L'usage de ce type d'AC présente des risques : autorité identique sur différents équipements, utilisation de gabarits non appropriés, etc.

R8

> Les clés privées associées aux certificats présentés par le proxy à ses clients doivent être protégées par des mécanismes adaptés (se reporter au §2.4).

R9

> Les certificats présentés par le proxy à ses clients doivent être générés en utilisant des gabarits qui respectent les recommandations mentionnées dans les annexes du RGS (se reporter au §2.3).

3.2.2.2 Renforcement de la sécurité de TLS

3.2.2.2.1 Sécurité TLS entre les clients et le proxy

Bien que les réseaux qui séparent les clients du proxy sont généralement considérés comme étant de confiance, le trafic HTTPS interne doit être correctement sécurisé afin d'éviter toute exposition inutile des données à protéger. Cela est d'autant plus aisé à mettre en œuvre que les deux entités (les clients et le proxy) sont maîtrisées.

Voici quelques recommandations visant à renforcer la sécurité TLS entre les clients et le proxy.

R10

> Le proxy ne doit permettre l'établissement de tunnels TLS qu'en utilisant les versions de TLS les plus récentes supportées par les navigateurs web des clients.

L'usage de TLS v1.1 (et versions supérieures) permet d'éviter d'exposer les tunnels HTTPS à des attaques récentes (*BEAST*[23] par exemple). L'ensemble des versions de SSL (v2.0 et v3.0) doit être désactivé au niveau du serveur TLS du proxy car les navigateurs récents supportent a minima TLS v1.0.

R11

> Le proxy ne doit offrir à ses clients que des suites cryptographiques robustes (se reporter à l'annexe B de ce document).

Il est nécessaire de vérifier la compatibilité des suites retenues avec celles que supporte les navigateurs web utilisés par les clients. La suite sélectionnée par le proxy pour établir le tunnel TLS avec le client doit être la plus robuste parmi celles proposées par son navigateur (cette suite n'est pas nécessairement celle qui a la préférence du client).

R12

> Les suites cryptographiques les plus robustes offertes par le proxy doivent permettre la *PFS*.

Le renforcement du niveau de sécurité de TLS à l'aide de la propriété *PFS* permet de se prémunir contre des attaques internes dont le but serait de capturer du trafic afin de le déchiffrer ultérieurement.

R13

> La compression TLS doit être désactivée au niveau du serveur TLS du proxy.

L'usage de la compression TLS rend vulnérable le flux HTTPS à l'attaque *CRIME*[24].

R14

> Si le proxy supporte la reprise des sessions TLS, il est nécessaire de vérifier quels mécanismes il implémente et comment ces derniers fonctionnent.

23. *Browser Exploit Against SSL/TLS* : Cette attaque publiée en 2011 concerne les versions de TLS inférieures à 1.1. Elle est référencée sous la CVE 2011-3389 (http://www.cvedetails.com/cve/CVE-2011-3389).

24. *Compression Ratio Info-leak Made Easy* : Cette attaque a été publiée en 2012, elle est référencée sous les CVE 2012-4929 (http://www.cvedetails.com/cve/CVE-2012-4929) et 2012-4930 (http://www.cvedetails.com/cve/CVE-2012-4930).

La reprise de sessions TLS peut s'effectuer en utilisant des identifiants de session (*session ID*[25]) ou des tickets de session (*session ticket*[26]). Dans les deux cas, les informations sensibles relatives à l'état des sessions en cours sont manipulées par le serveur TLS. Si ces données venaient à être compromises, les sessions TLS pourraient être déchiffrées (même si la *PFS* est activée). Il est donc nécessaire de s'assurer que ces informations ne sont pas conservées abusivement par le serveur TLS. Idéalement ces données ne doivent être stockées qu'en mémoire et durant un laps de temps défini, de préférence configurable.

3.2.2.2.2 Sécurité TLS entre le proxy et Internet

Les réseaux qui séparent le proxy du serveur web cible ne sont pas maîtrisés. Le trafic HTTPS doit donc être sécurisé au maximum afin d'éviter toute compromission des données à protéger. Cela est d'autant plus difficile à mettre en œuvre que l'une des deux entités (le serveur web) n'est pas maîtrisée.

Voici quelques recommandations visant à renforcer la sécurité TLS entre le proxy et le serveur web cible :

R15

> Le comportement du proxy en tant que client TLS doit être vérifié afin de s'assurer que celui-ci n'introduise pas de faiblesses lors de l'établissement des tunnels TLS. Il est par exemple nécessaire de valider le fonctionnement du proxy lorsqu'un serveur lui présente un certificat qui n'est plus valide.

Certaines solutions de proxy ne vérifient pas correctement les caractéristiques des certificats présentés par les serveurs web (durée de validité, chaîne de certification, etc.), ce qui expose par transitivité les clients[27]. Le fonctionnement des mécanismes de révocation supportés par le proxy doivent être testés afin de déterminer le comportement exact du proxy lorsque celui-ci reçoit un certificat révoqué.

R16

> Le client TLS du proxy doit supporter TLS v1.1 et v1.2 afin de disposer du meilleur niveau de sécurité lorsque les sites web visités supportent ces versions récentes du protocole.

R17

> Les suites cryptographiques supportées par le proxy doivent être proposées au serveur web selon un ordre pré-établi. Les suites les plus robustes (incluant la *PFS*) doivent être préférées.

R18

> Si le client du proxy supporte la renégociation TLS, celle-ci doit être sécurisée[28].

Le support de la renégociation sécurisée permet d'éviter de rendre vulnérable le canal de communication à certaines attaques actives dites de « l'homme du milieu »[29].

25. RFC 5246.
26. RFC 5077.
27. Pour plus de détails : http://www.secureworks.com/cyber-threat-intelligence/threats/transitive-trust/.
28. RFC 5746.
29. Vulnérabilité référencée sous la CVE 2009-3555 (http://www.cvedetails.com/cve/CVE-2009-3555).

R19

> Sauf à ce que ce soit expressément validé, lorsqu'un serveur web sélectionne une suite cryptographique qui n'est pas assez robuste, le proxy doit refuser l'établissement du tunnel TLS. Le proxy peut informer le client à l'origine de la demande et lui indiquer que le site qu'il cherche à joindre ne fournit pas un niveau de sécurité suffisant au regard des exigences qui lui sont fixées.

Si un serveur web retient une suite cryptographique trop faible, cela signifie qu'elle fait partie de celles proposées par le proxy. Ce dernier peut être amené à supporter des suites cryptographiques qui ne sont pas robustes (au sens RGS du terme) pour des questions de compatibilité. Cependant, le proxy peut être configuré pour interdire l'usage de ces suites lorsque le site web demandé par le client appartient à une catégorie qui exige un niveau de sécurité élevé (compatible avec les exigences du RGS par exemple).

R20

> La compression TLS doit être désactivée au niveau du client TLS du proxy.

L'usage de la compression TLS rend vulnérable le flux HTTPS à l'attaque *CRIME*.

R21

> La liste des AC publiques de confiance intégrées à la solution de proxy doit être vérifiée à l'installation et doit être revue de façon régulière.

Les AC publiques intégrées à la solution de proxy ont été jugées de confiance par l'éditeur. Il a donc choisi d'intégrer les chaînes de confiance associées dans le magasin de ses proxys. Cette liste n'est pas nécessairement la même que celle présente dans le magasin des navigateurs web des clients du proxy. Il est possible que l'équipement accorde sa confiance à des autorités qui ont été retirées du magasin des versions récentes des navigateurs, par exemple suite à des incidents de sécurité rendus publics. C'est la raison pour laquelle le magasin du proxy doit être vérifié et maintenu à jour par les personnes en charge de son exploitation. Il est possible que ce magasin soit modifié lors des mises à jour logicielles du proxy (les notes de version doivent être consultées pour le vérifier) ou par l'intermédiaire d'un mécanisme automatique spécifique à la solution de proxy employée.

3.2.2.3 Protection de la vie privée

Voici quelques recommandations visant à limiter au maximum le traitement de données à caractère personnel au niveau d'un proxy :

R22

> Ne pas procéder au déchiffrement de certains types de sites web identifiés comme étant destinés à un usage strictement personnel (certains sites bancaires par exemple).

La décision de ne pas déchiffrer le trafic HTTPS échangé avec certains sites web doit être prise par le proxy avant l'établissement complet du tunnel TLS. Plusieurs possibilités existent mais elles dépendent de la solution de proxy employée. Elles sont détaillées dans le paragraphe 3.3. Le déchiffrement sélectif présente également l'avantage de diminuer la consommation de ressources au niveau du proxy.

R23

La journalisation des flux déchiffrés doit être équivalente à celle configurée pour les flux HTTP standards traités par le proxy. Elle ne doit pas permettre d'enregistrer davantage d'informations.

R24

Si le proxy transmet à d'autres équipements le contenu déchiffré des flux HTTPS, les liens sur lesquels transitent ces informations doivent être protégés logiquement et physiquement pour éviter tout accès illégitime aux données.

Le proxy peut par exemple transmettre le contenu déchiffré à d'autres équipements à l'aide du protocole ICAP [30] qui n'est pas nécessairement sécurisé. Des proxys peuvent également s'échanger des données issues du traffic déchiffré lorsqu'ils sont déployés en grappe.

3.3 Traitement des flux HTTPS sans déchiffrement

Ce paragraphe détaille le cas où le proxy web ne déchiffre pas le trafic HTTPS. Dans cette configuration les possibilités d'action du proxy sont beaucoup plus limitées.

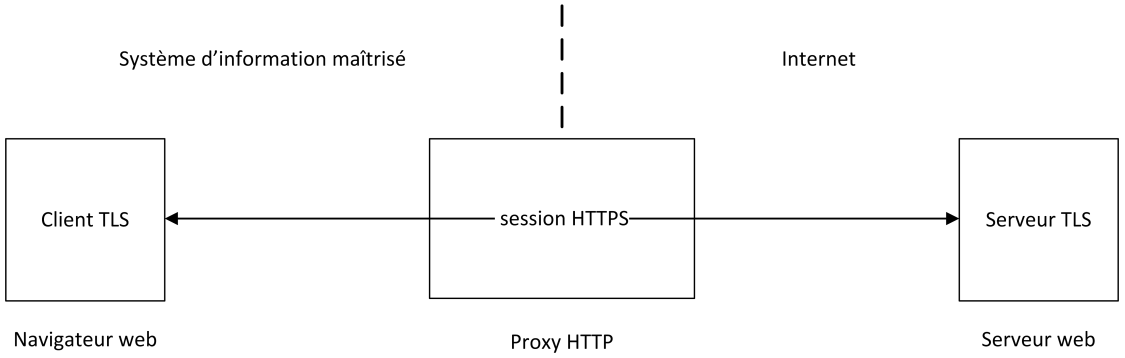

FIGURE 5 – Traitement des flux HTTPS sans déchiffrement

Sans procéder au déchiffrement du trafic HTTPS, il n'est pas possible d'inspecter le contenu des échanges HTTP. Cependant, il peut être envisageable de réaliser un filtrage élémentaire en procédant à l'analyse du contenu de certains flux transmis en clair avant l'établissement du tunnel TLS.

Il est par exemple possible d'analyser le contenu de la demande de connexion HTTPS initiale pour obtenir des informations concernant le site web demandé. Lorsque les clients accèdent à un proxy HTTP configuré en mode explicite, la première requête HTTPS utilise la méthode HTTP *CONNECT* et contient en argument le FQDN [31] associé à la première URL demandée par le client.

L'analyse de la séquence de négociation TLS, qui débute une fois le client connecté au serveur, permet également d'obtenir des informations concernant le site web demandé quel que soit le mode dans lequel le proxy est configuré.
Voici les éléments qui circulent en clair et qu'il est possible d'analyser pour obtenir le domaine ou le FQDN du site :

30. *Internet Content Adaptation Protocol.*
31. *Full Qualified Domain Name* ou nom d'hôte pleinnement qualifié (ex : www.ssi.gouv.fr).

– le champ *Common Name*, contenu dans le certificat présenté par le serveur, peut contenir un FQDN (voire un domaine) pour lequel le certificat est valide ;
– le champ *Subject Alternative Name*, contenu dans le certificat présenté par le serveur, peut contenir plusieurs FQDN (voire des domaines) pour lesquels le certificat est valide ;
– le champ *Server Name* de l'extension TLS *Server Name Indication (SNI)* contient le FQDN du site web lorsque cette extension est utilisée par le navigateur du client. Cette extension est employée par certains navigateurs pour formuler explicitement, au moment de l'initiation du tunnel TLS, le FQDN du site web auquel le client souhaite accéder. Si plusieurs sites web sont accessibles par une même adresse IP, le serveur web peut, grâce à ce mécanisme, être en mesure de présenter au client le certificat correspondant au site qu'il cherche à joindre.

L'inspection du contenu de la séquence de négociation TLS permet également de vérifier certains paramètres qui déterminent le niveau de sécurité du canal de communication (version de TLS, suites cryptographiques, certificats, etc.). Si certaines exigences ne sont pas satisfaites, le proxy peut choisir d'empêcher l'établissement du tunnel TLS.

4 Traitement des flux HTTPS entrants

Cette section a pour objectif de présenter les possibilités de traitement des flux HTTPS qui sont initiés à partir de clients externes (Internet par exemple) et destinées à des serveurs web hébergés au sein d'un système d'information maîtrisé .

4.1 Architecture

L'usage de serveurs mandataires inverses HTTP (ou reverse proxy HTTP) est une bonne pratique d'architecture permettant de s'assurer que l'ensemble des clients passent par un point de contrôle unique pour pouvoir accéder à des sites web hébergés sur des réseaux maîtrisés. C'est donc au niveau du reverse proxy que sont généralement analysés les flux HTTPS entrants, si celui-ci dispose des fonctionnalités permettant l'analyse de ce type de flux.

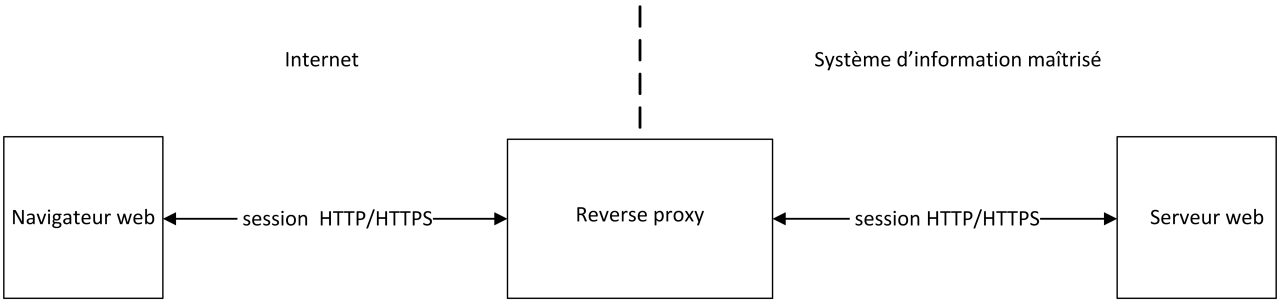

FIGURE 6 – Reverse proxy HTTP

Remarque : Il n'est pas recommandé de procéder à l'analyse des flux HTTPS entrants à l'aide d'équipements qui ne se positionnent pas en coupure des sessions TLS (ex : sonde « passive »). En effet, dans ce type d'architecture, l'équipement en charge de l'analyse ne peut procéder au déchiffrement des flux HTTPS que si les deux conditions suivantes sont remplies :

1. l'équipement dispose d'une copie des clés privées des sites web ;
2. les suites cryptographiques proposées par les serveurs web ne permettent pas la PFS. En effet, l'équipement en charge du déchiffrement doit être en mesure de déterminer les clés de session en utilisant uniquement les clés privées des serveurs web, ce qui n'est pas possible avec la PFS. Cette contrainte dégrade ainsi fortement le niveau de sécurité des tunnels TLS établis entre les clients et les serveurs.

Ces architectures spécifiques ne seront pas traitées dans la suite de ce document, nous considérerons que le traitement des flux HTTPS est réalisé par un équipement positionné en coupure des sessions TLS et qui assure la fonction de reverse proxy.

4.2 Traitement des flux HTTPS par déchiffrement

Ce paragraphe détaille le cas où le reverse proxy est le point de terminaison des sessions HTTPS initiées par les clients qui souhaitent accéder au site web hébergé en interne. Dans cette configuration le reverse proxy peut ensuite se connecter au serveur web cible en HTTP ou en HTTPS, cela va dépendre des choix réalisés au niveau de l'architecture interne (se référer au §4.2.2.3).

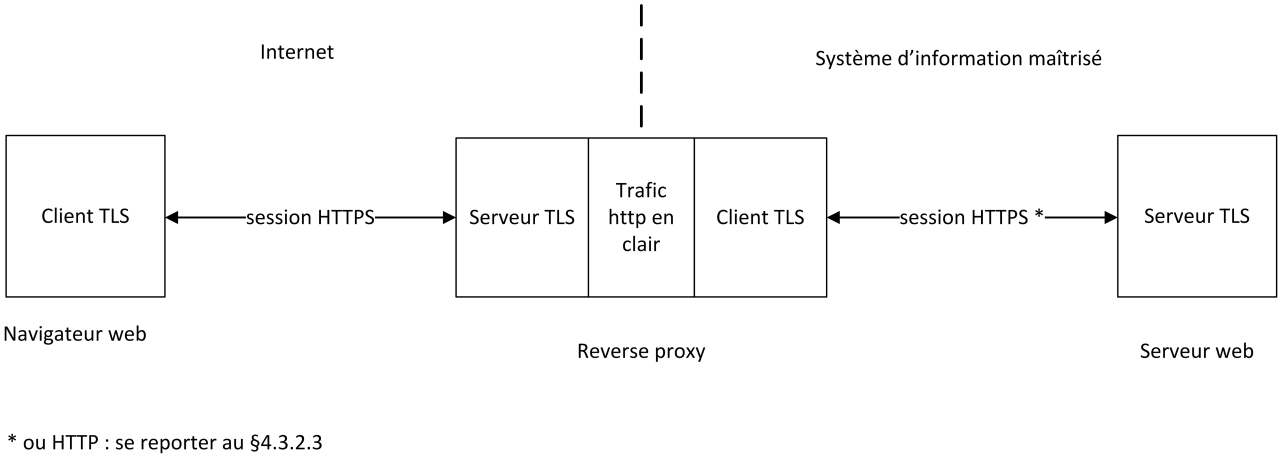

FIGURE 7 – Traitement des flux HTTPS entrants par déchiffrement

4.2.1 Les enjeux du déchiffrement

Avant de mettre en place des mécanismes de déchiffrement au niveau d'un reverse proxy web, il est nécessaire de bien comprendre les avantages, les inconvénients et les problématiques que cela induit.

Le fait de terminer les tunnels TLS au niveau du reverse proxy procure plusieurs avantages :
- il est possible d'analyser le contenu HTTP afin de protéger les serveurs web internes contre des menaces émanant des clients : attaque au niveau applicatif, envoi de fichiers malveillants, etc. ;
- il est possible d'agir sur le contenu HTTP délivré : mise en cache, réécriture, etc. ;
- il est possible de configurer de façon homogène et centralisée la politique de journalisation des accès aux sites web, au même titre que pour les flux HTTP non sécurisés ;
- la centralisation des configurations TLS de l'ensemble des sites web accessibles publiquement permet d'assurer une cohérence et une homogénéité au niveau du paramétrage TLS de chacun d'entre eux ;
- la protection des clés privées associées aux certificats publics peut être homogène et peut être plus facilement renforcée (usage de *HSM* par exemple). Celle-ci ne dépend plus des mécanismes de protection mis en œuvre par les serveurs qui hébergent les sites web internes ;
- il est possible de décharger les serveurs web internes. Lorsque ces derniers sont accédés à l'aide du protocole HTTP par le reverse proxy (se référer au §4.2.2.3), ils n'ont pas à exécuter les opérations cryptographiques nécessaires à l'établissement et au maintien de tunnels TLS.

Cependant, le déchiffrement présente quelques inconvénients :
- des données normalement chiffrées jusqu'au serveur web sont présentes en clair au niveau du reverse proxy ;
- la concentration de l'ensemble des bi-clés au niveau du reverse proxy accroît la criticité de ce composant ;
- si une authentification du client à l'aide d'un certificat doit être mise en place, c'est le reverse proxy qui porte cette fonction. Une fois le client correctement authentifié, le reverse proxy doit

ensuite employer des mécanismes sécurisés pour propager l'identité du client jusqu'au serveur web cible (se reporter au §4.2.2.4).

En résumé, si le déchiffrement des flux HTTPS au niveau d'un reverse proxy présente quelques inconvénients, ce choix d'architecture est particulierement pertinent pour exercer un contrôle des données échangées avec l'extérieur. Ce type d'architecture permet également d'assurer l'homogénéité d'un ensemble d'éléments de configuration contribuant à renforcer la sécurité des tunnels TLS établis avec des clients non maîtrisés.

4.2.2 Bonnes pratiques

4.2.2.1 Génération des certificats

Les certificats présentés aux clients par le reverse proxy doivent être générés à partir d'une AC dont la confiance est reconnue publiquement (dont la chaîne de confiance associée est présente dans le magasin des navigateurs). Cette condition est nécessaire pour qu'aucun message d'erreur ne soit présenté aux clients lorsqu'ils cherchent à accéder au serveur web hébergé au sein du système d'information interne. Le choix du PSCE [32] en charge de fournir les certificats publics est un élément structurant qui contribue à améliorer le niveau de sécurité d'un service HTTPS accessible depuis Internet.

Les PSCE qualifiés par l'ANSSI sont référencés sur le site web du LSTI [33]. Certaines recommandations mentionnées dans le guide ANSSI relatif à l'externalisation [34] peuvent également aider à la sélection d'un PSCE.

Les critères suivants peuvent également permettre de sélectionner un PSCE :

R25

> Utiliser une AC opérée sur le territoire national.

R26

> Utiliser une AC qui propose des certificats compatibles avec les exigences formulées par le RGS.

R27

> Utiliser une AC reconnue comme étant de confiance par une large majorité des navigateurs web du marché.

R28

> Choisir un prestataire qui est en mesure d'apporter des éléments prouvant son sérieux : accès sécurisé au service client, engagements du support technique, certification [35], délivrance d'*EV Certificate*, etc.

32. Prestataire de Service de Certification Électronique.
33. LSTI : Laboratoire en Sciences et Technologies de l'Information. La liste des PSCE qualifiés est détaillée à l'adresse http ://www.lsti-certification.fr/images/liste_entreprise/Liste PSCe.pdf.
34. `http://www.ssi.gouv.fr/IMG/pdf/2010-12-03_Guide_externalisation.pdf`.
35. Certification *WebTrust For CA* par exemple : `http://www.webtrust.org`.

4.2.2.2 Sécurité TLS entre le reverse proxy et Internet (les clients)

Les réseaux qui séparent les clients et le reverse proxy ne sont pas maîtrisés. Le trafic HTTPS doit donc être sécurisé au maximum afin d'éviter toute compromission des données à protéger. Cela est d'autant plus difficile à mettre en œuvre que l'une des deux entités (le client) n'est pas maîtrisée.

Les recommandations suivantes visent à renforcer la sécurité TLS entre le reverse proxy et Internet :

R29

> Le reverse proxy ne doit permettre l'établissement de tunnels TLS qu'en utilisant les versions récentes de ce protocole.

L'usage de TLS v1.1 (et versions supérieures) permet d'éviter d'exposer les tunnels HTTPS à des attaques récentes (*BEAST* par exemple). L'ensemble des versions de SSL (v2.0 et v3.0) doit être désactivé. Par contre, pour des questions de compatibilité, il est encore nécessaire de supporter TLS v1.0. Néanmoins, si cette version est activée, il est nécessaire de vérifier que la solution de reverse proxy employée implémente correctement les contre mesures corrigeant certaines vulnérabilités propres à cette version du protocole.

R30

> Le reverse proxy ne doit proposer que des suites cryptographiques dont le niveau de sécurité est acceptable (se reporter à l'annexe B de ce document). Il doit sélectionner en priorité les suites les plus robustes (incluant la *PFS*) parmi celles proposées par le client.

R31

> Seule une renégociation sécurisée peut être autorisée par le reverse proxy.

R32

> Si le reverse proxy supporte la reprise de sessions TLS, il est nécessaire de déterminer quels mécanismes il implémente et comment ces derniers fonctionnent.

La reprise de sessions TLS peut s'effectuer en utilisant des identifiants de session (*session ID*) ou des tickets de session (*session ticket*). Dans les deux cas, les informations sensibles relatives à l'état des sessions en cours sont manipulées par le serveur TLS. Si ces données venaient à être compromises, les sessions TLS pourraient être déchiffrées (même si la *PFS* est activée). Il est donc nécessaire de s'assurer que ces informations ne sont pas conservées abusivement par le reverse proxy. Idéalement ces données ne doivent être stockées qu'en mémoire et durant un laps de temps défini, de préférence configurable.

R33

> La compression TLS doit être désactivée au niveau du reverse proxy.

L'usage de la compression TLS rend vulnérable le flux HTTPS à l'attaque *CRIME*.

4.2.2.3 Sécurité du trafic interne (entre le reverse proxy et le serveur web)

Deux choix sont possibles concernant le trafic web qui circule entre le reverse proxy et le serveur web interne :

1. le protocole HTTPS peut être utilisé par le reverse proxy pour accéder au serveur web. Dans ce cas, les données qui circulent sur les réseaux internes sont protégées. Dans cette configuration, le serveur web doit disposer de bi-clés (couples certificat/clé privée) qui lui sont propres. Les certificats présents sur le serveur web doivent être signés par une AC non publique. Seul le reverse proxy aura connaissance de ces certificats. Il ne sera jamais visible par les clients (qui n'ont connaissance que des certificats présentés par le reverse proxy). Le reverse proxy doit également disposer dans son magasin de la chaîne de confiance associée à l'AC interne pour valider le certificat présenté par le serveur Web ;
2. le protocole HTTP peut être utilisé par le reverse proxy pour accéder au serveur web. Dans ce cas, les données qui circulent sur les réseaux internes ne sont pas protégées. Dans cette configuration, le serveur cible n'héberge pas de bi-clé.

Le choix de l'architecture dépend :

- du niveau de confidentialité du contenu : il est nécessaire de déterminer si l'architecture réseau interne permet de garantir une protection suffisante pour autoriser la circulation en clair de données qui sont protégées lorsqu'elles circulent sur Internet ;
- du besoin en intégrité : il est nécessaire de déterminer si la contrainte en intégrité est forte sur les données. Par exemple, lorsque le reverse proxy transmet au serveur web cible des informations relatives aux utilisateurs qu'il a préalablement authentifiés (se reporter au §4.2.2.4) ;
- du dimensionnement des serveurs web : il est nécessaire de déterminer si les serveurs qui hébergent les sites web internes disposent des ressources matérielles suffisantes pour pouvoir supporter les opérations cryptographiques nécessaires à l'établissement et au maintien de sessions HTTPS ;
- du coût d'exploitation : l'emploi du procotole HTTPS au niveau des serveurs web implique la gestion de certificats internes en plus de ceux hébergés par le reverse proxy.

4.2.2.4 Propagation de l'identité des clients

Les architectures nécessitant le déport de la première authentification des clients au niveau du reverse proxy (usage de certificats clients par exemple) pose la problématique de la propagation des informations d'identification jusqu'au site web cible. En effet, le reverse proxy doit être en mesure de communiquer au site les informations relatives aux utilisateurs qu'il a correctement authentifiés. Une solution simple (parmi d'autres) consiste à configurer le reverse proxy pour qu'il inscrive dans les en-têtes HTTP les informations relatives à l'identité des utilisateurs authentifiés. Le contenu de ces en-têtes est ensuite pris en compte par le site web cible qui peut ainsi différencier ses traitements en fonction des informations d'identification qu'il reçoit.

L'ajout d'informations dans les en-têtes HTTP par un reverse proxy n'est pas un mécanisme utilisé exclusivement pour propager des informations d'identification, il peut par exemple être employé pour transmettre au serveur web cible l'adresse IP originelle du client (en-tête *X-Forwarded-For*).

Quel que soit le cas d'usage, l'ajout d'informations dans les en-têtes HTTP peut présenter un risque si les clients sont en mesure de forger ces en-têtes en amont dans le but de tromper le serveur cible. Pour éviter cela, le reverse proxy doit vérifier au préalable les en-têtes des requêtes HTTP qu'il reçoit des clients et doit procéder à leur nettoyage si nécessaire.

Attention : L'acceptation du client ne doit pas se baser uniquement sur la validité des informations d'authentification qu'il présente. Un client peut s'authentifier correctement sans pour autant être autorisé à accéder à la ressource qu'il demande. En conséquence, la solution mise en oeuvre doit également permettre de vérifier que l'utilisateur dispose des droits d'accès au site web demandé, charge au reverse proxy ou au seveur web de réaliser cette vérification.

4.3 Traitement des flux HTTPS sans déchiffrement

Ce paragraphe présente le cas où le reverse proxy n'est pas le point de terminaison des sessions HTTPS initiées par les clients qui souhaient accéder au site web hébergé sur le système d'information interne. Dans cette configuration, le reverse proxy ne fait que transférer les données chiffrées entre le client et le serveur web cible. Il n'héberge alors aucun bi-clé.

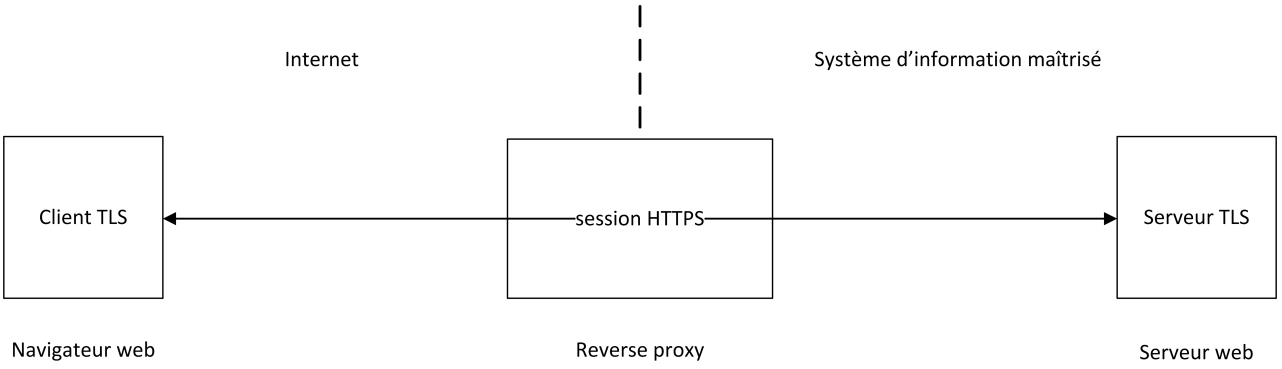

FIGURE 8 – Traitement des flux HTTPS entrants sans déchiffrement

La mise en œuvre de ce type d'architecture ne présente pas de réel intérêt dans la mesure où elle ne permet pas au reverse proxy d'accéder au contenu HTTP. Ce dernier peut simplement s'assurer que les serveurs web internes établissent les tunnels TLS en respectant les exigences de sécurité auxquelles ils sont normalement assujettis (suites cryptographiques, version de TLS, etc.). Ce type de configuration oblige également les serveurs web à héberger leurs clés privées accompagnées de leurs certificats signés par une AC dont la confiance est reconnue publiquement.

4.4 Sécurité web complémentaire

L'usage de TLS est fondamental pour assurer la sécurisation de flux HTTP mais l'emploi de ce protocole n'est pas suffisant. D'autres mesures complémentaires doivent être mises en œuvre au niveau des serveurs web pour ne pas exposer les informations à protéger.

Voici une liste non exhaustive de bonnes pratiques de sécurisation qui viennent en complément de l'usage du protocole HTTPS. Ces recommandations sont applicables directement au niveau de la configuration des sites web hébergés.

R34

Ne pas activer le protocole de compression HTTP nommé *SPDY*[36] lorsqu'un site web est accessible à l'aide du protocole HTTPS.

Cette mesure est nécessaire afin de se prémunir contre l'attaque *CRIME* mentionnée précédemment. Cette dernière fonctionne non seulement lorsque la compression TLS est activée mais également

36. SPDY (se prononce « speedy ») : Protocole de compression permettant l'accélération du chargement du contenu HTTP.

lors de l'emploi combiné du protocole HTTPS et de *SPDY*.

R35

> Ne pas inclure de liens HTTP dans une page web accessible à l'aide du protocole HTTPS (problème dit de *mixed content*[37]) .

Le fait d'inclure du contenu non sécurisé dans une page accessible à l'aide du protocole HTTPS rend vulnérable le site web à des attaques actives dites de « l'homme du milieu » réalisées à l'aide de scripts malveillants. Certains navigateurs alertent l'utilisateur lorsque des contenus de différentes natures sont présents dans une page web.

R36

> Renforcer la sécurité des cookies à l'aide des attributs *Secure* et *HttpOnly*.

L'attribut *Secure* indique que le cookie ne peut être transmis par le serveur que par l'intermédiaire du protocole HTTPS. Cela évite son envoi sans protection. L'attribut *HttpOnly* interdit l'accès au cookie par des scripts exécutés par le navigateur du client. Cela réduit le risque d'attaques de type *XSS*[38].

R37

> Utiliser les en-têtes HTTP *HSTS*[39] et *CSP*[40] pour renforcer la sécurité de l'accès et du contenu des sites web.

L'en-tête *HSTS* indique au navigateur du client (s'il est compatible) que le site web doit être accédé uniquement à l'aide du protocole HTTPS. L'en-tête *CSP* permet de restreindre l'origine des contenus (image, script, etc.) inclus dans une page. Cela permet de se prémunir contre les attaques de type *XSS* en restreignant les domaines de provenance des contenus tiers présents dans une page web.

La note technique ANSSI intitulée « Recommandations pour la sécurisation des sites web[41] » détaille plus largement les mesures de sécurisation d'un site web, qu'il soit accessible à l'aide du protocole HTTPS ou non.

37. https://developer.mozilla.org/en-US/docs/Security/MixedContent.
38. *Cross-site scripting* ou « injection de code indirecte ».
39. *HTTP Strict Transport Security*.
40. *Content Security Policy*.
41. http://www.ssi.gouv.fr/IMG/pdf/NP_Securite_Web_NoteTech.pdf.

5 Validation des configurations

La configuration TLS d'un proxy ou d'un reverse proxy doit être validée afin de s'assurer que celle-ci est correctement sécurisée. Elle peut être testée à l'aide d'outils, voire de services, spécifiques disponibles sur Internet.

Voici quelques exemples d'outils[42] permettant de tester le paramétrage d'un serveur TLS :

- client *OpenSSL* : *OpenSSL*[43] est une implémentation de SSL/TLS en source ouverte. Cette boîte à outils inclut en particulier un client SSL/TLS en ligne de commande : *openssl s_client*. Cet outil offre des fonctionnalités étendues permettant d'interroger un serveur TLS et d'investiguer en détail d'éventuels problèmes de configuration. TLS v1.2 n'est supporté qu'à partir de la version 1.0.1 d'*OpenSSL* ;
- *SSLScan* : cet outil repose sur *OpenSSL*. Il permet de tester un service SSL/TLS et offre en particulier la possibilité de lister les suites cryptographiques supportées par le serveur. À noter que TLS v1.1 et v1.2 ne sont supportés qu'à partir de la version 1.10.0[44] de *SSLScan* ;
- *SSLyze* : cet outil est également un scanner SSL/TLS qui repose aussi sur *OpenSSL*. TLS v1.2 n'est supporté qu'à partir de la version 0.4 de *SSLyze*[45].

Pour les services en ligne, le site https ://www.ssllabs.com (qui appartient à l'éditeur de solutions de sécurité Qualys) met à disposition gratuitement de nombreuses ressources concernant TLS (bonnes pratiques, tableaux de bord, etc.). Il fournit en particulier des services permettant d'évaluer en ligne la configuration TLS d'un client ou d'un serveur.

42. Les outils et services sont donnés à titre indicatif sans garantie de l'ANSSI sur la confiance à leur accorder.
43. http://www.openssl.org.
44. https://github.com/dinotools/sslscan.
45. https://github.com/isecpartners/sslyze.

Annexes

A Aspects juridiques

Cette annexe présente les aspects juridiques liés au déchiffrement de flux dans un but informatif et de manière non-exhaustive. En particulier, elle se limite aux problématiques liées aux flux entrants et sortants depuis les postes de travail de salariés qu'ils soient fixes ou nomades.

La question du déchiffrement d'un contenu implique nécessairement celle du chiffrement de celui-ci en amont. Or les deux opérations comportent des risques juridiques qui peuvent apparaître antinomiques. En effet, un contenu chiffré peut être source de comportement délictueux donnant lieu à la mise en jeu de la responsabilité de l'entité. Le déchiffrement, qui rend visible un contenu destiné à être confidentiel, permet de se prémunir contre ce risque mais parfois au prix des risques censés être couverts par le chiffrement.

Le chiffrement étant relié à une technologie à laquelle peuvent correspondre plusieurs usages, la présente note se limite au fait qu'il répond au seul besoin de garantir la confidentialité des échanges de données électroniques, à l'exclusion du chiffrement à usage de signature ou d'authentification.

En tout état de cause, le lecteur qui souhaite obtenir des informations plus détaillées est invité à se faire assister par un conseil.

De façon synthétique, si la mise en place d'outils de chiffrement est autorisée et soumise à un régime juridique particulier, elle n'en est pas moins susceptible d'entraîner un risque non négligeable de mise en cause de la responsabilité de l'entité qui en bénéficie, et ce sur plusieurs fondements. Pour se préserver, l'employeur peut effectivement être amené à mettre en place un système de déchiffrement et de contrôle des échanges d'information réalisés par ses salariés sur les systèmes d'information qu'il met à leur disposition qui, s'il n'est pas strictement encadré, peut conduire l'employeur à engager sa responsabilité sur d'autres fondements.

Le régime juridique du chiffrement

L'utilisation des moyens de cryptologie est libre en France [46]. En revanche, la fourniture, le transfert depuis ou vers un État membre de la Communauté européenne ainsi que l'importation et l'exportation de ces moyens sont réglementés lorsqu'ils n'assurent pas exclusivement des fonctions d'authentification ou de contrôle d'intégrité. En sont donc, par nature exclus, les outils de contrôle d'accès et les passeports. Ces opérations doivent alors faire l'objet d'une déclaration préalable ou d'une autorisation de l'ANSSI [47].

La mise en place d'outils de chiffrement est soumise à des dispositions légales relatives à la cryptologie dont la violation peut être sanctionnée sur le plan administratif, civil et pénal [48].

46. Loi n° 2004-575 du 21 juin 2004 pour la confiance dans l'économie numérique (LCEN), art. 30 ; décret n° 2007-663 du 2 mai 2007 pris pour l'application des articles 30,31 et 36 de la loi n° 2004-575 du 21 juin 2004 pour la confiance dans l'économie numérique et relatif aux moyens et aux prestations de cryptologie.

47. Informations complémentaires disponibles sur : http ://www.ssi.gouv.fr/fr/reglementation-ssi/cryptologie/ . Contact : controle [at] ssi.gouv.fr.

48. En cas de non-respect de ces dispositions, des sanctions administratives telles l'interdiction de mise en circulation du moyen de cryptologie (art. 34 LCEN), des sanctions pénale assorties, le cas échéant, de peines complémentaires telles

Les risques juridiques liés au chiffrement

L'utilisation de moyens de chiffrement peut être source de responsabilité de l'employeur, notamment, lorsque le chiffrement a permis ou facilité la commission d'infractions ou a conduit au non-respect des obligations de sécurité à sa charge. Les fondements sont multiples, à savoir :

- selon l'article 1384 alinéa 5 du Code civil, l'employeur est responsable de l'agissement de ses salariés [49] notamment sur les réseaux informatiques [50]. Un employeur peut être tenu responsable si l'employé commet des infractions sur son lieu de travail et avec les outils professionnels mis à sa disposition sauf si l'employeur démontre que le salarié a agi sans autorisation et en dehors de ces attributions [51] ;
- l'employeur peut être soumis à certaines obligations précisées à l'article 6-I-7° de la LCEN (notamment la conservation des éléments de journalisation relatifs aux échanges) s'il est considéré comme un fournisseur d'accès à Internet [52] au sens de l'article 6-I de la LCEN [53] ;
- en tant que responsable du traitement des données à caractère personnel qui est soumis aux obligations de sécurité et de notification de la violation des données à caractère personnel imposées par la loi « Informatique et Libertés » [54] ;
- le titulaire d'un accès à des services de communications au public en ligne est responsable de la sécurisation de cet accès selon les articles L. 336-3 et R. 335-5 du code de propriété intellectuelle.

L'utilisation de moyens de chiffrement n'entraîne pas, par elle-même, la responsabilité de l'employeur dès lors qu'elle est libre.
En revanche, sa responsabilité peut être engagée soit :
- **en raison d'agissements délictueux qui pourraient être commis par les salariés grâce aux moyens qu'il leur a fournis (matériels, accès internet, etc.) et dissimulés grâce au chiffrement ;**
- **en raison du non-respect d'obligations liées à la sécurité (des données, de l'accès à Internet, etc.).**

En conséquence, l'employeur est légitime à déchiffrer les contenus de flux chiffrés transitant sur les postes de travail de ses salariés, mais uniquement de façon encadrée en raison des risques juridiques liés au déchiffrement.

la confiscation, la fermeture d'établissement et l'exclusion des marchés publics peuvent être appliquées. Dans ce dernier cas, les articles 35, 36 et 37 de la LCEN prévoient notamment un an d'emprisonnement et 15.000 euros d'amende en cas de non déclaration ou de communication d'informations et jusqu'à deux ans d'emprisonnement et 30.000 euros d'amende en cas de non obtention d'autorisation.

49. CA Paris, 4 mai 2007 : il est en outre impossible de se prévaloir de la faute contractuelle du cocontractant du fait des comportements fautifs de ses propres salariés.

50. TGI Marseille, 11 juin 2003, D. 2003, le TGI avait jugé solidairement responsable une société qui avait fourni un accès internet à l'un de ses salariés qui avait diffusé des contenus préjudiciables sur internet par l'intermédiaire de pages personnelles.

51. Cass. Ass.plén., 19 mai 1988, n°87682.654 et CA Versailles 5ème ch. du 31 mars 2011 Michaël P.C./Mireille B.-P. illustrent la validation du licenciement pour faute grave d'un salarié ayant téléchargé illégalement des fichiers musicaux sur son poste de travail.

52. Un arrêt de la Cour d'appel de Paris du 4 février 2005 a pu jeter le doute sur l'application du régime juridique des FAI aux employeurs mettant à disposition de leurs salariés un accès à Internet.

53. Cette interprétation est présentée de manière détaillée dans la note technique «Recommandations de sécurité pour la mise en ouvre d'un système de journalisation» rédigée par l'ANSSI : http ://www.ssi.gouv.fr/IMG/pdf/NP_Journalisation_NoteTech.pdf

54. Articles 34 et 34 bis de la loi n°78-17 du 6 janvier 1978 relative à l'informatique, aux fichiers et aux libertés.

L'identification et la qualification des risques juridiques liés au déchiffrement

À titre de rappel, les articles 100 et suivants du code de la procédure pénale imposent une obligation légale de déchiffrement dans le cadre des interceptions judiciaires, les articles L 241-1 à L 245-3 du code de la sécurité intérieure dans le cadre des interceptions de sécurité et l'article 230-1 du code de procédure pénale dans le cadre d'une enquête ou d'une instruction [55].

En dehors de ces hypothèses, le déchiffrement d'un flux chiffré, en particulier lorsqu'il concerne la sphère personnelle, sur le lieu de travail pourrait notamment porter atteinte :

- au secret des correspondances privées [56] ;
- à la protection des données à caractère personnel [57] ;
- à la vie privée [58] des utilisateurs en dehors et dans le cadre du travail ;
- à la sensibilité des informations (déchiffrement de données protégées par le secret professionnel, par une réglementation telle la réglementation relative à la protection du patrimoine scientifique et technique de la nation [59] ou le secret de la défense nationale [60]).

Le déchiffrement soulève également des risques juridiques dans le cadre de l'intervention de tiers sur les systèmes d'information (sous-traitance, audit des systèmes découvrant des vulnérabilités contenant des données à caractère personnel, etc.). Afin de prévenir ces risques, il est primordial d'encadrer juridiquement cette possibilité dans le contrat d'externalisation, notamment en soumettant le sous-traitant à des obligations identiques à celles auxquelles se soumet l'employeur pour préserver sa responsabilité [61].

Le déchiffrement d'un flux chiffré peut porter atteinte aux libertés individuelles et engager la responsabilité de l'employeur qui n'aurait pas prévu les mesures destinées à préserver celles-ci.

55. L'article 434-15-2 du code pénal prévoit également trois ans d'emprisonnement et 45.000 euros d'amende si le détenteur des clés de déchiffrement de données chiffrées ne les fournit pas à l'autorité judiciaire et aux services d'investigation qui peuvent en à avoir besoin dans le cadre d'une enquête pénale.
56. Article 226-15 du code pénal.
57. Articles 226-16 à 226-24 du code pénal.
58. Articles 226-1 à 226-7 du code pénal.
59. La protection du potentiel scientifique et technique de la nation est définie par le décret n°2011-1425 du 2 novembre 2011 portant application de l'article 413-7 du code pénal et relatif à la protection du potentiel scientifique et technique de la nation et par l'arrêté du Premier ministre du 3 juillet 2012 relatif à la protection du potentiel scientifique et technique de la nation.
60. Instruction générale interministérielle n°1300/SGDSN/PSE/PSD du 30 novembre 2011 sur la protection du secret de la défense nationale.
61. À titre d'exemple, T. Corr Nanterre, 10 novembre 2011 : la responsabilité pénale de l'entreprise est retenue du fait des infractions commises par son sous-traitant sur le système d'informations d'une autre entité. Il est donc primordial d'encadrer le contenu des contrats des sous-traitants et surtout le périmètre de leurs interventions pour ne pas risquer de contrevenir aux législations précisées dans la note. En outre, la mise en place de procédures de contrôle internes pour éviter toute ingérence d'un sous-traitant est recommandée (sous-traitant en charge de déchiffrer les contenus, par exemple, qui devra respecter les obligations mises à la charge d'un administrateur.).

Encadrement juridique du déchiffrement

Généralités

La mise en place par une entité d'un mécanisme de déchiffrement des flux doit s'accompagner du respect des principes généraux suivants [62] :

- transparence et loyauté de l'employeur vis-à-vis de ses salariés en les informant sur la nature des mesures informatiques prises sur le réseau informatique de l'entité et en recueillant leur consentement individuel sur la charte informatique ainsi qu'en consultant les instances représentatives du personnel [63] ;
- nécessité et proportionnalité [64] de la mise en place d'un outil de déchiffrement par rapport aux finalités annoncées par l'employeur (telles la sécurité du réseau, la protection des données sensibles de l'entité) qui imposent que les mesures prises soient justifiées par la nature de la tâche et proportionnées à la finalité ;
- la protection des données à caractère personnel et notamment la vérification que les formalités accomplies auprès de la Commission nationale de l'informatique et des libertés (CNIL) couvrent les traitements et les données opérées par de tels mécanismes ;
- le contrôle de l'accès aux outils de déchiffrement (logiciels de déchiffrement, clés de chiffrement des matériels ou des utilisateurs) en particulier lorsqu'une mauvaise utilisation peut engager la responsabilité d'un tiers (salarié notamment). Les accès aux clés de chiffrement notamment lorsqu'il s'agit d'un accès au séquestre des clés des utilisateurs doivent être journalisés et être prévu dans la charte informatique de l'entité.

Le déchiffrement ne doit pas être mis en place sans avoir satisfait à des formalités légales et sans avoir anticipé sa mise en place pour assurer son opposabilité aux salariés de l'entité.

Encadrement juridique de la fonction d'administrateur de réseau ou de parc informatique en matière de déchiffrement

L'administrateur a pour mission d'assurer le fonctionnement normal et la sécurité des réseaux et systèmes dont il a la charge. En conséquence de cette mission, l'administrateur est tenu par une obligation de confidentialité et ne doit donc pas divulguer des informations dont il a eu connaissance dans le cadre de ses fonctions.

Selon la jurisprudence [65], l'administrateur peut, dans le cadre de ses fonctions et en particulier pour garantir la sécurité du réseau, de manière intentionnelle ou non, avoir accès à des informations relevant de la vie privée et des correspondances privées des utilisateurs. L'accès à de telles données ne peut être justifié que dans les cas où le bon fonctionnement et le maintien en condition de sécurité du réseau des systèmes informatiques ne peuvent être assurés par d'autres moyens moins intrusifs. L'utilisateur doit être présent, ou « dûment appelé » [66], en cas de lecture de contenus identifiés comme personnels ou privés [67].

62. Ces principes généraux sont les principes retenus dans les jurisprudences relatives au droit du travail et dont l'appréciation du respect reste soumise à l'appréciation souveraine des juges.
63. Article 2323-32 du code du travail.
64. Article 1121-1 du code du travail.
65. CA de Paris, 11ème chambre, 17 décembre 2001 : « [i]l est dans la fonction des administrateurs de réseaux d'assurer le fonctionnement normal de ceux-ci ainsi que leur sécurité ce qui entraîne entre autre, qu'ils aient accès aux messageries et à leur contenu [...]. Par contre, la divulgation du contenu des messages, et notamment du dernier qui concernait le conflit latent dont le laboratoire était le cadre, ne relevait pas de cet objectif ».
66. Cass, Soc., 17 juin 2009, n° pourvoi 08-40274.
67. Cass, Soc., 17 mai 2005, n° pourvoi 03-40017.

L'administrateur, fonctionnaire ou tout agent public contractuel, est tenu par une obligation de dénonciation de portée générale, qui est de nature à le délier de son obligation de secret professionnel y compris en cas de délit commis par un membre de sa hiérarchie dans l'exercice de ses fonctions [68].

L'administrateur peut dans le cadre de ses missions déchiffrer des données tout en restant soumis à une obligation de confidentialité.

En conclusion, la mise en place d'un dispositif de déchiffrement par l'entité doit être encadrée juridiquement :

- **par la charte d'utilisation des moyens informatiques et de communications électroniques rédigée par l'employeur et annexée au règlement intérieur de l'entité, après consultation des instances représentatives du personnel. Celle-ci devra prévoir, notamment, les règles d'utilisation des moyens mis à disposition par l'employeur, les modalités d'accès aux données et aux équipements, l'hypothèse, les sanctions en cas de non-respect, etc. ;**
- **par la mise en place d'un administrateur expressément autorisé à accéder aux contenus déchiffrés moyennant le respect d'une obligation de confidentialité qui le lie, y compris à l'égard de l'employeur ;**
- **par la politique de sécurité des systèmes d'information de l'entité qui devra envisager cette hypothèse, et éventuellement le cas des sous-traitants ;**
- **par les déclarations adaptées à la CNIL en considérant avec soin les finalités pour lesquelles le déchiffrement est envisagé.**

68. Article 40 alinéa 2 du code de procédure pénale.

B Suites cryptographiques acceptables

L'ensemble des suites cryptographiques qu'il est possible d'employer avec TLS sont référencées sur le site de l'*IANA* [69]. Il en existe plus de 300.

Chaque suite cryptographique est identifiée par un code hexadécimal ainsi que par une description qui obéit à la convention de nommage suivante :

$$Clé\_[Auth]\_Sym\_Hach$$

Détail des champs :

- *Clé* : mécanisme d'échange de clés (ex : ECDHE) ;
- *Auth* : algorithme (facultatif) utilisé pour l'authentification des parties (ex : RSA) ;
- *Sym* : algorithme de chiffrement symétrique utilisé pour chiffrer les données ; il est accompagné de la taille de clé en bits ainsi que du mode utilisé (ex : AES_128_GCM) ;
- *Hach* : fonction de hachage utilisée pour protéger en intégrité les échanges (ex : SHA256).

Il est souvent très difficile de restreindre la liste des suites cryptographiques en se basant uniquement sur les exigences listées dans le RGS. Cela conduit en effet à la définition d'une liste très réduite ne permettant pas d'assurer une compatibilité suffisante pour pouvoir être utilisée dans un environnement de production hétérogène. Cependant les suites à privilégier doivent permettre la *PFS* et doivent employer des algorithmes et des tailles de clés robustes au sens RGS du terme.

Les suites cryptographiques qu'il est possible de considérer comme acceptables [70], pour TLS uniquement (v1.0, 1.1 et 1.2), doivent respecter les recommandations suivantes :

- l'usage de suites dont l'une des composantes est à NULL est à proscrire ;
- l'usage de suites permettant une authentification anonyme (anon) est à proscrire ;
- à moins qu'elles ne soient explicitement souhaitées, les suites reposant sur l'un des mécanismes d'échange de clés suivant sont à proscrire : PSK, SRP, KRB5 ;
- l'usage de suites utilisant MD5 comme fonction de hachage est à proscrire ;
- l'usage de suites utilisant un des algorithmes de chiffrement symétrique suivant est à proscrire : DES, RC2, RC4 ;
- l'usage de suites utilisant un mécanisme d'échange de clés basé sur DSS est à proscrire ;
- l'usage de suites utilisant SHA1 comme fonction de hachage est toléré, mais il est conseillé de privilégier des fonctions plus robustes telles que SHA256 ;
- l'usage de suites utilisant 3DES comme algorithme de chiffrement symétrique est toléré en dernier recours, mais il est conseillé d'utiliser des algorithmes robustes tels que AES128 ;
- si TLS v1.0 est activé, il est nécessaire de s'assurer que l'implémentation du composant utilisé (client ou serveur TLS) inclut les contre mesures permettant de rendre inopérante l'attaque *BEAST* sur le mode CBC des algorithmes concernés (CVE 2011-3389 [71]).

69. La liste de l'ensemble des suites : http://iana.org/assignments/tls-parameters/tls-parameters.xhtml.
70. À la date de rédaction de ce document.
71. http://www.cvedetails.com/cve/CVE-2011-3389.