Test d'intrusion

UnSAFE Bank

Vulnerable Banking Suite

Sommaire

Introduction

1. Contexte et expression du besoin

Contexte

UnSAFE Bank est une société spécialisée dans les services bancaires en ligne, proposant une suite d'applications virtuelles accessibles via une interface web et des applications mobiles (Android et iOS). Les principales fonctionnalités incluent le transfert de fonds, la gestion des bénéficiaires, la consultation des relevés de compte, la demande de prêts, et la gestion des paramètres utilisateur. Ces services visent à fournir une expérience bancaire fluide et efficace à leurs clients.

En tant qu'entreprise du secteur financier, UnSAFE Bank héberge un volume important de données sensibles, y compris des informations personnelles de ses clients et des détails financiers critiques. Afin de préserver la confiance de ses utilisateurs et de garantir la sécurité des données collectées, UnSAFE Bank s'engage à respecter des normes strictes de sécurité et de confidentialité.

Dans ce contexte, la société souhaite s'assurer que l'application web répond aux exigences en matière de cybersécurité et est protégée contre les cybermenaces potentielles. Elle a mandaté **Oteria-Pentest** pour réaliser un audit de sécurité approfondi, visant à identifier les vulnérabilités présentes dans le système et à proposer des recommandations pour leur correction.

Cet audit est mené dans un **contexte de boîte noire**, où les auditeurs ne disposent que d'une information initiale : l'adresse IP **192.168.10.20**. L'objectif est de simuler une attaque externe afin de révéler les failles de sécurité et d'en évaluer l'impact.

Dans la suite du document, les termes suivants seront utilisés :

- La société **UnSAFE Bank** sera désignée comme **« le client »**.
- Les pentesters **Elise Chabeniuk**, **Florian Mora** et **Josselin Menguy** de la société **Oteria-Pentest** seront désignés comme **« les auditeurs »**.

Expression du Besoin

L'objectif principal de cet audit est d'identifier, analyser, et documenter les vulnérabilités présentes dans l'application web et ses services associés. Le client souhaite obtenir un rapport détaillé permettant d'évaluer les risques liés aux failles détectées et de disposer de recommandations pour leur correction.

Les besoins spécifiques incluent :

1. **Identification des vulnérabilités critiques** :
   Failles exploitables via des attaques courantes (injections, authentification, gestion de sessions, etc.).
   Exposition d'informations sensibles ou compromission de données clients.
2. **Évaluation de l'impact des vulnérabilités** :
   Risques opérationnels : perturbation des services en ligne.
   Risques financiers : pertes monétaires ou fraude.
   Risques réglementaires : non-conformité avec des normes comme le RGPD.
3. **Simulation d'un scénario d'attaque réaliste** :
   Exploitation des services détectés sur l'adresse IP fournie.
   Exploration des API ou terminaux exposés pour identifier des failles potentielles.
4. **Fourniture de recommandations stratégiques et techniques** :
   Mesures correctives spécifiques pour chaque faille identifiée.
   Bonnes pratiques et recommandations pour limiter l'exposition aux menaces futures.

Les résultats attendus comprennent une évaluation des vulnérabilités classées par criticité, un plan d'action hiérarchisé, et des solutions concrètes pour renforcer la sécurité de l'application et de son infrastructure.

Cet audit s'inscrit dans une démarche proactive visant à protéger les actifs critiques du client tout en renforçant la confiance des utilisateurs dans les services d'UnSAFE Bank.

2. Déroulement de l'audit

Phase 1 – Recherche d'informations / Prise d'empreinte

Le but est de rechercher de l'information sur l'organisation et ses membres. Des logiciels dits OSINT (open source intelligence) permettent de récupérer de l'information avec un cheminement précis. Elle permet aussi de récupérer tous éléments liés à l'IP qui nous a été transmise afin par la suite de trouver des angles d'attaques.

Phase 2 - Analyse des vulnérabilités

Cette phase a pour objectif la recherche de diverses brèches sur le réseau et l'application web et de classifier le degré de criticité des vulnérabilités trouvées.

Phase 3 – Test Applicatifs

En adéquation avec ce qui a été trouvé dans la phase précédente, le testeur tente de pénétrer le serveur web par le biais de différentes attaques comme des injections.

Phase 4 - Rapport et recommandations

La phase de « reporting » consiste à rassembler les résultats des différents tests et de consolider le rapport autour d'une étude des risques applicatifs. Elle amène aussi à des recommandations sur les risques rencontrés lors du pentest.

Phase 5 - Remédiation

Cette phase est gérée par le client après la remise du rapport et le compte-rendu oral. Elle lui permet de remédier aux vulnérabilités et failles trouvées grâce aux recommandations données pas les auditeurs.

3. Présentation de l'échelle utilisée

L'échelle de **Risque** est classée selon 4 niveaux :

Niveau de dangerosité	Description
Faible	Élément présentant un danger mineur quant à la sécurité du système d'information. La correction de celui-ci n'est pas obligatoire mais conseillée
Moyenne	Élément pouvant exposer publiquement des informations non souhaitables permettant à un attaquant d'obtenir des informations sur l'entreprise. La correction de cet élément est fortement conseillée.
Major	Élément mettant potentiellement en danger le système d'information. La correction de cet élément est obligatoire pour revenir à un niveau de sécurité acceptable.
Critical	Élément mettant en péril le système d'information où des données clients. La correction de celui-ci est extrêmement urgente et obligatoire.

L'échelle de **l'impact** est classée selon 4 niveaux :

Niveau de dangerosité	Description
Mineur	Aucun impact opérationnel ni sur les performances de l'activité ni sur la sécurité des personnes et des biens. La société surmontera la situation sans trop de difficultés (consommation des marges).
Significative	Dégradation des performances de l'activité sans impact sur la sécurité des personnes et des biens. La société surmontera la situation malgré quelques difficultés (fonctionnement en mode dégradé).
Major	Forte dégradation des performances de l'activité, avec d'éventuels impacts significatifs sur la sécurité des personnes et des biens. La société surmontera la situation avec de sérieuses difficultés (fonctionnement en mode très dégradé).
Critical	Incapacité pour la société d'assurer tout ou partie de son activité, avec d'éventuels impacts graves sur la sécurité des personnes et des biens. La société ne surmontera vraisemblablement pas la situation (sa survie est menacée).

L'échelle de **vraisemblance** est classée selon 4 niveaux :

Niveau de dangerosité	Description
Négligeable	I ne semble pas possible que les sources de risques retenues puissent réaliser la menace en s'appuyant sur les caractéristiques des supports (ex. : vol de supports papiers stockés dans un local de l'organisme dont l'accès est contrôlé par badge et code d'accès).
Limité	il semble difficile pour les sources de risques retenues de réaliser la menace en s'appuyant sur les caractéristiques des supports (ex. : vol de supports papiers stockés dans un local de l'organisme dont l'accès est contrôlé par badge)
Medium	Forte dégradation des performances de l'activité, avec d'éventuels impacts significatifs sur la sécurité des personnes et des biens. La société surmontera la situation avec de sérieuses difficultés (fonctionnement en mode très dégradé).
Critical	Incapacité pour la société d'assurer tout ou partie de son activité, avec d'éventuels impacts graves sur la sécurité des personnes et des biens. La société ne surmontera vraisemblablement pas la situation (sa survie est menacée).

Audit boîte noire de UnSAFE Bank

1. Prise d'empreinte

Avant de lancer la phase de tests, une étape de récupération d'informations sur le site a été effectuée.

Récupération des informations du site correspondant à l'IP fournie

```
florian@florian-G3-3590:~$ curl -I http://192.168.10.20/
HTTP/1.1 404 Not Found
Date: Wed, 30 Oct 2024 09:39:56 GMT
Server: Apache/2.4.33 (Unix)
X-Powered-By: PHP/7.2.7
Access-Control-Allow-Origin: *
Access-Control-Allow-Headers: *
Access-Control-Allow-Methods: *
Access-Control-Allow-Credentials: true
Content-Type: text/html; charset=UTF-8
```

Utilisation de l'outil Nmap afin de détecter des ports ouverts

```
florian@florian-G3-3590:~$ nmap  192.168.10.20
Starting Nmap 7.80 ( https://nmap.org ) at 2024-10-30 10:47 CET
Nmap scan report for 192.168.10.20
Host is up (0.053s latency).
Not shown: 996 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
80/tcp   open  http
199/tcp  open  smux
3000/tcp open  ppp
```

Récupération des versions des services trouvés précédemment avec NMap

```
nmap -sS -p- -A 192.168.10.20

PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 9.2p1 Debian 2+deb12u3 (protocol 2.0)
80/tcp   open  http    Apache httpd 2.4.33 ((Unix))
|_http-server-header: Apache/2.4.33 (Unix)
|_http-title: Site doesn't have a title (text/html; charset=UTF-8).
199/tcp  open  smux    Linux SNMP multiplexer
3000/tcp open  http    nginx 1.27.2
|_http-server-header: nginx/1.27.2
|_http-title: UnSAFE Bank
```

Grâce à ces informations, nous relevons quelques recommandations et bons points :

- Mettre à jour le serveur Apache 2.4.33 à la dernière version sortie (2.4.62) afin de remédier à de nombreuses vulnérabilités qui touchent les versions antérieures à la nouvelle.

- Mettre à jour le serveur PHP 7.2.7 à la dernière version sortie (8.4.1) afin de remédier à de nombreuses vulnérabilités qui touchent les versions antérieures à la nouvelle.

- Restreindre les en-têtes car elles donnent trop d'autorisations.
    - L'en-tête **Access-Control-Allow-Origin: *** autorise n'importe quel site à accéder aux API ou ressources, même si ces ressources contiennent des données sensibles.
    - L'en-tête **Access-Control-Allow-Headers: *** autorise n'importe quel en-tête personnalisé, ce qui peut faciliter des attaques en manipulant les requêtes (comme les en-têtes d'autorisation).
    - L'en-tête **Access-Control-Allow-Methods: *** autorise toutes les méthodes (GET, POST, PUT, etc.), un attaquant peut donc exploiter des méthodes non sécurisées pour manipuler ou supprimer des données.

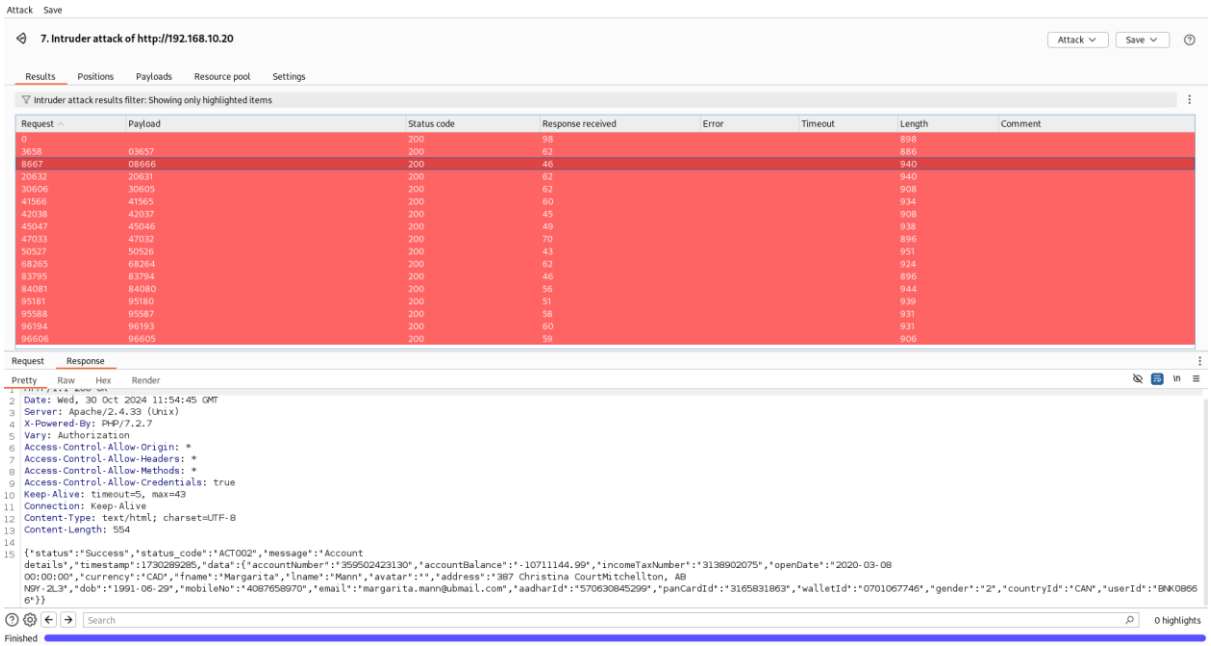
- Version de Nginx à jour ce qui assure très peu de vulnérabilités même s'il faut rester vigilant.

2. Détermination des vulnérabilités à l'aide des outils automatiques

a. Récupération des informations des comptes avec BURP (IDOR) - ID01

Description de la vulnérabilité

Cette vulnérabilité de type IDOR permet de récupérer l'ensemble des informations de tous les comptes existants. Il suffit d'intercepter la requête et d'énumérer tous les ID utilisateur (userid) et de récupérer les réponses HTTP. L'IDOR (Insecure Direct Object Reference) est une vulnérabilité de sécurité qui permet à un utilisateur d'accéder à des ressources ou données auxquelles il ne devrait pas avoir accès en modifiant simplement un identifiant dans une requête.



Étapes pour reproduire la vulnérabilité

1. **Utilisation de Burp Suite**
   - Configurer Burp Suite pour intercepter les requêtes HTTP.
   - Maintenir la connexion avec l'option keep-alive.
2. **Lancer un Intruder**
   - Configurer un Intruder dans Burp Suite avec la page d'informations de compte.
   - Utiliser une liste de nombres allant de 00000 à 99999 pour énumérer tous les comptes existants.

Impact

Cette vulnérabilité permet à un attaquant d'avoir l'ensemble des informations des utilisateurs (nom, adresse, numéro de téléphone, montant etc...) et de changer les mots de passe de tous les comptes utilisateurs, compromettant ainsi la sécurité de l'application et la confidentialité des données et cela sans droits administrateur.

Exploitability	Impact	Risk
Medium 2.3	Critical 6	Critical

Recommandations

- **Contrôle d'accès côté serveur** : Avant de renvoyer les informations de profil, le serveur doit vérifier si l'utilisateur authentifié a les droits d'accès aux informations demandées.

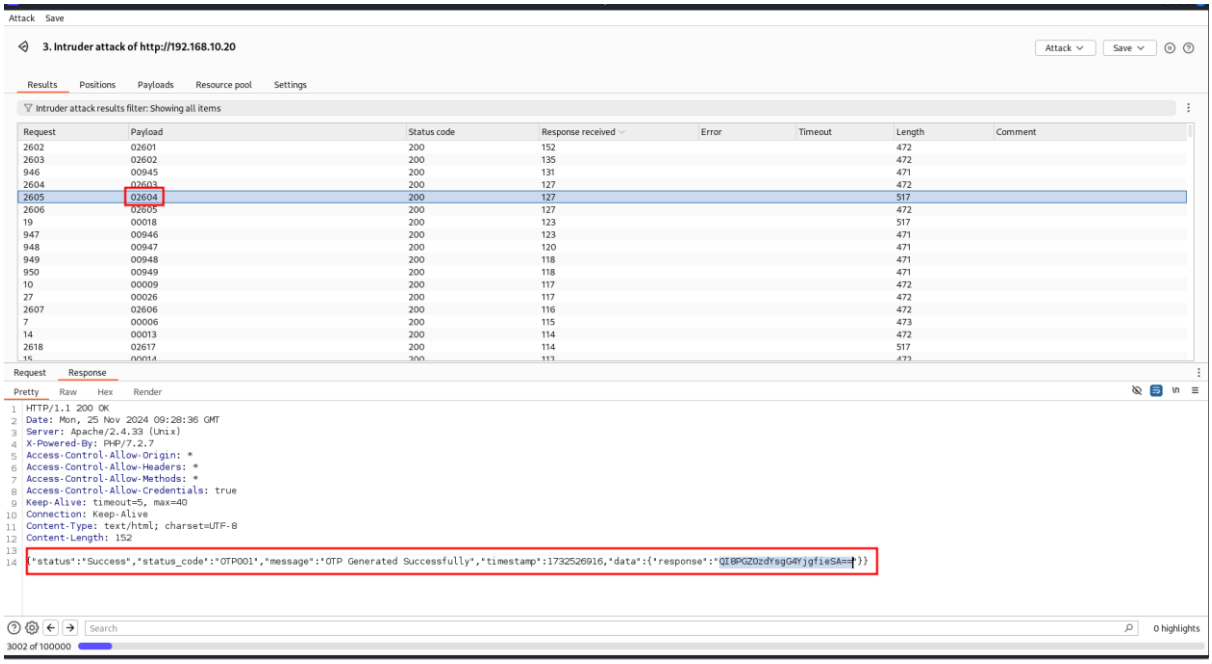
- **Utilisation de jetons sécurisés** : Au lieu d'IDs manipulables comme customer_id, utiliser des identifiants sécurisés (comme des UUID ou des jetons) pour réduire les possibilités d'accès non autorisé.

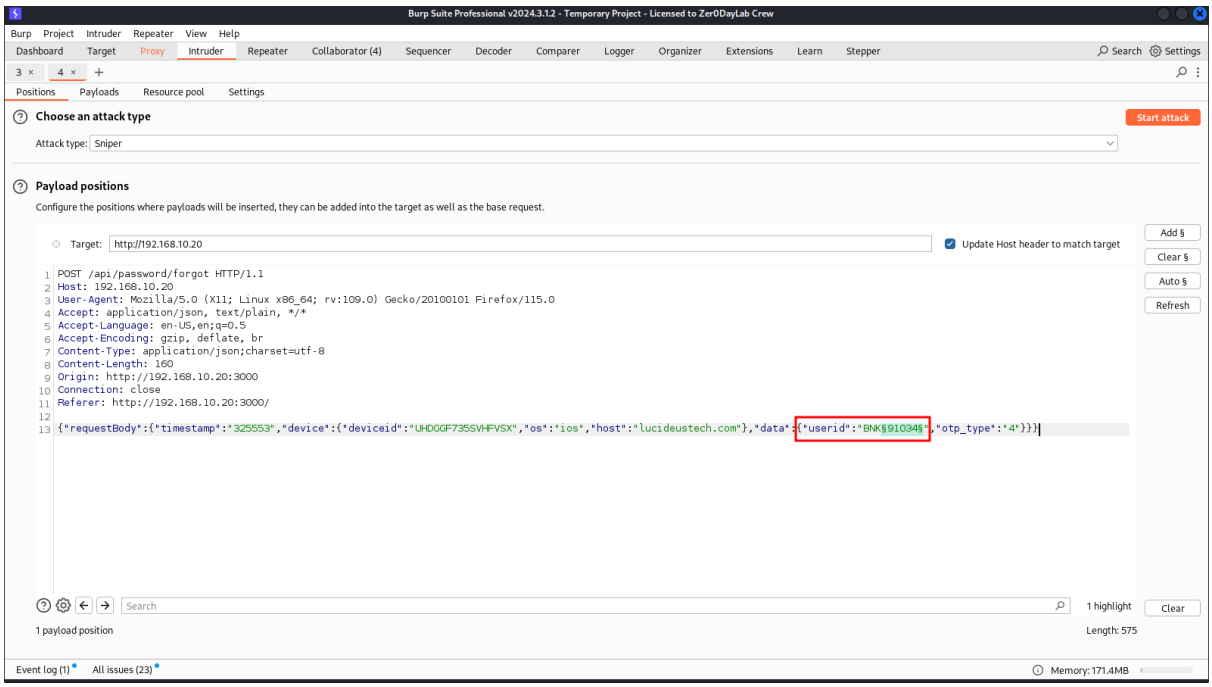
- **Contrôles d'accès rigoureux** : L'implémentation d'un contrôle d'accès global pour l'application peut bloquer l'accès aux données d'autres utilisateurs en vérifiant les permissions à chaque requête.

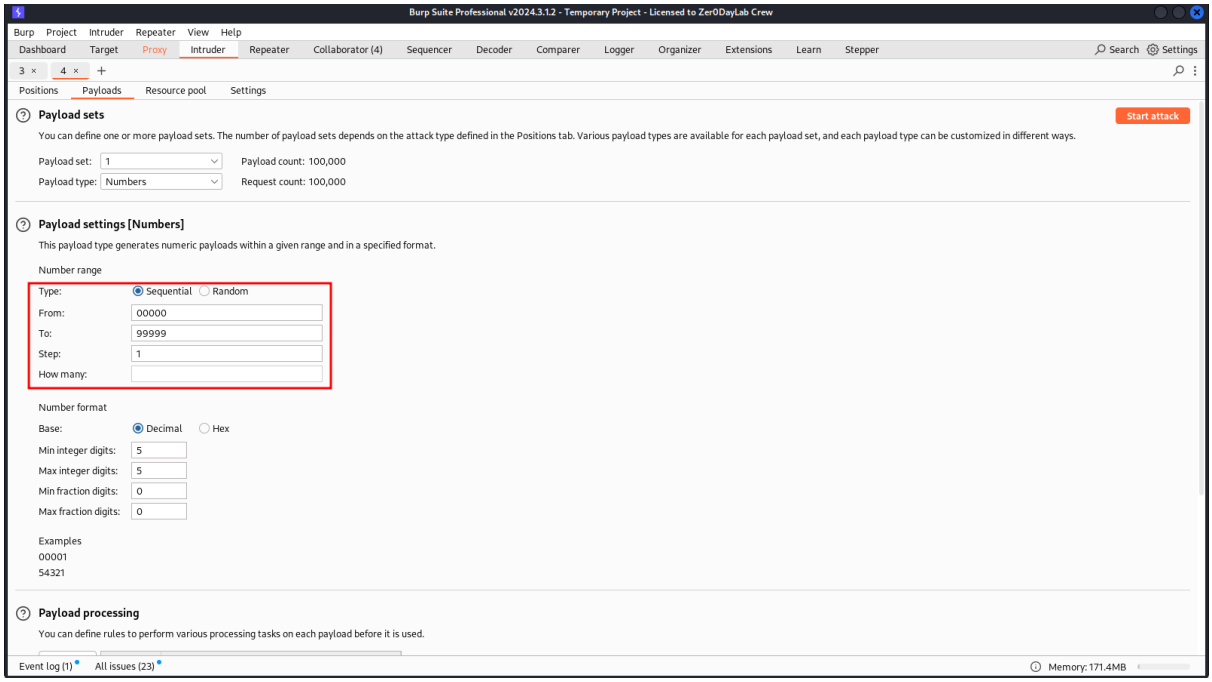
b. Enumération des comptes et changement des mots de passe avec BURP – ID02

Description de la vulnérabilité

Ce rapport illustre une vulnérabilité détectée sur la page "Mot de Passe Oublié" du site cible, qui permet l'énumération des comptes des utilisateurs. Cette vulnérabilité expose les utilisateurs à des attaques potentielles, tel que le détournement de compte.







Étapes pour reproduire la vulnérabilité

1. **Point d'Entrée** : La page "Mot de passe oublié" est utilisée pour cette attaque.

2. **Attaque** : L'attaque consiste à utiliser un outil de test d'intrusion (comme Burp Suite) pour envoyer des requêtes HTTP à la page en question. Les requêtes utilisent des numéros d'identification d'utilisateur incrémentaux ou prédictifs pour tester l'existence des utilisateurs.

3. **Réponse du Serveur** : Le serveur renvoie différentes réponses en fonction de l'existence ou non du compte, ce qui permet la déduction de comptes valides.

Impact

Exposition des Comptes : La capacité à identifier quels comptes existent sur le système.

Risque accru de détournement de compte : Avec la connaissance des comptes valides, un attaquant pourrait mener des attaques de force brute ou d'autres techniques pour accéder à ces comptes.

Perte de confiance : Les utilisateurs pourraient perdre confiance dans la sécurité du système.

Exploitability	Impact	Risk
Medium 2.8	Critical 6	Critical

Recommandations

- **Réponse Uniforme** : Assurez-vous que le système renvoie une réponse uniforme quelle que soit l'entrée fournie (utilisateur valide ou non).

- **Rate Limiting** : Implémentez une limitation du nombre de requêtes pour prévenir les attaques automatisées.

- **Captcha** : Inclure un CAPTCHA pour compliquer l'automatisation des tentatives.

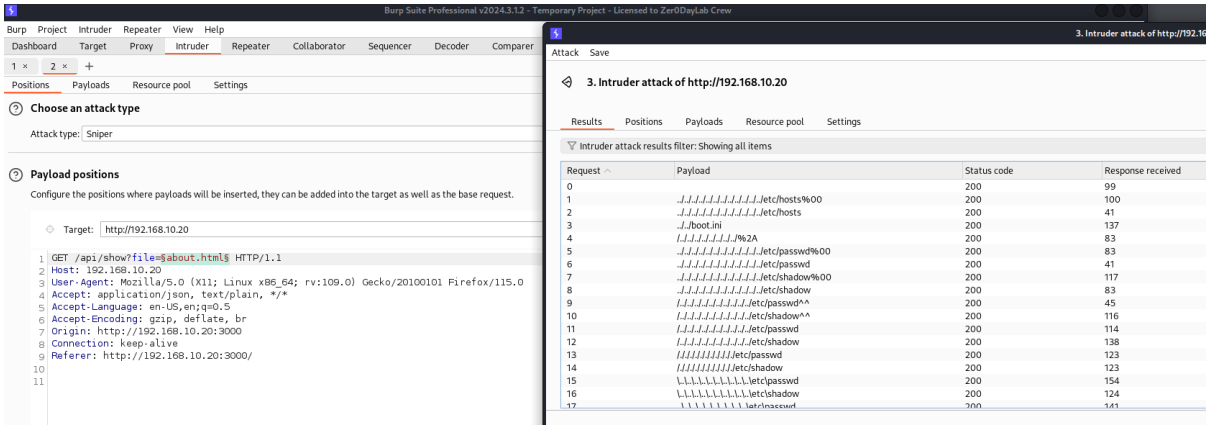
- **Chiffrement et Nonce** : Utilisez un jeton unique qui expire après un certain temps pour les requêtes oubli de mot de passe afin d'ajouter une couche de sécurité supplémentaires.

- **Sensibilisation des Utilisateurs** : Informer les utilisateurs des bonnes pratiques en matière de sécurité de leurs mots de passe.

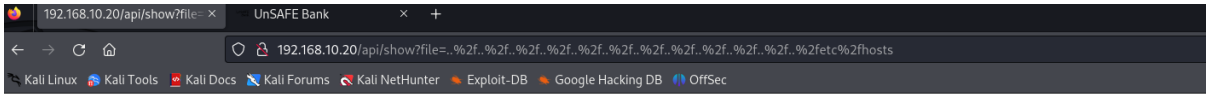
c. Énumération de fichiers de configuration serveur (IDOR) - ID03

Description de la vulnérabilité

L'application web présente une vulnérabilité de type Path Traversal au niveau du paramètre file, utilisé dans les requêtes HTTP pour spécifier des fichiers. En manipulant ce paramètre, il est possible de parcourir les répertoires du serveur et d'accéder à des fichiers sensibles en dehors du répertoire prévu par l'application.

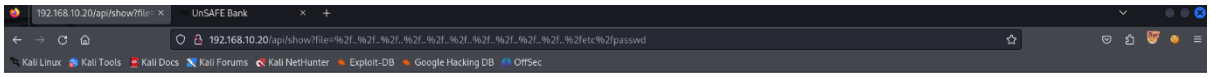


Fichier /etc/hosts



127.0.0.1 localhost ::1 localhost ip6-localhost ip6-loopback fe00::0 ip6-localnet ff00::0 ip6-mcastprefix ff02::1 ip6-allnodes ff02::2 ip6-allrouters 172.18.0.3 0f154620e934

Fichier /etc/passwd



root:x:0:0:root:/root:/bin/ash bin:x:1:1:bin:/bin:/sbin/nologin daemon:x:2:2:daemon:/sbin:/sbin/nologin adm:x:3:4:adm:/var/adm:/sbin/nologin lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin sync:x:5:0:sync:/sbin:/bin/sync shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown halt:x:7:0:halt:/sbin:/sbin/halt mail:x:8:12:mail:/var/spool/mail:/sbin/nologin news:x:9:13:news:/usr/lib/news:/sbin/nologin uucp:x:10:14:uucp:/var/spool/uucppublic:/sbin/nologin operator:x:11:0:operator:/root:/bin/sh man:x:13:15:man:/usr/man:/sbin/nologin postmaster:x:14:12:postmaster:/var/spool/mail:/sbin/nologin cron:x:16:16:cron:/var/spool/cron:/sbin/nologin ftp:x:21:21::/var/lib/ftp:/sbin/nologin sshd:x:22:22:sshd:/dev/null:/sbin/nologin at:x:25:25:at:/var/spool/cron/atjobs:/sbin/nologin squid:x:31:31:Squid:/var/cache/squid:/sbin/nologin xfs:x:33:33:X Font Server:/etc/X11/fs:/sbin/nologin games:x:35:35:games:/usr/games:/sbin/nologin postgres:x:70:70::/var/lib/postgresql:/bin/sh cyrus:x:85:12::/usr/cyrus:/sbin/nologin vpopmail:x:89:89::/var/vpopmail:/sbin/nologin ntp:x:123:123:NTP:/var/empty:/sbin/nologin smmsp:x:209:209:smmsp:/var/spool/mqueue:/sbin/nologin guest:x:405:100:guest:/dev/null:/sbin/nologin nobody:x:65534:65534:nobody:/:/sbin/nologin www-data:x:82:82:Linux User,,,:/home/www-data:/bin/false

Étapes pour reproduire la vulnérabilité

1. Utiliser Burp Suite pour intercepter les requêtes HTTP. Voir les requêtes dans l'onglet Proxy > Intercept.
2. Repérez la requête qui envoie un paramètre utilisateur via une URL : « GET/api/show ?file=$about.html$ http/1.1 »
3. Faire un clic droit sur la requête et sélectionner **Send to Intruder**.

4. Allez dans l'onglet **Intruder** et sélectionnez la requête que vous avez envoyée.
5. Dans l'onglet Positions, définir les positions où les payloads (valeurs malveillantes) seront insérées.
6. Sélectionner le paramètre file=about.html et cliquer sur Add § pour marquer cette position.
7. Sélectionner l'onglet **Payloads** pour définir les valeurs malveillantes (payloads) à tester.
8. Mettre plusieurs payloads.
9. Lancer l'attaque.
10. Observer les résultats.

Impact

1. **Divulgation d'informations sensibles**

a. Accès à des fichiers système importants comme /etc/passwd, /etc/shadow, /etc/hosts.

b. Exposition de données sensibles telles que les mots de passe chiffrés, les configurations système et réseau.

2. **Divulgation d'informations sensibles**

a. Accès aux mots de passe chiffrés dans /etc/shadow, ce qui permet des attaques par force brute pour casser les mots de passe.

1 Compromission de comptes utilisateurs (y compris les comptes avec des privilèges élevés).

3. **Perte de confidentialité et intégrité**

a. Exposition de données personnelles ou sensibles des utilisateurs.

b. Modification ou altération de fichiers systèmes ou d'applications, compromettant l'intégrité du serveur.

4. **Compromission complète du serveur**

a. Prise de contrôle totale du serveur si l'attaquant obtient des privilèges élevés ou des informations critiques.

b. Utilisation des informations recueillies pour attaquer d'autres systèmes sur le réseau, provoquant une propagation de l'attaque.

Exploitability	Impact	Risk
Medium 2.3	Important 3.7	Important

Recommandations

1. **Validation des entrées utilisateur**

- **Vérification stricte des entrées** : s'assurer que les paramètres de l'URL, tels que les fichiers demandés, sont validés pour interdire les séquences de traversal (../ ou ..\). Utiliser des expressions régulières ou des fonctions de validation de chaîne de caractères pour bloquer ces caractères.

- Liste blanche : Limiter les fichiers accessibles aux utilisateurs en utilisant une **liste blanche** de fichiers et de répertoires autorisés. S'assurer que seuls des fichiers spécifiques, connus et sûrs, peuvent être accédés.

2. **Utilisation de chemins absolus sécurisés**

- **Chemins relatifs sûrs** : Eviter d'accepter directement des chemins relatifs fournis par l'utilisateur.

- Normalisation des chemins : Avant d'utiliser un chemin, le normaliser pour éliminer toute séquence de traversal (../).

3. **Limiter les permissions du serveur**

Principes de moindre privilège : Restreindre les permissions d'accès aux fichiers et répertoires sensibles pour les utilisateurs de l'application et les processus serveur.

Accès aux fichiers uniquement nécessaire : L'application ne doit avoir accès qu'aux fichiers et répertoires strictement nécessaires au bon fonctionnement de l'application.

4. **Configurer les serveurs pour bloquer les attaques**

Contrôles côté serveur : Configurer le serveur web (Apache, Nginx, etc.) pour qu'il n'accepte pas de requêtes contenant des traversées de répertoires ou des chemins absolus potentiellement malveillants.

Désactivation de la navigation dans les répertoires : Empêcher la navigation dans les répertoires, c'est-à-dire l'affichage des répertoires s'ils sont laissés vides ou non protégés (par exemple, en désactivant Options Indexes dans Apache).

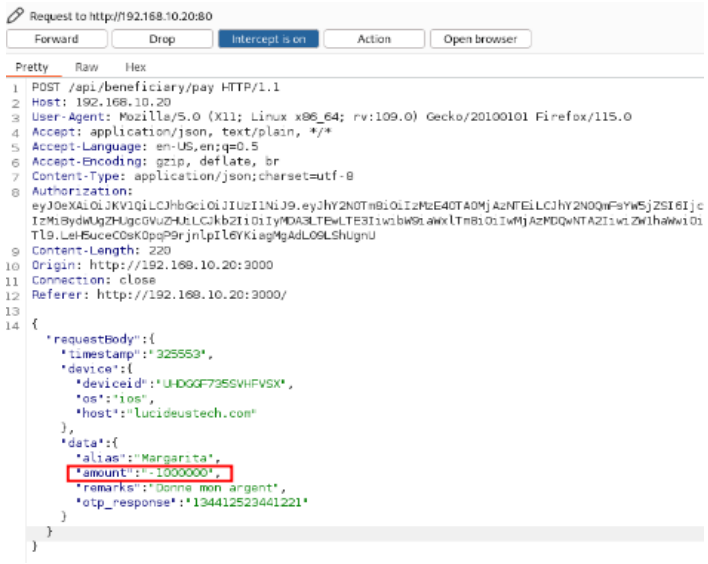
3. Détermination des vulnérabilités par des tests manuels

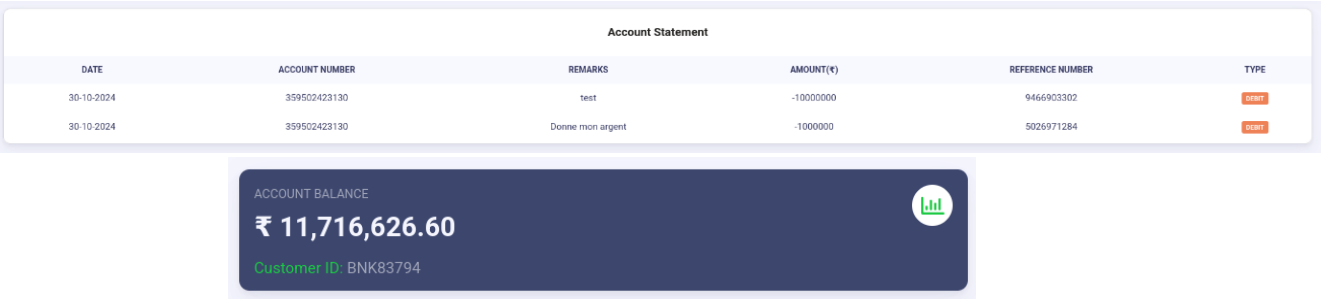
a. Gain d'argent grâce par attaque par rejeu (BURP interception) - ID04

Description de la vulnérabilité

Cette vulnérabilité de type **attaque par manipulation des données d'entrée (Input Manipulation Attack)** réside dans le manque de validation côté serveur concernant les transactions financières. Lorsqu'un utilisateur effectue un virement bancaire, la requête est interceptée via un proxy comme Burp Suite.
L'attaquant peut modifier la valeur de la somme envoyée, notamment en saisissant un montant négatif.
Cela permet de créditer indûment son propre compte ou de débiter des sommes importantes du compte cible, créant une faille critique dans le système bancaire.

Input Manipulation Attack : Les attaques par manipulation d'entrée sont un terme générique qui inclut les attaques adverses, un type d'attaque dans lequel un attaquant modifie délibérément les données d'entrée pour induire le modèle en erreur.





Étapes pour reproduire la vulnérabilité

1. Connexion à l'application :
    a. Connectez-vous avec des identifiants valides sur l'application bancaire.
    b. Accédez à la section permettant d'effectuer des virements bancaires (Bank transfer).
2. Préparation de l'interception avec Burp Suite :
    a. Configurez le navigateur ou l'application mobile pour passer par Burp Suite comme proxy.
    b. Activez l'interception dans Burp Suite pour capturer les requêtes sortantes.
3. Initiation d'un virement bancaire légitime :
    a. Saisissez un montant positif (ex. : 100) pour effectuer un transfert vers un autre compte.
    b. Cliquez sur "Envoyer" ou "Confirmer" pour initier le virement.
4. Modification de la requête interceptée :
    a. Lorsque Burp Suite intercepte la requête, repérez le champ contenant le montant (amount).
    b. Modifiez ce champ pour inclure un montant négatif (ex. : -500).
5. Renvoyer la requête modifiée :
    a. Relâchez la requête modifiée en la transmettant au serveur.
    b. Vérifiez si la transaction est acceptée par le serveur sans contrôle adéquat.
6. Observation des résultats :
    a. Consultez les soldes des comptes pour confirmer le crédit indu ou le débit excessif.
    b. Reproduisez éventuellement l'attaque avec d'autres montants pour valider l'ampleur du problème.

Impact

Cette vulnérabilité peut avoir des conséquences **critiques** pour l'application et ses utilisateurs, notamment :

1. Perte financière directe

- Un utilisateur malveillant peut exploiter cette faille pour transférer de l'argent indûment depuis les comptes d'autres utilisateurs, ou même créditer son propre compte de montants fictifs.
- La fraude peut entraîner des pertes financières importantes pour l'entreprise, car elle serait tenue de rembourser les victimes, parfois même en absorbant des montants très élevés.

2. Perte de confiance des clients

- Les clients concernés pourraient perdre confiance en la sécurité de la plateforme bancaire, ce qui pourrait se traduire par une fuite massive de clients vers des concurrents.

- Une telle perte de confiance peut aussi compromettre l'attractivité de l'entreprise auprès de nouveaux clients potentiels.

3. Risques juridiques et réglementaires

- L'entreprise pourrait être poursuivie pour avoir échoué à protéger les fonds et les données sensibles de ses utilisateurs.
- Les régulateurs (ex. : autorités financières) pourraient imposer des amendes sévères pour non-conformité aux normes de sécurité, comme le RGPD ou les directives spécifiques au secteur bancaire.

4. Atteinte à la réputation de l'entreprise

- Une faille de sécurité critique, si elle devient publique, pourrait nuire à l'image de marque de l'entreprise et faire les gros titres dans la presse, renforçant la méfiance des parties prenantes (clients, partenaires, investisseurs).

5. Exploitation à grande échelle

- Si un attaquant décide de mener une exploitation automatisée ou massive, les pertes pourraient être exponentielles.
- Les systèmes de surveillance de l'entreprise pourraient ne pas détecter immédiatement ces transactions frauduleuses, aggravant ainsi la situation.

Exploitability	Impact	Risk
Medium 2.3	Major 4.7	Major

Recommandations

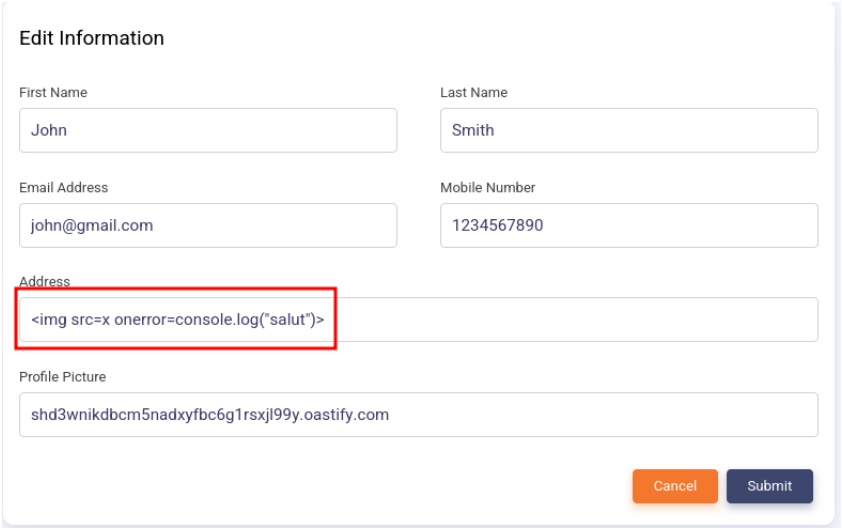
Pour corriger cette vulnérabilité et prévenir de futurs exploits similaires, voici les actions à entreprendre :

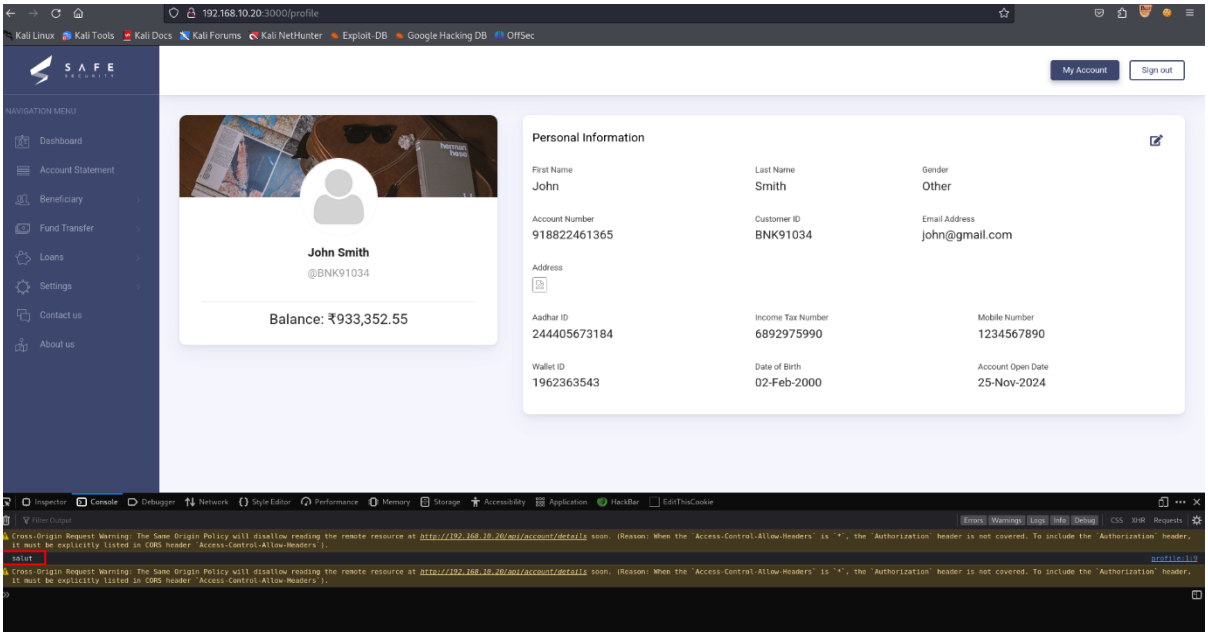
1. **Validation côté serveur** :
    a. Mettre en place une vérification stricte de toutes les données reçues côté serveur, en particulier pour le champ du montant.
    b. Ajouter des règles métiers interdisant explicitement les montants négatifs ou incohérents (par exemple, amount > 0).
2. **Vérification de l'intégrité des requêtes** :
    a. Ajouter des signatures numériques ou des hachages pour garantir que les données envoyées par le client n'ont pas été modifiées lors de leur transit.
3. **Utiliser des tokens anti-rejeu** :
    a. Implémenter des tokens uniques pour chaque requête, empêchant leur modification ou leur réexécution.
    b. Rejeter toute requête répétée ou altérée.

b. Injection XSS sur la page de profil - ID05

Description de la vulnérabilité

Une injection **XSS (Cross-Site Scripting)** est une vulnérabilité de sécurité web où un attaquant injecte un code malveillant (souvent du JavaScript) dans une application web. Ce code est ensuite exécuté par le navigateur des utilisateurs qui visitent la page compromise. Ici elle est possible dans le champ adresse de la page de modifications des informations d'un utilisateur.





Étapes pour reproduire la vulnérabilité

1. Se connecter à l'application.

2. Accéder à la page de profil
3. Modifier les informations du profil
4. Insérer du code XSS dans le champ "Address"
5. Valider les modifications

Impact

- Défacement du site donc perte d'argent et de réputation
- Phishing
- Vol de session
- Modifier les données de Business Logic
- Port Scanning

Exploitability	Impact	Risk
Medium 2.1	Major 4.2	Major

Recommandations

- **Escaping** : Échapper toutes les entrées d'utilisateur pour les sécuriser avant d'être rendus à vos utilisateurs finaux.
- **Validation des entrées**
- **Sanitising** : désinfection des données qui sont dans un format inacceptable et qui les met dans un format acceptable

4. Configuration générale de l'application WEB

a. Mauvaises configurations

Configuration	Description
Complexité des mots de passe	Pas de vérification de la complexité ce qui permet de créer des mots de passe sensibles aux attaques par brute force
One Time Password	Sécurisation mais mauvaise implémentation, accessible directement sur le site ce qui permet aux attaquants de valider leurs actions sans difficulté.
SSL	Pas de SSL ce qui rend le site web sensible à des attaques de Man-In-The-Middle etc...
Fichiers sources	Accès aux fichiers sources et à l'arborescence du site en inspectant le site. Cela permet aux attaquants de comprendre la structure du site et de trouver des failles.

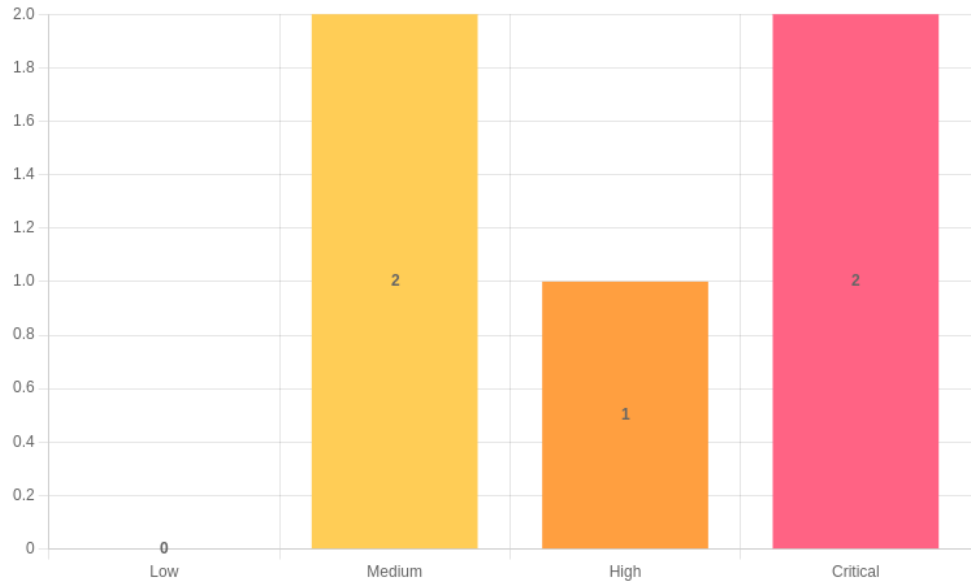
b. Bonnes configurations

Configuration	Description
Champs dans la page modification de profil	Vérification des champs ce qui permet de ne pas insérer des payloads dans les champs de numéro et l'url pour l'image s'ils ne sont pas dans le format demandé
ID sur la page de connexion	Vérification du format de l'ID qui permet de ne pas entrer une injection SQL dans le champ ID
Session	La session s'expire au bout d'un certain temps ce qui permet d'éviter d'abuser des droits d'accès de la session

5. Synthèse du test d'intrusion

a. Bilan de l'audit

Le test d'intrusion a permis d'identifier plusieurs vulnérabilités sur le périmètre cible, le diagramme ci-dessous représente la répartition en termes de gravité technique :



Les vulnérabilités de gravité forte et critique constituent la principale source de menace envers le SI, celles-ci permettent à un attaquant de compromettre rapidement et facilement l'organisation.

Les vulnérabilités moyennes et faibles sont principalement des vulnérabilités liées à un non-respect des bonnes pratiques de configuration et sécurité pouvant amener (mise bout à bout) à une compromission du SI.

b. Plan d'actions

Gravité	ID	Vulnérabilité	Remédiation
Critical	ID01	IDOR - Informations comptes	• **Contrôle d'accès côté serveur** • **Utilisation de jetons sécurisés** • **Contrôles d'accès rigoureux**
Critical	ID02	Changement mot de passe non sécurisé	• **Réponse Uniforme** • **Rate Limiting** • **Captcha** • **Chiffrement et Nonce** • **Sensibilisation des Utilisateurs**
Medium	ID03	IDOR – Fichiers de configuration serveur	• **Validation des entrées utilisateur** • **Utilisation de chemins absolus sécurisés** • **Limiter les permissions du serveur** • **Configurer les serveurs pour bloquer les attaques**
High	ID04	Input Manipulation Attack – Gain d'argent	• **Validation côté serveur** • **Vérification de l'intégrité des requêtes** • **Utiliser des tokens anti-rejeu**
Medium	ID05	Injection XSS	• **Escaping** • **Validation des entrées** • **Sanitising**