Validation des données entrantes

Les données entrant dans une application (qu'il s'agisse de celles saisies par un utilisateur interactif ou celles envoyées par un procédé de transmission quel qu'il soit), sont une source de préoccupation particulière en cyber sécurité.

En effet, en l'absence de précautions particulières, on peut imaginer qu'un individu malveillant utilise ces entrées de données pour détourner le comportement prévu de l'application en injectant du contenu actif non envisagé par les concepteurs (injection SQL, injection de code, injection XML, etc.). De tels détournements peuvent remettre en cause la confidentialité et l'intégrité des données.

Pour se prémunir de ce risque, il est nécessaire de procéder systématiquement à une étape de validation avant traitement. La validation pourra prendre 2 formes distinctes qui peuvent se cumuler :

- contrôle de la **nature des données** par rapport à celles attendues (un email est-il un email valide ? une date est-elle une date valide ? etc.) ;
- **normalisation des contenus** de manière à assainir les entrées (éliminer tout caractère inadapté, échapper des caractères à usage possiblement détournés, etc.)

Aujourd'hui, de nombreux outils existent sur lesquels s'appuyer pour la validation :

- les différents <input type> du html5 ;
- les fonctions de filtrage (**filter_input** en PHP) et d'échappement (**addslashes**, **htmlspecialchars**, **htmlentities** en PHP) ;
- les expressions régulières (**Regex**) accessibles dans tous les langages ;

Considérant que le Client/Serveur est partout, pour maximiser la qualité du procédé, il faut prévoir de réaliser une validation côté Client ET côté Serveur !

Pour ce qui est des **injections SQL**, on n'hésitera pas à cumuler validation **et** requêtes préparées.