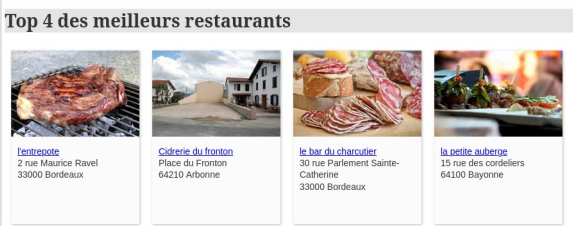



## Ticket n°9

Titre du ticket : faille de sécurité sur la page d'authentification

<b>Type du ticket</b> : incident (évolution/incident)	<b>Niveau de gravité</b> : <input type="checkbox"/> Bloquant <input checked="" type="checkbox"/> Majeur <input type="checkbox"/> Mineur
<b>Émetteur</b> : Nicolas BOURGEOIS (nom de l'émetteur)	<b>Date signalement</b> : 26/09/2022 (jj/mm/aaaa)
<b>Assignation</b> : Maindron Kélian (nom du membre de l'équipe en charge du ticket)	<b>Date de résolution souhaitée</b> : 03/10/2022 (jj/mm/aaaa)
<b>Application concernée</b> : R3st0.fr <b>Version</b> : 1.0 initiale – septembre 2022	
<b>Description du problème</b> (avec éventuelles captures d'écran, messages d'erreurs) :  On m'a signalé qu'une attaque par injection SQL est possible sur la page de connexion. <b>Scénario</b> : L'utilisateur saisit la chaîne de caractères suivante dans le champ de saisie de l'email : zzz' OR 1 = 1 ; DELETE FROM photo WHERE '1' = '1' et une valeur quelconque dans le mot de passe. L'application refuse l'authentification en affichant le message d'erreur suivant : <div style="background-color: red; color: white; padding: 5px; margin: 10px 0;"> <b>Liste des erreurs</b> </div> <ul style="list-style-type: none"> <li>connexion : Erreur dans la méthode modele\dao\RestoDAO::getAimesByIdU : &lt;br&gt;SQLSTATE[HY000]: General error: 2014 Cannot execute queries while there are pending result sets. Consider unsetting the previous PDOStatement or calling PDOStatement::closeCursor()</li> </ul> <p>Mais, ensuite, on peut constater que l'attaque a réussi, car <b>les photos des restaurants ne sont plus affichées</b> sur la page d'accueil (ni ailleurs) : les données de la table photo ont été supprimées !</p> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <b>AVANT :</b>   </div> <div style="text-align: center;"> <b>APRÈS :</b>   </div> </div> <p>On souhaite donc rendre impossibles les attaques par injection SQL sur ce formulaire.</p>	
<b>Solution</b> :  Il faut préparer la requête avant de l'exécuter pour éviter au maximum les injection SQL. Puis après une vérification avant et après on voit que l'injection ne fonctionne plus. <pre> \$requete = "SELECT * FROM utilisateur WHERE mailU = ':mailU'"; \$stmt = Bdd::connecter()-&gt;prepare(\$requete); \$stmt-&gt;bindParam(':mailU', \$mailU); \$stmt-&gt;execute(); </pre>	

