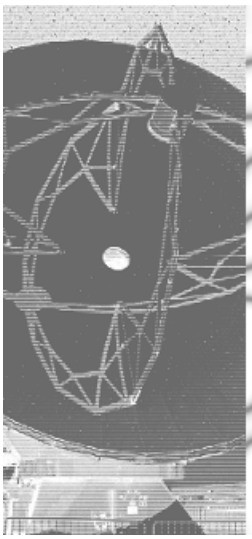

Chapitre 3. La cybersécurité dans le développement

Ce domaine de la cybersécurité s'intéresse aux mesures à mettre en place au cours de *projets de développement* de systèmes ou de services intégrés dans le SI. Pour nous, le terme « développement » couvre à la fois la création de nouveaux services et la modification de services existants.

Qu'entend-on par « développement » ?

Le SI d'une organisation est un objet en constante évolution – le volume et la fréquence de ces évolutions peuvent bien sûr dépendre du contexte, mais dans toute organisation de taille significative, il y a toujours au moins une ou deux évolutions en cours. Qu'il s'agisse de modifier des services existants, ou de créer de nouveaux services, chacune de ces évolutions présente un certain nombre de traits qui permettent de la caractériser :

- L'évolution entraîne un volume plus ou moins important de *développement* et d'*intégration* :
  - des *développements informatiques* – du *codage* nécessitant l'emploi d'*outils* et de *langages de programmation* –, ce qui comprend également le *test* de ces développements informatiques,
  - des *installations* de nouveaux *équipements* informatiques (serveurs, équipements réseau) ou des *remplacements* des équipements existants par des nouveaux,

- des *installations*, des *mises-à-jour* ou des *remplacements* d'intergiciels sur les équipements informatiques : systèmes d'exploitation, SGBD, serveurs Web, serveurs d'application, environnements d'exécution de scripts, librairies logicielles, environnement bureautique, etc.
- des *configurations* d'intergiciels nouvellement installés ou des modifications de celle d'intergiciels existants.

Toutes ces actions visent à *construire un nouveau sous-système* du SI ou à *modifier un sous-système existant* ;

- L'**intégration** du nouveau sous-système **au sein du SI** doit être assurée, ce qui impose le respect d'un certain nombre de **règles** – par exemple des *règles de sécurité de l'urbanisme du SI*. En cas de modification d'un sous-système existant, la modification doit maintenir le respect des règles d'intégration du sous-système dans le SI ;
- Les actions concrètes constituant l'évolution, dont nous avons donné des exemples ci-dessus, doivent rentrer dans le cadre d'une **conception** visant à répondre au *besoin métier* émanant d'une ou plusieurs *parties prenantes* et ayant fait l'objet d'une **spécification**.

  En plus de **vérifier** les développements informatiques en interne à l'équipe de développement, comme énoncé ci-dessus, on doit effectuer la **validation** de la satisfaction des besoins métiers par le sous-système développé, sur la base de la spécification, sous l'autorité des parties prenantes qui sont *maîtresses d'ouvrage* du sous-système – ce sont notamment elles qui financent l'évolution ;
- En plus de la *conception*, du *codage* et de la *vérification / validation* des développements, il faut aussi décrire le **déploiement** de l'évolution au sein du SI, sous la forme d'un certain nombre d'étapes techniques visant à assurer la *transition* du SI *de production* d'un état antérieur à l'évolution vers un état où elle aura été réalisée. La date et l'heure où ces étapes vont être réalisées dans le SI doit faire l'objet d'une planification et d'un suivi, au même titre que la planification des développements ;
- En parallèle des activités *techniques* énoncées ci-dessus, l'évolution peut également concerner l'environnement *organisationnel*, voire *physique*, du sous-système, c'est-à-dire les modalités de fourniture du *service* que le sous-système est censé soutenir.

  Par exemple, l'évolution peut porter sur :
  - le modèle de commande ou de facturation du service,
  - les conditions générales d'utilisation du service, les responsabilités contractuelles ou légales de l'organisation,
  - la répartition des tâches associées à la fourniture du service au sein des équipes de l'organisation,

- le recours à l'externalisation d'une tâche associée à la fourniture du service, ou l'internalisation d'une tâche auparavant externalisée,
- un ajustement significatif à la hausse ou à la baisse du volume des équipes dédiées à la fourniture du service, pouvant impliquer un changement des procédures ou des outils utilisés dans le cadre du service,
- le déménagement de la plate-forme et des équipes du service dans d'autres locaux,
- etc.

Le SI présente donc un « volet courant » de **développements** visant à mettre en œuvre des *évolutions*. Chacun de ces développements devrait être compris par la direction de l'organisation comme un **projet** – c'est-à-dire « *un effort temporaire entrepris pour créer un produit unique, un service ou un résultat* » [27] –, et géré comme tel, par des équipes dédiées aux développements dans le cadre de projets et la gestion desdits projets.

Pourquoi faut-il mettre en œuvre la cybersécurité dans les développements ?

En quoi ce volant courant de développements du SI concerne-t-il le RSSI ? Et surtout : *pourquoi faut-il que lui où ses équipes aillent suivre de l'intérieur le processus de développement, et aient la possibilité d'intervenir dans ce déroulement* ?

En effet, on pourrait se dire qu'après tout, à partir du moment où la posture de cybersécurité du SI est correctement mise en place, celui-ci devrait être suffisamment robuste vis-à-vis des failles de sécurité qui pourraient être introduites par des développements particuliers. En somme, le SMSI fixerait des règles d'urbanisme, et les développements devrait respecter ces règles sous peine que le déploiement se voie refusé. Et puis, s'il y avait lieu d'intégrer des mesures de sécurité techniques à l'intérieur du développement, il suffirait de s'en préoccuper uniquement une fois les développements terminés : on appliquerait une « couche de cybersécurité » sur le code développé et sur les configurations préparées, juste avant le déploiement.

Cette approche est erronée. Le RSSI *doit se préoccuper* de la cybersécurité au cours des développements pour au moins deux raisons :

- La *confiance* : le développement n'est pas seulement la construction ou la modification d'un système, c'est aussi la construction de la *confiance dans le fait que ce système vérifie bien les propriétés qu'on attend de lui*. L'*Assurance Qualité* construit la confiance dans le fait que le système se comporte bien de la manière

attendue[1]. L'**Assurance Sécurité** construit la confiance dans le fait que le système répond à la problématique de sécurité constituée par les *menaces* issues de l'*estimation des risques*, les *exigences externes* de sécurité et les *bonnes pratiques internes* de sécurité ;

- L'*efficacité du développement* et la *réduction des coûts* : l'expérience montre que la démarche consistant à « patcher » les problèmes de cybersécurité après le codage et l'intégration est *source d'inefficacité* dans la mise en œuvre des mesures de sécurité.

  Il faut bien comprendre que certains choix que l'on peut faire au niveau de la conception peuvent avoir une influence majeure sur la posture de cybersécurité du système. Si la problématique de cybersécurité qui s'impose au système à développer n'est pas prise en compte au stade de la conception, on s'expose au risque de constater, *à la fin du développement*, que ces choix obèrent partiellement, voire totalement, la possibilité de répondre à cette problématique. L'expérience enseigne que des détails *mineurs* dans la conception, le codage ou la configuration peuvent avoir un impact *majeur* sur la sécurité du système.

  L'équipe de développement, partie prenante des choix de conception, et celle des experts en cybersécurité, porteuse de l'Assurance Sécurité du système, seront alors confrontées à un dilemme entre, d'une part *défaire toute ou partie des choix de conception et reprendre une partie des développements*, et, d'autre part, *renoncer à répondre à toute ou partie de la problématique de cybersécurité* – dilemme évidemment porteur de conflits entre elles.

  Dans l'hypothèse où on choisirait de reprendre en partie la conception et les développements, le coût en sera nécessairement plus élevé, en application du principe attribué à B. Boehm et V. Basili [28], [29] qu'on peut résumer par : « *plus une erreur est introduite tôt et corrigée tard dans le processus de développement, plus elle coûte cher à corriger* ».

  Ainsi, l'application de mesures de sécurisation du système postérieurement à son développement est non seulement une source d'inefficacité dans la mise en œuvre de la cybersécurité, mais également une *source de surcoûts* dans le développement.

---

1 En toute rigueur, la portée de l'Assurance Qualité déborde du cadre strict de la vérification du comportement fonctionnel du système. Mais, dans la pratique, quand on parle d'« Assurance Qualité » aux développeurs informatiques, ils pensent d'abord et avant tout aux tests visant à vérifier le comportement fonctionnel du système lors de son développement.

Qu'est-ce que la cybersécurité dans le développement ?

Il nous paraît donc important de donner la définition suivante de la cybersécurité dans le développement, en insistant sur la nécessité que ces activités soient *embarquées* au sein du processus de développement standard :

> La **cybersécurité dans le développement** est un ensemble d'activités dédiées à la sécurité **au sein du processus de développement d'un système informatique**, ayant pour but de développer et d'intégrer des systèmes informatiques qui **présentent des caractéristiques de sécurité adaptées**.

Précisons quelles sont les « caractéristiques de sécurité adaptées » que nous assignons comme objectif aux activités de cybersécurité dans le développement. Il s'agit essentiellement que le système développé :

- Réponde, par des *fonctions* et des *mécanismes de sécurité* pertinents, à la *problématique de sécurité*, issue de l'estimation des risques, à laquelle le système va être confronté en opération.

  La problématique de sécurité se traduit sous la forme :
  - de *menaces* perpétrées par des *attaquants*, à chacune desquelles est associé un *risque*, qu'il s'agit de *réduire*,
  - d'*exigences de sécurité légales*, *réglementaires* ou *métier* – par exemple le RGPD, ou la LCEN –, auxquelles le système doit être *conforme* ;
- Soutienne, par les fonctions et mécanismes de sécurité développés, les besoins techniques des *processus de sécurité opérationnelle* (cf. chapitre 4) ;
- Présente le moins possible de *vulnérabilités* à l'issue du développement.

Parmi ces trois points, le dernier est pratiquement le plus facile à mettre en œuvre, parce qu'il est basé sur des méthodes pouvant être exécutées de manière relativement autonome par les équipes de développement.

Le soutien aux processus de sécurité opérationnels demande une approche proactive vis-à-vis des équipes de cybersécurité opérationnelle. Elle dépend du degré de maturité de l'organisation concernant la prise en compte des problèmes spécifiques portant sur la transition entre le développement et l'exploitation [2].

---

2 Dans certaines organisations, des responsables ayant un rôle dédié, les *delivery managers*, ont spécifiquement pour tâche d'assurer une communication et une transition harmonieuse entre les développements et les opérations.

Quant au fait de répondre à la problématique de sécurité par les fonctions et les mécanismes de sécurité – ce que nous avons appelé plus haut l'*Assurance Sécurité* –, force est de constater que ce point n'est encore que trop rarement appliqué. Parmi les causes, nous pouvons citer le fait que la démarche d'estimation des risques n'est pas mise en œuvre par toutes les DSI, loin s'en faut, car elle exige une certaine maturité dans l'approche des risques au niveau métier, et elle est ainsi souvent vue comme trop complexe et trop sophistiquée.

En règle générale, en l'absence d'effort spécifique pour aligner les fonctions développées sur la problématique de sécurité, les développeurs vont s'appuyer, s'ils sont consciencieux et sensibilisés au sujet de la cybersécurité, sur des bonnes pratiques relevant de l'état de l'art (par ex. : « mettre du HTTPS partout »), et par la prise en compte des aspects réglementaires ou légaux, à condition que la maîtrise d'ouvrage en fasse mention dans les exigences de sécurité en entrée du projet de développement.

Avant de proposer un modèle d'activités de cybersécurité à embarquer dans le processus de développement (§3.3), il nous faut préciser quelques *notions-clés* qui vont faire l'objet desdites activités (§3.1) et définir quels sont les *rôles* pour leur mise-en-œuvre (§3.2).

3.1 Notions-clés pour la cybersécurité dans le développement

La prise en compte de la cybersécurité au cours du développement permet de *construire la confiance* dans le fait que les mesures de cybersécurité implémentées au cours du développement contrent bien les *menaces* identifiées sur le système en amont du développement.

Pour contrer les menaces identifiées, le système développé doit présenter des *fonctions de sécurité*, du genre de celles que nous avons illustrées aux §2.3 à 2.6 (contrôle d'accès, identification et authentification, journalisation, etc.).

La construction de la confiance est réalisée par des analyses démontrant que les fonctions de sécurité développées dans le système sont bien pertinentes et efficaces pour contrer les menaces retenues. C'est ce qu'on appelle les *argumentaires de sécurité*.

Pour réaliser les menaces, les attaquants s'appuient sur des *vulnérabilités*. Ces vulnérabilités peuvent permettre par exemple de contourner ou de désactiver les fonctions de sécurité. Certaines d'entre elles sont constituées par des *bugs de sécurité* introduits, généralement involontairement, dans le développement. La prise en

compte de la cybersécurité dans le développement implique d'*identifier* et de *contrôler la correction* de ces bugs de sécurité.

Les fonctions de sécurité s'appuient sur des *mécanismes de sécurité*, qui sont une description détaillée et technique des algorithmes, protocoles et caractéristiques de conception qui sous-tendent ces fonctions.

3.1.1 Menace

Au §2.10, p. 68, nous avons défini la notion de *risque*, dont nous avons expliqué que les deux composantes permettant de le mesurer étaient l'*impact* et la *vraisemblance*. Cette dernière représente « *la possibilité que les scénarios du risque se réalisent, indépendamment des conséquences* ».

Une **menace** raffine la notion de risque, en ajoutant des informations plus détaillées sur les scénarios par lesquels il peut se réaliser. Il s'agit essentiellement du passage du « quoi » au « comment » au cours de la réflexion stratégique sur les risques. En décrivant les menaces que l'on estime peser sur le système à développer, ou sur le SI dans son ensemble, on rend explicites les facteurs sur lesquels est fondée l'estimation de la *vraisemblance* des risques. De plus, la description des menaces peut aussi contribuer à l'estimation de l'*impact* des risques.

L'ensemble des menaces identifiées sur le système forme un *modèle* (q.v.) qui constitue un raffinement de la *problématique de cybersécurité* constituée par le résultat de l'estimation des risques.

Une menace peut correspondre à plusieurs risques. Alors qu'un risque porte sur les *biens essentiels* (cf. figure 8, p. 76) du service auquel contribue le système, une menace se décrit plutôt en termes d'événements affectant les ressources informatiques du système, c'est-à-dire ses *biens support*.

Une menace est caractérisée par :

- Un *agent* ou un *facteur* :
  - *attaquant* humain, ou agent informatique agissant pour son compte (*maliciel*), dans le but de *nuire volontairement* à l'organisation ou aux infrastructures informatiques en général,
  - *agent* humain à l'origine de la menace, par maladresse, négligence, ignorance ou incompétence, et ce *sans intention de nuire*,
  - phénomène naturel *accidentel* : panne réseau, « glitch » causant une erreur dans la mémoire, … ;
- Un ou plusieurs *scénarios d'attaque* ;
- Les *ressources informatiques* attaquées au sein de l'architecture du système, au cours du déroulement du scénario ;

- Les *vulnérabilités* (q.v.) exploitées au cours du déroulement du scénario. Ces vulnérabilités peuvent être :
  - *avérées*, ou encore *effectives*, si elles correspondent à des faiblesses constatées voire inhérentes aux choix d'architecture ou à l'organisation,
  - ou bien *supposées*, ou encore *hypothétiques*, si on les formule en se plaçant dans l'hypothèse où, malgré les mesures d'Assurance Sécurité du développement, une vulnérabilité d'un type donné existerait sur le logiciel, qu'il ait été développé par l'équipe projet, ou bien acheté sur étagère ;
- Le *préjudice causé* – i.e. les impacts possibles des risques qui correspondent à la menace.

3.1.2 Vulnérabilités et bugs de sécurité

Contrairement à ce qu'on peut penser, la notion de vulnérabilité prête à énormément de quiproquos lors des débats entre « experts sécu ».

Les discussions portent souvent sur le caractère de vulnérabilité d'un fait technique relevé : certains ne voient pas la nécessité de rattacher une vulnérabilité avérée sur le système à une des menaces retenues, et ne voient même pas l'utilité de prendre connaissance de la problématique de sécurité du système analysé, produisant, à l'issue de leurs prestations d'audit, des listes de « vulnérabilités qui ne servent à rien » (pour les attaquants). Dans le même ordre d'idée, d'autres confondent « vulnérabilité » et « non-conformité aux exigences de sécurité, ou aux bonnes pratiques », surtout lorsqu'ils appliquent à votre système leur propre référentiel de bonnes pratiques, sans se poser le moins du monde la question de sa pertinence.

D'autres encore ont du mal à comprendre qu'une vulnérabilité avérée n'a pas forcément vocation à être corrigée, et qu'on peut « vivre avec », provisoirement ou définitivement, dans certaines circonstances.

Sans parler de ceux qui ne se donnent pas la peine de vérifier si les vulnérabilités de leur liste sont avérées sur votre système. Lors de la restitution de l'audit, ils déroulent leur liste, accompagnée de l'ordonnance pour y remédier (par ex. : « tout passer en HTTPS »), à appliquer obligatoirement.

Et encore, nous parlons ici des experts qui parlent de vulnérabilités techniques. Il y a aussi ceux qui voient des vulnérabilités dans l'organisation, dans les personnes, ou dans l'environnement physique, et qui les situent à un niveau de description plus élevé que le niveau technique.

On peut alors se poser la question suivante : un expert en cybersécurité technique, nourri à BugTraq – liste de vulnérabilités techniques sur les logiciels du