

Sécurité applicative : l'avenir est aux « secure coders » agiles

Les applications sont devenues la cible privilégiée des cyberattaques qui tirent avant tout profit du hiatus culturel entre développement et sécurité. En acculturant toute l'équipe agile à la sécurité et en formant les développeurs pour en faire des « secure coders », quelques semaines d'un programme d'insertion de la sécurité dans l'agile renforce considérablement la sécurité applicative.

Face à la multiplication des cyberattaques, et au coût grandissant de leurs conséquences, les entreprises ont investi massivement dans la sécurité. Elles se sont équipées de pare-feux et de sondes de détection, elles ont mis en place des *Security Operations Center* (SOC), et leurs infrastructures sont aujourd'hui beaucoup moins vulnérables qu'il y a quelques années.

Mais loin de décourager les pirates, ces mesures de protection les ont réorientés vers de nouvelles cibles : les applications.

Avec la révolution digitale, le nombre d'applications a explosé. Moins sécurisées et moins surveillées que l'infrastructure, et jamais très éloignées de données ou de systèmes sensibles, elles sont particulièrement exposées. En outre, avec l'essor de l'agile, le nombre des évolutions, leur rythme et la rapidité de leur propagation rendent difficile la mise en place de solutions de sécurité dédiées comme les Web Application Firewalls (WAF). C'est pourquoi on estime aujourd'hui que 75 % à 80 % des attaques empruntent la voie applicative.

“ Il faut replacer la sécurité dans la perspective de la valeur métier et de l'expérience utilisateur car c'est le langage et la préoccupation de l'équipe agile. ”

Sébastien Tabarly
Directeur offre Agile4Security
Sogeti France



Une question de culture, pas d'outils

Extension de la philosophie DevOps, DevSecOps constitue une réponse importante. En intégrant les enjeux de sécurité aux projets agiles, cette approche permet en effet de mieux les prendre en compte et de limiter drastiquement – et à moindre coût – les vulnérabilités applicatives. Mais, trop souvent, DevSecOps se trouve réduit à une question d'outillage, comme l'intégration de tests de sécurité au pipeline de déploiement continu. Ce faisant, on néglige la cause profonde des failles de sécurité des applicatifs : la dimension culturelle.

Par formation et tradition, les mondes du développement et de la sécurité sont en effet assez éloignés, pour ne pas dire hostiles. Pour les développeurs, la sécurité est perçue comme une discipline rébarbative pratiquée par des spécialistes obtus dont les exigences formulées après coup n'ont aucune considération pour leur travail ni pour les enjeux auxquels ils doivent répondre. À l'inverse, pour la sécurité, les développeurs sont brouillons, négligents et inconscients des risques qu'ils font courir à l'entreprise. La communication est compliquée et ces difficultés nuisent avant tout à la sécurisation des applicatifs.

Il n'y a cependant là aucune fatalité. Les développeurs traitent couramment les bugs fonctionnels et ils peuvent donc tout aussi bien traiter non pas des « vulnérabilités » - ce qui ne correspond pas à leur logique - mais ce que l'on désignera plutôt comme des « anomalies de sécurité », soit des insuffisances fonctionnelles susceptibles d'être exploitées pour dévoyer l'application. En d'autres termes, il faut replacer la sécurité dans la perspective de la valeur métier et de l'expérience utilisateur, et parler le langage du développeur. Il faut se préoccuper de l'humain et moins de l'outil.

Sensibiliser et former les développeurs

Pour opérer ce « shift-left », c'est-à-dire faire remonter la prise en compte de la sécurité vers l'amont du cycle de vie applicatif, la priorité est de sensibiliser et de former les développeurs. Il ne s'agit pas d'en faire des experts en sécurité car ils n'en ont ni la vocation, ni l'envie, ni le temps. En revanche, ils peuvent devenir des « secure coders », c'est-à-dire des spécialistes de la sécurité dans leur langage (Java, Python, C++...), capables de réaliser eux-mêmes des tests de sécurité statiques et dynamiques, d'en comprendre les résultats et d'apporter les corrections nécessaires.

La clé du succès est d'assurer le transfert de connaissances et de compétences par des personnes à la fois spécialistes du développement et de la sécurité, personnes qui auront par conséquent la capacité à établir l'indispensable passerelle culturelle. Simultanément, il ne faut pas perdre de vue l'objectif agile et donc s'assurer que l'impact sera minimal sur la vélocité. La démarche de formation se concentrera donc sur le langage du développeur, ce qui, en outre, valorisera et renforcera ses compétences. On recourra à des exercices assez courts et interactifs pour ne pas perturber la journée de travail et palier l'aspect rébarbatif et peu efficace des sessions de sensibilisation et autres vidéos de e-learning sécurité. Enfin, un accompagnement personnalisé permettra d'intégrer les nouveaux réflexes de sécurité aux pratiques quotidiennes.

3 points à retenir

- Ciblées par les pirates, les applications pâtissent des difficultés de communication entre le monde du développement et celui de la sécurité.
- La principale réponse n'est pas l'outillage mais l'acculturation et la formation de toute l'équipe agile, au premier rang de laquelle les développeurs.
- Une démarche en phase avec les principes et les objectifs de l'agile permet aux développeurs d'acquérir rapidement des compétences et des bonnes pratiques de sécurité.



Toutefois, il ne faut pas seulement embarquer les développeurs mais acculturer toute l'équipe agile. Le *scrum master* devient le dépositaire et le promoteur de la démarche, le responsable technique le superviseur de la sécurité et le *product owner* doit lui aussi prendre conscience de la valeur qu'apporte une sécurité applicative renforcée. Enfin, un travail similaire doit être mené en parallèle avec les Ops, dont les interventions sur l'infrastructure ont elles aussi des impacts sur la sécurité.

Transfert de compétences, pas de responsabilités

Pour autant, le « shift-left » n'est pas un transfert de responsabilité. Les équipes sécurité conservent leur rôle de sentinelle du risque mais elles tirent aussi bénéfice de cette approche, qui leur permet de se concentrer sur des tests approfondis et l'explication de leurs résultats. C'est pourquoi il faut aussi les inclure pour leur faire prendre conscience du fossé qui les sépare des développeurs et les aider à le surmonter tant sur le fond que sur la forme.

L'agile constitue un terrain favorable à cette transformation somme toute importante. Les développeurs sont dans une logique d'amélioration continue : progresser dans leur langage les intéresse et, lorsque l'on parle de sécurité dans leur langage, ils sont souvent très demandeurs. Au point qu'il faut parfois les réfréner ! Mais une fois qu'ils connaissent, par exemple, les mécanismes d'injection et l'importance de « désinfecter » ce qui vient du navigateur, ils y prennent scrupuleusement garde. Quelques semaines suffisent pour avoir des « secure coders » et bâtir à moindre frais des applications qui offrent un minimum de prise aux pirates. Étant donné les enjeux, c'est un investissement que les entreprises auraient tort de négliger.