Exercice post formation

# Contenu de l’exercice

A l’issue de la formation à la programmation défensive et des ateliers, vous êtes normalement en mesure de détecter une bonne partie des failles dans une application qui en contiendrait. L’application BadSecApp a justement été conçue pour cela, et elle fourmille de vulnérabilités.

L’exercice qu’il vous est demandé de réaliser, en individuel ou en équipe selon vos préférences, et qui nous permettra d’évaluer l’impact de la formation, consiste à partir de l’application BadSecApp et proposer (au maximum) :

1. Une localisation des failles
2. Une analyse de ces failles
   1. De quel type sont-elles dans la catégorie OWASP Top 10 ?
   2. Quel niveau de risque leur associez-vous ?
   3. Comment pourraient-elles être exploitées par un attaquant ?
3. Une résolution technique de ces failles (mitigation par une mesure de réduction de la faille ou bien correction directe de la faille dans le code)

# Informations pratiques

L’application BadSecApp se trouve sur <https://github.com/jp-gouigoux/BadSecApp>.

Trois branches sont disponibles, en fonction de l’estimation de votre propre niveau :

Une image contenant texte

Description générée automatiquement

Dans la branche **main**, les failles sont indiquées par des tags SECU dans le code et leur catégorie est détaillée. La première partie du travail est donc déjà réalisée pour vous. En fait, c’est même la totalité du travail qui est faite pour vous avec toutes les modifications de code pour servir de correction à l’exercice. Du coup, pour démarrer au bon endroit, il faut utiliser l’endroit où le merge a été réalisé, c’est-à-dire <https://github.com/jp-gouigoux/BadSecApp/tree/27b2872c6dbaaa71823c0a05964ecc78ca13c492>

La branche **indicateurs** vous laisse un petit peu plus de travail, dans le sens où les textes SECU vous indiquent toujours où sont les problèmes, mais il n’y a rien d’expliqué dessus et c’est à vous de décrire la vulnérabilité, de l’analyser, etc.

La branche **experts** est la plus proche de la réalité : le code vous est fourni tel quel, sans aucune indication d’où sont les problèmes, et vous devez donc réaliser le travail complet pour chaque vulnérabilité que vous trouverez… et bien sûr en localiser le plus possible !

# Rendu de l’exercice

Toute votre analyse et les corrections doivent être réalisées directement dans le code et le rendu sera fait sous cette forme au correcteur. Afin de ne pas surcharger le contenu avec les binaires, commencez par lancer la commande suivante :

Une image contenant texte

Description générée automatiquement

Puis compressez toute la solution dans une archive ZIP intitulée avec votre adresse mail et envoyez-là à [jp.gouigoux@salviadeveloppement.com](mailto:jp.gouigoux@salviadeveloppement.com), pour le 15 décembre 2022 au plus tard. Précisez dans le mail à partir de quelle branche vous êtes parti, même si normalement ceci se voit dans le code. Le correcteur en attendra bien sûr plus de votre proposition si vous êtes partis des vulnérabilités marquées dans le code que si vous êtes partis de la version experts…

Les corrections seront normalement réalisées sur la fin de l’année et les résultats communiqués sur le début 2023. Ils vous seront envoyés individuellement, et si certains réalisent un travail qui sort de l’ordinaire, des prix seront peut-être affectés 😊.

Si vous avez déjà réalisé l’exercice l’an prochain, deux cas de figure :

* Vous n’étiez pas arrivé au bout (malheureusement le cas de beaucoup de rendus, par manque de temps ou peut-être manque de formation) 🡺 c’est le moment de creuser encore plus et de faire un pas supplémentaire, en repartant de votre travail de l’an passé. Pensez à marquer clairement dans le code vos modifications et analyses de 2022, de façon que le correcteur n’ait pas à regarder les exercices de l’an passé et perdre du temps à comparer les codes.
* Vous aviez couvert une bonne partie de l’exercice et aviez reçu une bonne évaluation 🡺 pas la peine de recommencer, car le contenu de l’application de test n’a pas évolué cette année.