|  |
| --- |
| Rapport de pentest |
| Vulnerability Report |
| Sunday, June 30, 2024 |

Table of Contents

[1. Introduction 3](#_Toc177396537)

[1.1. Contexte et objectifs de la mission 3](#_Toc177396538)

[1.2. Périmètre de la mission 3](#_Toc177396539)

[1.3. Méthodologie 3](#_Toc177396540)

[1.4. Déroulement des travaux 3](#_Toc177396541)

[1.5. Limites 4](#_Toc177396542)

[1.6. Remise en état 4](#_Toc177396543)

[2. Summary 5](#_Toc177396544)

[2.1. Executive summary 5](#_Toc177396545)

[2.1. Vulnérabilités et recommandations 6](#_Toc177396546)

[3. Constats détaillés 7](#_Toc177396547)

# Introduction

## Contexte et objectifs de la mission

IBC ask to carry out undefined. The objectives of these tests were to:

* Identify vulnerabilities in the web application and the mobile application (iOS for iPad);
* Identify the risks associated with the identified vulnerabilities;
* To propose prioritized corrective actions;
* Formalize these elements in a clear and understandable report for technical and business contacts.

## Périmètre de la mission

IBC has mandated us to perform security tests on the following scope:

* www.secure-bank.com
* www.sudo.co.

## Méthodologie

The engagement was based the following approaches:

* Phase 1: Information gathering & target mapping;
* Phase 2: Vulnerability scan & attack plan identification;
* Phase 3: Exploitation of identified vulnerabilities & elevation of privileges;
* Phase 4: Consolidation & results restitution.

## Déroulement des travaux

The testing activities were performed between 06/18/2024 and 07/02/2024.

## Limites

## Remise en état

# Summary

## Executive summary = synthese manageriale

IBC a mandaté Corellia afin d’effectuer des tests d’intrusion sur ses applications web «sudo.co.il», «securebank.com». Le but de ces tests était de s’assurer que ces applications sont protégées contre des cyberattaques.

Lors de nos tests d’intrusions, nous avons identifié deux vulnérabilités de gravité très élevée (V01 & V02), qui pourrait permettre à un attaquant opérant depuis Internet d’usurper l’identité de n’importe quel utilisateur des applications web sudo.co.il et securebank.com, et ce, sans authentification. Cela lui permettrait ainsi d’accéder aux informations du compte et d’effectuer toutes les actions disponibles à partir de celui-ci.

Nous avons également identifié une vulnérabilité de gravité modérée. La vulnérabilité (V03) pourrait permettre à un attaquant d’obtenir les identifiants de connexion des utilisateurs du serveurs hébergeant l’application securebank.com. Cela pourrait permettre à l’attaquant de prendre le contrôle à distance du serveur.

Les risques métiers associées à ces différentes vulnérabilités sont les suivants :

· Mise en cause par rapport aux réglementations de la protection des données personnelles, telles que le RGPD : un attaquant pourrait parvenir à récupérer des données personnelles (données personnelles, transactions bancaires, factures, RIB, numéro de sécurité sociale) des utilisateurs de securebank.com, ce qui exposerait IBC à un risque de sanction par les organismes de régulation.

· Fraude / Usurpation d'identité : un attaquant pourrait exploiter diverses vulnérabilités afin d’usurper l’identité d’un utilisateur et accéder à ses données personnelles stockées sur l’application securebank.com.

· Contournement de contrôle interne : un attaquant pourrait obtenir des accès non autorisés à des ressources de securebank.com.

· Aide à l’exploitation d’autres vulnérabilités.

Corellia estime que les vulnérabilités identifiées peuvent constituer une menace grave pour le système d’information de IBC. De ce fait, Corellia préconise de corriger au plus vite les vulnérabilités en appliquant les recommandations associées.

## Vulnérabilités et recommandations

Following vulnerabilities have been discovered:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Vulnérabilités | Gravité | Recommandations | Priorité |
| V007 | SQL Injection | Critique | Protéger l'application contre les attaques de type injection SQL | Forte |
| V003 | XSS | Forte | Nous recommandons de proteger les entrées utilisateurs. | Forte |
| V008 | Relative Path Traversal | Forte | Utilisez une stratégie de validation des entrées | Moyenne |

# Constats détaillés

The following details the vulnerabilities identified in the scope, together with an estimate of the impact of the exploitation of the vulnerability, the ease of exploitation of the vulnerability and its criticality. Also referenced are the business risks associated with the vulnerability in the context of your environment, as well as the recommendation for remediating the vulnerability and reducing the risks.

|  |  |  |
| --- | --- | --- |
| V007 | SQL Injection | Gravité |
| Critique |
| Libellé | L'injection de code SQL (SQLi) permet de modifier la logique des instructions SQL  L'application web securebank dispose d'un champ de connexion a l'url www.securebank.com. Ce formulaire de connexion est vulnerable a l'injection de code SQL.  Ce type de faille survient lorsque le site web utilise les données saisies par l'utilisateur pour effectuer des actions de lecture, écriture, modification ou suppression dans la base de données, sans effectuer de contrôle sur le contenu des données soumises par l'utilisateur. Un attaquant pourrait alors soumettre des requêtes contenant des mots-clés ayant un sens au niveau du langage de requête de la base de données. Il pourrait ainsi modifier le comportement des requêtes effectuées dans la base de données par l'application web ciblée et accéder notamment à la structure et aux contenus des tables de la base. Il pourrait également, dans certains cas, impacter la disponibilité ou la performance de la base de données en lançant des commandes de type BENCHMARK. | |
| Périmètre |  | |

|  |  |  |
| --- | --- | --- |
| Recommandation | | |
| E007 | Protéger l'application contre les attaques de type injection SQL | Priorité |
| 4 |
| Libellé | Protéger l'application contre les attaques de type injection SQL Nous recommandons de corriger l'application securebank afin qu'elle soit protégée contre les attaques de type injection SQL. Pour ce faire, nous recommandons de modifier le code de l'application securebank afin d'utiliser systématiquement des requêtes préparées ("prepared statements") bindées lorsque des paramètres sous le contrôle de l'utilisateur sont utilisés dans la construction des requêtes SQL.  À défaut (et cela doit rester une exception), nous recommandons de filtrer en mode "liste blanche" les données sous le contrôle de l'utilisateur avant leur insertion dans la requête SQL. Dans cette approche, seuls les caractères ou chaînes de caractères explicitement définis sont autorisés. La présence de caractères non autorisés doit entraîner l'échec de la requête. | |

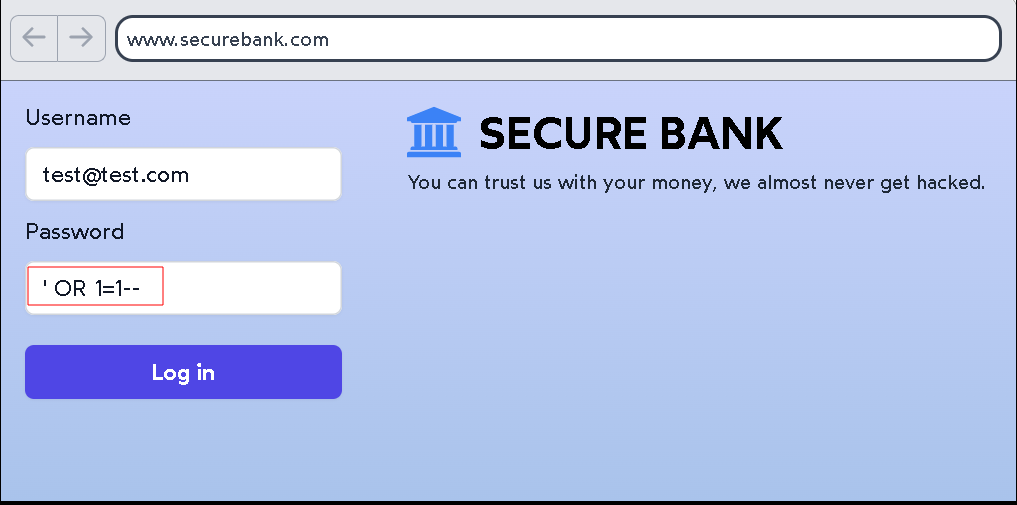


Image **1** – Injection de code Sql dans le champ de connexion

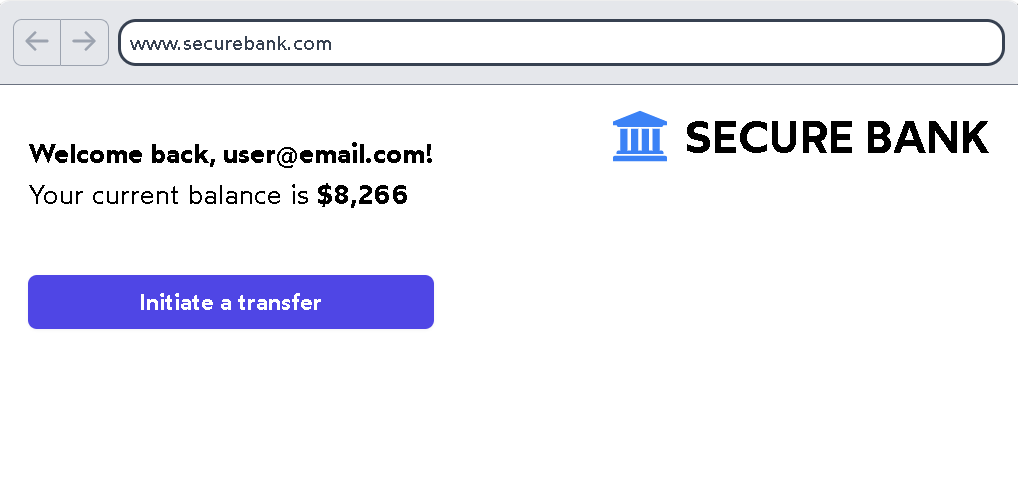


Image **2** – Execution de code Sql reussie

|  |  |  |
| --- | --- | --- |
| V003 | XSS | Gravité |
| Forte |
| Libellé | La vulnérabilité « Cross-Site Scripting (XSS) » permet à des tiers d'injecter du code Java Script ou HTML dans des pages Web d'applications spécifiques.  L'application web sudo.co dispose d'une fonctionnalite d'abonnement a la nexsletter d'IBC a l'url sudo.co/abonnement. Cette fonctionnalite est vulnerable a l'injection de code xss.  Un attaquant pourrai l'exploiter pour voler des sessions utilisateur, de compromettre la sécurité du système et d'obtenir des informations sensibles ou exclusives de l'application. | |
| Périmètre |  | |

|  |  |  |
| --- | --- | --- |
| Recommandation | | |
| E003 | Nous recommandons de proteger les entrées utilisateurs. | Priorité |
| 4 |
| Libellé | Nous recommandons de proteger les entrées utilisateurs. Chaque paramètre géré par l'application doit être validé, notamment lorsqu'il s'agit de données d'entrée fournies par l'utilisateur. Cette validation est toujours effectuée côté serveur et ne repose JAMAIS sur des validations effectuées côté client. | |

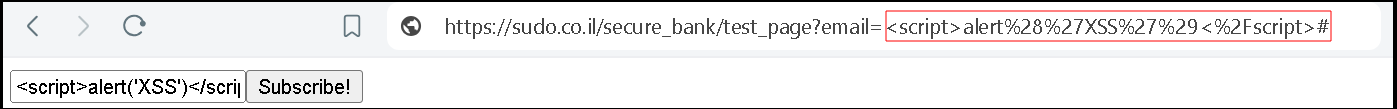


Image **3** – Injection d'un code javascript dans un champ d'abonnement

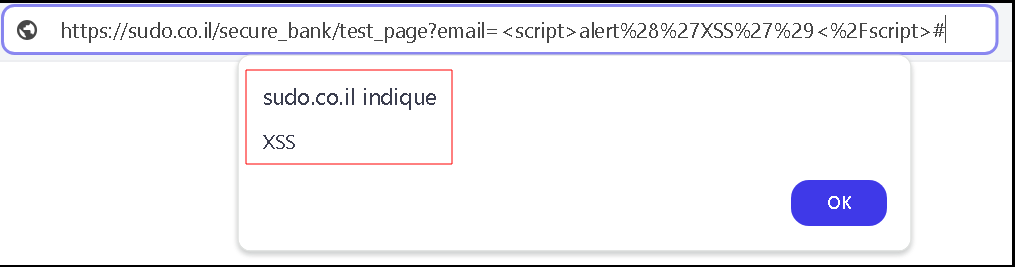


Image **4** – Execution du code Xss reussie

|  |  |  |
| --- | --- | --- |
| V008 | Relative Path Traversal | Gravité |
| Forte |
| Libellé | La vulnérabilité Path Traversal permet à des tiers d'acceder a des fichier depuis l'url d'un site web.  L'application web securebank est vulnérable à une faille de type Local File Inclusion (LFI). En effet, il est possible de spécifier un argument dans une requête pour determiner un chemin relatif de fichier (en utilisant la notation "../.." qui permet de sortir de la racine du site web au niveau du système de fichiers). La requête met en œuvre un mécanisme qui affiche le contenu du fichier dont le nom est passé en paramètre.  Un attaquant opérant depuis Internet pourrait ainsi utiliser cette fonctionnalité pour afficher le contenu de n'importe quel fichier texte présent sur le serveur et accessible en lecture au processus exécutant le serveur web. | |
| Périmètre |  | |

|  |  |  |
| --- | --- | --- |
| Recommandation | | |
| E008 | Utilisez une stratégie de validation des entrées | Priorité |
| 4 |
| Libellé | Utilisez une stratégie de validation des entrées « accepter les entrées connues comme bonnes », c'est-à-dire utilisez une liste blanche d'entrées acceptables strictement conformes aux spécifications. Rejetez toute entrée qui n'est pas strictement conforme aux spécifications ou transformez-la en quelque chose qui l'est. Lors de la validation des entrées, tenez compte de toutes les propriétés potentiellement pertinentes, notamment la longueur, le type d'entrée, la plage complète de valeurs acceptables, les entrées manquantes ou supplémentaires, la syntaxe, la cohérence entre les champs associés et la conformité aux règles métier. À titre d'exemple de logique de règle métier, « boat » peut être syntaxiquement valide car il ne contient que des caractères alphanumériques, mais il n'est pas valide si l'entrée ne doit contenir que des couleurs telles que « rouge » ou « bleu ». Ne vous fiez pas exclusivement à la recherche d'entrées malveillantes ou malformées (c'est-à-dire, ne vous fiez pas à une liste noire). Une liste noire est susceptible de manquer au moins une entrée indésirable, en particulier si l'environnement du code change. Cela peut donner aux attaquants suffisamment de marge pour contourner la validation prévue. Cependant, les listes noires peuvent être utiles pour détecter des attaques potentielles ou déterminer quelles entrées sont si mal formées qu'elles doivent être rejetées d'emblée. Lors de la validation des noms de fichiers, utilisez des listes blanches strictes qui limitent le jeu de caractères à utiliser. Si possible, n'autorisez qu'un seul caractère « . » dans le nom de fichier pour éviter les faiblesses telles que CWE-23, et excluez les séparateurs de répertoire tels que « / » pour éviter CWE-36. Utilisez une liste blanche d'extensions de fichiers autorisées, ce qui permettra d'éviter CWE-434. Ne vous fiez pas exclusivement à un mécanisme de filtrage qui supprime les caractères potentiellement dangereux. Cela équivaut à une liste noire, qui peut être incomplète (CWE-184). Par exemple, le filtrage de « / » constitue une protection insuffisante si le système de fichiers prend également en charge l'utilisation de « » comme séparateur de répertoire. Une autre erreur possible peut se produire lorsque le filtrage est appliqué d'une manière qui produit toujours des données dangereuses (CWE-182). Par exemple, si les séquences '../' sont supprimées de la chaîne '.../...//' de manière séquentielle, deux instances de '../' seraient supprimées de la chaîne d'origine, mais les caractères restants formeraient toujours la chaîne '../'. | |

image

Image **5** – Recuperation du fichier /etc/shadow



Image **6** – Fichier shadow récupéré

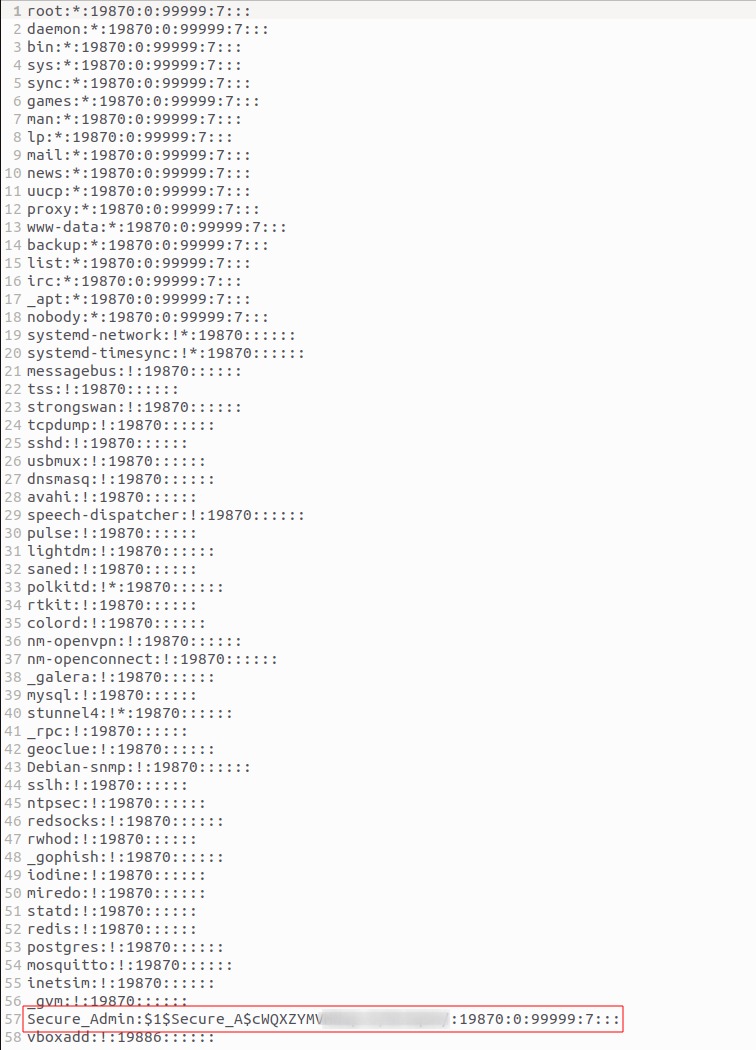


Image **7** – utilisateur Secure\_Admin et son hash