# HOMOLOGATION ET MAÎTRISE DU SI

## Les 3 axes à ne pas négliger

Think
Create
Digitize

Magellan Sécurité

# SOMMAIRE

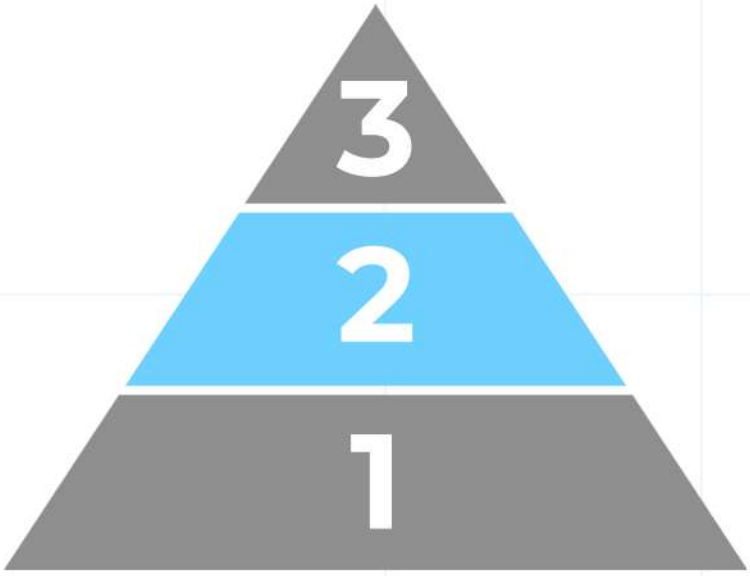


Dossier écrit par Eloi GUILLAUD, consultant senior Magellan Sécurité

# SSI et maîtrise du SI, vastes sujets ?

**Quand on parle de Sensibilisation Sécurité des SI (SSI), on désigne et mélange parfois trois niveaux bien distincts de maturité Cyber :**

1. **La sensibilisation des collaborateurs aux menaces extérieures**, ce qui englobe la communication externe d'informations, la prévention contre le phishing, et l'hygiène cyber du poste de travail
2. **La maîtrise du SI**, pour s'appuyer sur un socle technique de confiance et une organisation éprouvée.
3. **Le cloisonnement interne de l'information** entre VIP, Key People et sachants, qui concerne surtout des mesures palliatives aux menaces internes (fraude, corruption, espionnage etc...).




Si les étages inférieurs et supérieurs de la pyramide sont également importants pour augmenter son niveau de protection, le cœur des projets techniques d'envergure se concentrent sur la maîtrise du SI.

Ce sont ces projets qui permettent, quand ils portent leurs fruits, de s'inscrire dans une démarche de respect d'exigences RGS, LPM, ou autres.

On va donc s'intéresser ici à 3 problématiques concrètes de la maîtrise du SI qui d'après notre expérience, sont à la fois les plus difficiles à surmonter et les plus constatées dans les démarches d'audit et d'homologation.

# La Cartographie des assets

**La cartographie des assets d'un SI d'entreprise est un socle crucial pour toute DSI qui souhaite ne pas avancer à l'aveugle.**

Elle est aussi difficile à maintenir dans le temps (changement de services, migrations des infrastructures) et très coûteuses à acquérir, en l'absence totale de documentation existante (nécessite un projet long et vaste).

On remarque d'ailleurs que la cartographie applicative est souvent la plus dure à obtenir dans des SI plus « standards », tandis que dans le secteur de l'informatique industriel et scientifique, ce sont les infrastructures sous-jacentes qui sont souvent les moins répertoriées.

## Cas concret

**En quoi c'est important pour la Cyber ?**

En 2017, en pleine vague d'attaque WannaCry, nous avons pu constater chez un de nos clients une tentative de propagation interne du ransomware dans une de ses filiales. Il a fallu un certain temps d'investigation et de capture de trames réseaux pour comprendre que l'attaque avait pour origine un vieux fax oublié.

En effet, cette appliance de fax fonctionnait sous Windows 2003, avec donc SMBv1 (Server Message Block version 1) activé et fonctionnel. L'OS de l'appliance de fax n'était pas patchable. Elle ne figurait dans aucun fichier de l'entreprise et bien que non-utilisée depuis des années, n'avait jamais été décommissionnée.

# Solution

Bien sûr, le décommissionnement immédiat dans notre exemple est une première solution palliative.

Mais la **solution préventive** ici, est à penser en 3 étapes, pour une mise en place sur le long terme.

### 1 Acquisition et utilisation d'une solution de CMDB (Configuration Management Data Base)

La pierre angulaire de l'inventaire d'un SI, agissant comme référentiel des assets et de leurs nature. Cependant, il ne suffit pas d'en posséder une, il faut aussi la maintenir et l'exploiter dans le temps.

### 2 Adaptation des processus de mise en service

Ils doivent intégrer comme prérequis obligatoire le renseignement de cette CMDB. C'est à intégrer dans le process de CAB (Change Advisory Board) : un **non-renseignement** de la CMDB sera motif de **no-go** pour la mise en service de l'infrastructure ou de la solution.
Les équipes doivent y être sensibilisées, et être formées à son utilisation.

### 3 Financement des projets de décommissionnement

La phase de décommissionnement d'un projet de migration est souvent sous-évaluée voir tout simplement absente. Cela explique parfois la grande part d'assets obsolètes, qui continuent à exister et vivre.

Outre la part d'énergie consommée qui est rarement intégré dans les OPEX (dépenses d'exploitation), **quantifier le coût d'une attaque Cyber** via un élément obsolète peut aider à convaincre le management de financer ce type de projet. Il n'existe pas d'outil simple et exhaustif pour ce faire, mais partir d'un arrêt de service sur une période donnée, ou de la destruction/recréation d'une infrastructure précise permet souvent d'avoir des ordres de grandeur financiers du risque cyber.

# Patching et MCS

## Dans le quotidien de l'exploitation d'un SI, les urgences sont parfois rythmées par les différentes alertes de sécurité qui paraissent.

La problématique que les grandes entités rencontrent, c'est que plus le SI est hétérogène, plus les alertes potentiellement pertinentes sont nombreuses.

Il est difficile de déterminer facilement ce qui est pertinent ou pas, et de prioriser les alertes à traiter.

### Cas concret

Dans la vague de ransomware en 2017, les CVE (Common Vulnerabilities and Exposures) publiées par Microsoft ont joué un rôle crucial dans la prévention et la réaction rapide.

Plus récemment, on peut mentionner les vagues d'attaques ciblées dans un contexte international très tendu, en raison de la guerre en Ukraine.

En général, la protection préventive contre les nouvelles failles et les attaques type supply-chain passent par la veille et l'analyse des nouvelles CVE publiées par les différents éditeurs et du CERT officiel Français.

## Solution

Devant le manque de temps des équipes de RUN et le nombre d'alertes publiées, il est nécessaire d'établir un **CERT local**.

Un département CERT (Computer Emergency Response Team) est chargé d'effectuer ce travail de veille pour déterminer quand le SI est menacé par l'arrivée de nouvelles vulnérabilités et menaces.

Cette entité se fait le relais personnalisé du CERT FR, le site officiel de l'ANSSI dédié à la publication des alertes et des incidents de Cybersécurité.

C'est un travail qui doit être fait par un acteur sécurité de la DSI (RSSI, Analyste, Correspondant Local par exemple), et qui nécessite un temps non négligeable, car chaque CVE publiée doit être décortiquée.

Les efforts de patching ayant un coût (prestation extérieure, indisponibilité temporaire du service), on ne peut et ne doit pas se contenter de tout considérer comme critique.

Pour effectuer ce discernement, l'acteur de la DSI va s'appuyer sur les CVSS (Common Vulnerability Security Score) des CVE concernées. Il va se charger de récupérer les informations décrivant les scénarios d'exploitation des failles (sur le site du CERT-FR et sur les sites des éditeurs de solutions).

Il va également faire le tri, prioriser les actions de mise à jour et contacter les équipes d e RUN pour planifier les actions.

# Security by design

**La grande majorité des CVE mentionnés ci-dessus exploitent des vulnérabilités préexistantes dans les outils et solutions.**

Si certains protocoles historiques ne peuvent être réécrits ou remplacés, ce n'est pas le cas des applications développées depuis l'histoire moderne de l'informatique.

La problématique survient lorsqu'une entreprise souhaite développer en interne un outil ou acquérir une solution sur étagère : des précautions de conceptions et/ou de recette s'imposent.

## Cas concret

On dénombre une grande diversité d'attaques opportunistes se basant sur les failles usuelles suivantes :

**Insertion SQL dans la base de données de l'application**
Les formulaires présentés à l'utilisateurs non sécurisés permettent à un attaquant de supprimer des données en insérant une requête SQL, par exemple.

**Faille XSS et récupération des cookies**
Là encore, une interface web mal conçue et des formulaires non protégés permettent d'exécuter du code arbitrairement sur l'application, ou de récupérer des données dans les cookies.

**Utilisation de certificats auto-signés, ou non-vérification du certificat (nom, date d'expiration, révocation) par l'application**
Cela permet à un attaquant d'usurper son identité.

**Utilisation de protocoles de chiffrement faibles par défaut (md5, 3DES)**
Cela permet de casser le flux de données d'une application.

**Utilisation de mot de passe par défaut faibles**
Cela permet une simple attaque par Bruteforce/Dictionnaire (c'est le cas notamment de beaucoup d'objets connectés).

**Stockage en clair de mots de passe dans des fichiers accessibles par l'utilisateur**
Fichiers html, javascript, cookies, fichiers de configuration divers.

## Solution

La plupart de ces failles peuvent se régler facilement en adoptant de bonnes méthodes de développement. C'est donc une approche de sécurité dès la conception (security by design) qui va permettre de s'en prémunir.
Cette approche de Security By Design concerne presque tous les métiers de la DSI

**Pour les développeurs et les administrateurs (profils devops) :**

Sensibiliser les développeurs sur les principes de **Saltzer and Schroeder** :

• Principe d'accès via moindre privilège : on ne donnera à l'application que les droits strictement nécessaires pour s'exécuter sur l'infrastructure (pas de droits domain admin ou root, par exemple...)

• Autorisation par liste blanche plutôt que par liste noire: tout ce qui n'est pas explicitement autorisé est bloqué par défaut, et non l'inverse

• Séparation des responsabilités : autant que possible dans le cas d'une application métier, l'administration fonctionnelle (Métier) doit être dissociée de l'administration technique (DSI). Chacun doit avoir son propre profil et son propre périmètre.

• Traçabilité des actions : gestion adéquate des logs intégrées dans l'application

• Economie d'opérations : Garder la solution la plus simple possible pour faciliter son adoption par les utilisateurs

**Pour les administrateurs et les profils devops :**
Moindre privilège requis (on n'utilisera pas un privilège domaine admin pour exécuter un script powershell de rotation des logs, par exemple).
De même, on n'utilisera pas un accès root pour de l'orchestration de tâches courantes.

**Pour les architectes et les décideurs IT :**
Pour limiter les dégâts en cas de compromissions, il faut penser le cloisonnement applicatif dans les infrastructures au démarrage des projets.