





Ticket n°10

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

Type du ticket : incident <i>(évolution/incident)</i>	Niveau de gravité : <input type="checkbox"/> Bloquant <input checked="" type="checkbox"/> Majeur <input type="checkbox"/> Mineur
Émetteur : Nicolas BOURGEOIS <i>(nom de l'émetteur)</i>	Date signalement : 21/09/2024 <i>(jj/mm/aaaa)</i>
Assignation : Stefen <i>(nom du membre de l'équipe en charge du ticket)</i>	Date de résolution souhaitée : 14/10/2024 <i>(jj/mm/aaaa)</i>
Application concernée : R3st0.frVersion : 1.0 initiale – septembre 2024	
<div>Description du problème <i>(avec éventuelles captures d'écran, messages d'erreurs)</i> :</div> <div>On m’a signalé qu’une attaque par injection SQL est possible sur la page de connexion.</div> <div>Scénario :</div> <div>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</div> <div>et une valeur quelconque dans le mot de passe.</div> <div>L'application refuse l'authentification en affichant le message d'erreur suivant :</div> <div>Liste des erreurs</div> <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></div> <div>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 !</div> <div>AVANT :</div> <div><div>Top 4 des meilleurs restaurants</div><div><div><div><a href="#">l'entresolée</a> 2 rue Maurice Ravel 33000 Bordeaux</div></div><div><div><a href="#">Cidrerie du fronton</a> Place du Fronton 64210 Arbonne</div></div><div><div><a href="#">le bar du charcutier</a> 30 rue Parlement Sainte-Catherine 33000 Bordeaux</div></div><div><div><a href="#">la petite auberge</a> 15 rue des cordeliers 64100 Bayonne</div></div></div></div> <div>APRÈS :</div> <div><div>Top 4 des meilleurs restaurants</div><div><div><div><a href="#">l'entresolée</a> 2 rue Maurice Ravel 33000 Bordeaux</div><div><a href="#">Cidrerie du fronton</a> Place du Fronton 64210 Arbonne</div><div><a href="#">le bar du charcutier</a> 30 rue Parlement Sainte-Catherine 33000 Bordeaux</div><div><a href="#">la petite auberge</a> 15 rue des cordeliers 64100 Bayonne</div></div><div><small>*Classement basé sur les notations des non-utilisateurs</small></div></div></div> <div>On souhaite donc rendre impossibles les attaques par injection SQL sur ce formulaire.</div>	
Solution <i>(diagnostic, localisation, modification, test)</i> :	
<div>La méthode qui permet l'authentification est défini dans le fichier authentication.inc.php. Elle utilise la fonction getOneByEmail() pour savoir si l'utilisateur est connu de la BDD ; c'est cette fonction qui nous intéresse.</div> <div>Pour empêcher les injections SQL, il faut utiliser des requêtes préparées pour séparer le requête SQL et les données utilisateur.</div>	

```
public static function getOneByMail(string $mailU): ?Utilisateur {
    $leUser = null;
    try {
        $requete = "SELECT * FROM utilisateur WHERE mailU = :mailU";
        $stmt = Bdd::getConnexion()->prepare($requete);
        $stmt->bindParam(':mailU', $mailU, PDO::PARAM_STR);
        $stmt->execute();

        // Si au moins un (et un seul) utilisateur (car login est unique), c'est que le mail existe dans la BDD
        if ($stmt->rowCount() > 0) {
            $enreg = $stmt->fetch(PDO::FETCH_ASSOC);
            $idU = $enreg['idU'];
            $lesRestosAimes = RestoDAO::getAimesByIdU($idU);

            $leUser = new Utilisateur($idU, $enreg['mailU'], $enreg['mdpU'], $enreg['pseudoU']);
            $leUser->setLesRestosAimes($lesRestosAimes);
        }
    } catch (PDOException $e) {
        throw new Exception("Erreur dans la méthode " . get_called_class() . "::getOneByMail : <br/>" . $e->getMessage());
    }
    return $leUser ;
}
```

Un nouveau message d’erreur apparaît stipulant cette fois une erreur lors de la connexion.

Liste des erreurs

- Connexion : erreur de login ou de mot de passe

Connexion

Email de connexion

Mot de passe

Soumettre

[Inscription](#)

Et les images sont toujours présentes.

