LOG8371

Ingénierie de la qualité logicielle

Travail Pratique #3

Équipe: Sherbrooke

2002127 – Chebbi, Aymen

1621855 – Gafsi, Ahmed

2089012 – Nde Tsakou, Homer

2091172 – Potvin, David

Remis à : Patrick Loic Foalem

Hiver 2024

Table de matières

DSVW

Introduction DSVW

Damn Small Vulnerable Web (DSVW) est une application web délibérément vulnérable écrite en moins de 100 lignes de code, créée à des fins éducatives. Cette application prend en charge la majorité des vulnérabilités les plus courantes des applications web, ainsi que les attaques appropriées. Conçue pour être facile à déployer et à utiliser, DSVW permet aux développeurs, aux étudiants et aux professionnels de la sécurité de comprendre et de tester les faiblesses potentielles de leurs propres applications web.

Développée en Python 3.x, DSVW offre une plateforme simple et légère pour explorer diverses failles de sécurité telles que les attaques XSS (Cross-Site Scripting) et bien d'autres. Son déploiement est également simplifié grâce à sa configuration minimale. Pour lancer l'application, il vous suffit d'exécuter la commande "python3 dsvw.py", puis de naviguer vers l'URL fournie dans votre navigateur. Les utilisateurs peuvent également installer les dépendances nécessaires via pip en exécutant "pip install -r requirements.txt", ce qui garantit un déploiement fluide et rapide de l'application. En bref, DSVW offre une solution pratique et efficace pour expérimenter et comprendre les vulnérabilités courantes des applications web.

Résultats d’Analyse Statique de DSVW

Sommaire des résultats par SonarCloud

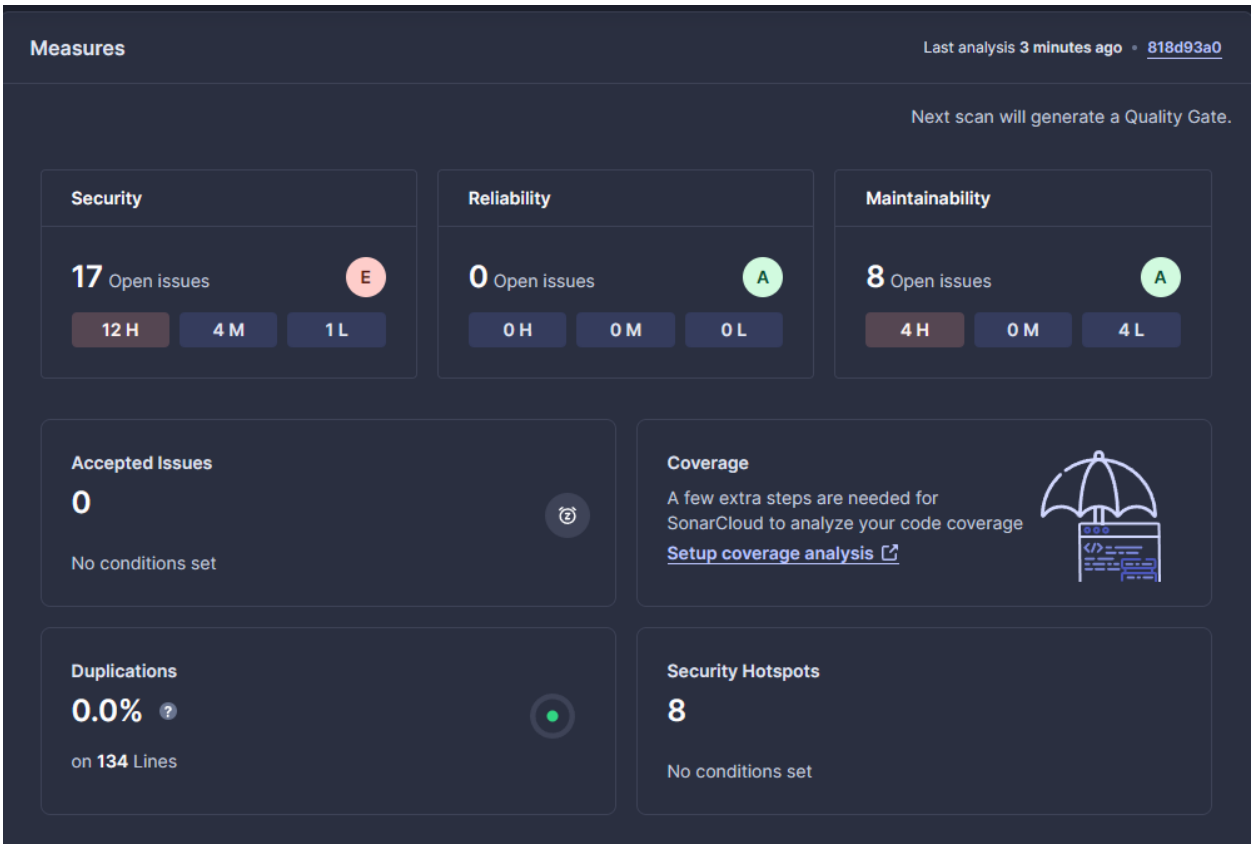

Figure 1. Sommaire des résultats de SonarCloud de DSVW

Commentaires sur des vulnérabilités ou des hotspots de sécurités

Tableau 1. Vulnérabilités (ou hotspots) identifiées de DSVW

id	Nom du fichier	Criticité	Type de vulnérabilité*	Description du risque	Recommendation
1	dsvw.py	Élevée	Password Plaintext Storage	Le fichier contient un mot de passe et est directement accessible sur le repos et le mot de passe n’est pas sécuriser.	Créer un fichier de configuration local contenant les identifiants et ne pas le push sur le repos.
2	dsvw.py	Moyenne	Insecure Randomness	L’application utilise un générateur pseudoaléatoire de	Utiliser un algorithme de génération plus

				lettres et chiffres pour générer des données sensibles. Un individu mal intentionné pourrait prédire les séquences résultantes.	élaboré et moins facile à reproduire.
3	dsvw.py	Faible	Use of Obsolete Methods	L'application utilise à plusieurs reprises le protocole HTTP au lieu du protocole HTTPS. Ce dernier est déprécié, car il ne crypte pas les paquets lors de la communication, rendant ainsi possible l'interception de ces derniers.	Passer au protocole HTTPS.
4	dsvw.py	Élevée	Injection Problem	L'application construit du code SQL directement depuis des données contrôlées par l'utilisateur.	Effectuer la sanitisation des données avant de générer le code SQL.
5	dsvw.py	Élevée	Deserialization of untrusted data	Le code désérialise des données contrôlées par le client, ce qui expose le serveur à des attaques.	Mettre en place une validation stricte des données entrantes et d'utiliser des mécanismes de désérialisation sûrs, comme des bibliothèques ou des fonctions spécifiques qui évitent les injections de code malveillant
6	dsvw.py	Élevée	Injection problem	L'application charge des fichiers en construisant un chemin provenant directement des données contrôlées par l'utilisateur.	Valider les données et le chemin résultant contre les permissions de l'utilisateur avant de load le fichier.
7	dsvw.py	Élevée	XML External Entity (XXE) Processing	Le code permet d'accéder à des entités externes lors du parsing	Désactiver l'accès aux entités externes lors du

				de XML, ce qui peut mener à des attaques XML External Entity (XXE).	"parsing".
8	dsvw.py	Élevée	Missing XML Validation	Le code exploite du XML non validé.	Ajouter une étape de validation avant d’utiliser le XML.
9	dsvw.py	Élevée	Information exposure through query strings in URL	Le code construit des URL directement depuis des données contrôlées par le client sans les valider.	Ajouter une étape de contrôle des URL construits.
10	dsvw.py	Élevée	Privacy Violation	Le code a des vulnérabilités permettant d’accéder aux données d’autres utilisateurs à l’aide d’attaques Reflected cross-site scripting	Mettre en place une validation rigoureuse des entrées utilisateur, en particulier en ce qui concerne les données sensibles.

Résultats des tests d’intrusion de DSVW

Alert type	Risk	Count
Cross Site Scripting (réfléchi)	Haut	2 (11,1 %)
Injection SQL - SQLite	Haut	9 (50,0 %)
Redirection externe	Haut	1 (5,6 %)
SQL Injection - SQLite	Haut	6 (33,3 %)
Traversée de chemin	Haut	1 (5,6 %)
Absence de Jetons Anti-CSRF	Moyen	1 (5,6 %)
Content Security Policy (CSP) Header Not Set	Moyen	29 (161,1 %)
Missing Anti-clickjacking Header	Moyen	27 (150,0 %)
Application Error Disclosure	Faible	6 (33,3 %)
Private IP Disclosure	Faible	2 (11,1 %)
Server Leaks Version Information via "Server" HTTP Response Header Field	Faible	44 (244,4 %)
X-Content-Type-Options Header Missing	Faible	36 (200,0 %)
Authentication Request Identified	Pour information	3 (16,7 %)
Information Disclosure - Sensitive Information in URL	Pour information	6 (33,3 %)
Information Disclosure - Suspicious Comments	Pour information	29 (161,1 %)
Modern Web Application	Pour information	1 (5,6 %)
User Controllable Charset	Pour information	1 (5,6 %)
User Controllable HTML Element Attribute (Potential XSS)	Pour information	2 (11,1 %)
Total		18

Figure 2. Résultat du test d’intrusion de ZAP pour l’application DSVW

Rapport d’analyse de Sécurité - ZAP

Date de Génération: Mercredi 3 Avril 2024
Version ZAP: 2.14.0

Site Web Testé: http://127.0.0.1:6541

Identification des Vulnérabilités:

Les vulnérabilités détectées par ZAP incluent certaines énumérées dans l'OWASP Top 10. Voici les plus remarquables [1][2][3][4] :

Tableau 2. Vulnérabilités remarquables détectées par ZAP pour DSVW

Vulnérabilité	Risque	Nombre d’occurrences	Impact Potentiel
Injection SQL	Élevé	9	Permet aux attaquants de lire, modifier, ou supprimer des données dans la base de données.
Cross Site Scripting (réfléchi)	Élevé	2	Permet l'insertion de scripts malveillants qui peuvent être exécutés dans le navigateur des utilisateurs
Redirection externe	Élevé	1	Permet de rediriger les utilisateurs vers des sites malveillants ou d’hameçonnage.
Traversée de chemin	Élevé	1	Accès non autorisé à des fichiers et répertoires en dehors de la racine du serveur web.

Description des vulnérabilités

Dans cette section, nous allons explorer les diverses vulnérabilités identifiées par ZAP lors des tests d'intrusion. Notre analyse a révélé 18 vulnérabilités distinctes, illustrées dans le graphique ci-dessus. Ces vulnérabilités seront détaillées et contextualisées en fonction des listes OWASP Top 10 des années 2021, 2017 et 2013. Pour chacune d'entre elles, détectées dans notre rapport ZAP, nous fournirons une description spécifique du risque encouru, ainsi que des recommandations stratégiques pour leur résolution et atténuation.

1. Cross Site Scripting (réfléchi) (Risque Haut)
   - OWASP Top 10 : Présent dans les listes de 2021 (A3:2021-Injection), 2017 (A7:2017-Cross-Site Scripting (XSS)), et 2013 (A3-Cross-Site Scripting (XSS)).
   - Description : Permet l'insertion de scripts malveillants dans des pages Web qui sont ensuite visualisées par d'autres utilisateurs.
   - Recommandation : Sanitiser et valider toutes les entrées utilisateur pour éviter l'injection de scripts.

2. Injection SQL - SQLite (Risque Haut)
   - OWASP Top 10 : Présent dans les listes de 2021 (A3:2021-Injection), 2017 (A1:2017-Injection), et 2013 (A1-Injection).
   - Description : Permet à un attaquant d'injecter et d'exécuter des commandes SQL arbitraires sur la base de données.
   - Recommandation : Utilisez des requêtes préparées et des procédures stockées pour empêcher l'injection SQL.

3. Redirection externe (Risque Haut)
   - OWASP Top 10 : Non spécifiquement listée, elle se rapporte aux problèmes de sécurité liés à la gestion des redirections et des requêtes.
   - Description : Redirection non sécurisée vers des sites externes, pouvant être utilisée pour des attaques d’hameçonnage ou de redirection malveillante.
   - Recommandation : Valider toutes les redirections URL pour s'assurer qu'elles pointent vers des destinations sûres.

4. SQL Injection - SQLite (Risque Haut)
   - Description : Variante de l'injection SQL permettant l'exécution de commandes SQL non désirées.
   - Recommandation : Appliquer les mêmes mesures que pour l'injection SQL classique, comme l'utilisation de requêtes préparées.

5. Traversée de chemin (Risque Haut)
   - OWASP Top 10 : Proche de 2021 (A5:2021-Security Misconfiguration), 2017 (A5:2017-Broken Access Control).
   - Description : Accès non autorisé à des fichiers hors du répertoire web racine.

- Recommandation : Vérifiez et validez tous les chemins d'accès aux fichiers sur le serveur pour prévenir l'accès non autorisé.

6. Absence de Jetons Anti-CSRF (Risque Moyen)
- OWASP Top 10 : Proche de 2013 (A8-Cross-Site Request Forgery (CSRF)).
- Description: Cette vulnérabilité survient lorsqu'une application web n'utilise pas de jetons anti-CSRF, rendant le site vulnérable à des attaques où un attaquant peut forcer l'utilisateur à exécuter des actions non désirées sur une application web où il est authentifié.
- Recommandation : Implémenter des jetons CSRF pour toutes les formes et requêtes sensibles. Les jetons CSRF doivent être uniques et imprévisibles, assurant ainsi que chaque demande d'un utilisateur authentifié est validée pour son origine légitime.

7. Content Security Policy (CSP) Header Not Set (Risque Moyen)
- OWASP Top 10 : Associé à 2021 (A5:2021-Security Misconfiguration).
- Description: Cette vulnérabilité se présente quand une application web ne définit pas une politique de sécurité du contenu (CSP). Sans CSP, le site est plus vulnérable aux attaques XSS, car il ne restreint pas les ressources (scripts, CSS, etc.) pouvant être chargées et exécutées.
- Recommandation: Mettre en place une Content Security Policy stricte. Définissez des directives CSP pour contrôler les sources de contenu et les types de ressources pouvant être chargés et exécutés sur le site, réduisant ainsi le risque d'attaques XSS et d'autres injections de contenu malveillant.

8. Missing Anti-clickjacking Header (Risque Moyen)
- OWASP Top 10 : Associé à des problèmes de sécurité des clients, mais pas spécifiquement listé dans OWASP Top 10.
- Description : Cette vulnérabilité se produit lorsque les en-têtes HTTP nécessaires pour prévenir le clickjacking ne sont pas présents. Le clickjacking est une technique où un utilisateur est trompé pour cliquer sur quelque chose de différent de ce qu'il perçoit, souvent par le biais d'un contenu malveillant superposé.
- Recommandation : Implémenter l'en-tête 'X-Frame-Options' sur le serveur web. Cet en-tête peut être configuré pour empêcher la page d'être chargée dans un iframe ou seulement autoriser son chargement dans des iframes provenant du même origine, aidant ainsi à prévenir les attaques de clickjacking.

9. Application Error Disclosure (Risque Faible)
- OWASP Top 10 : Proche de 2021 (A5:2021-Security Misconfiguration).
- Description : Divulgation d'informations sensibles à travers les messages d'erreur de l'application.
- Recommandation : Personnaliser les messages d'erreur pour éviter la divulgation d'informations détaillées.

10. Private IP Disclosure (Risque Faible)
- OWASP Top 10 : Relatif à A5:2021-Security Misconfiguration.
- Description : Fuite d'adresses IP privées qui peuvent révéler la structure interne du réseau.

- Recommandation : Configurer les serveurs et applications pour masquer les informations internes.

11. Server Leaks Version Information (Risque Faible)
- OWASP Top 10 : Relatif à A5:2021-Security Misconfiguration.
- Description : Cette vulnérabilité apparaît quand un serveur web divulgue des informations sur sa version, par exemple dans les en-têtes HTTP. Ces informations peuvent aider un attaquant à identifier des failles de sécurité spécifiques à la version du serveur ou du logiciel utilisé.
- Recommandations : Configurer le serveur pour masquer ou supprimer les détails de version des en-têtes HTTP et autres réponses du serveur. Il est crucial de s'assurer que les informations divulguées n'aident pas les attaquants à cibler des vulnérabilités spécifiques.

12. X-Content-Type-Options Header Missing (Risque Faible)
- OWASP Top 10 : Lié à la sécurité des clients, bien que non spécifiquement listé.
- Description : Cette vulnérabilité se produit lorsque l'en-tête HTTP 'X-Content-Type-Options' est absent, permettant aux navigateurs de deviner et d'interpréter le type MIME du contenu servi, ce qui peut conduire à des attaques basées sur MIME-sniffing.
- Recommandations : Ajouter l'en-tête 'X-Content-Type-Options: nosniff' dans les réponses HTTP de votre serveur. Cela empêche les navigateurs de deviner le type MIME du contenu et les oblige à s'en tenir au type déclaré dans l'en-tête 'Content-Type', réduisant ainsi le risque d'attaques XSS et autres vulnérabilités liées au type de contenu.

13. Authentication Request Identified
- OWASP Top 10 : Non spécifiquement listé, mais relatif à A7:2021-Identification and Authentication Failures.
- Description : Identification des requêtes d'authentification qui peuvent nécessiter un examen.
- Recommandations : Vérifiez ces requêtes pour garantir leur sécurité.

14. Information Disclosure - Sensitive Information in URL
- OWASP Top 10 : Non spécifiquement listé, mais peut être lié à A3:2021-Injection pour les fuites d'informations potentielles.
- Description : Divulgation potentielle d'informations sensibles dans les URL.
- Recommandation : Évitez de placer des données sensibles dans les URL.

15. Information Disclosure - Suspicious Comments
- OWASP Top 10 : Non spécifiquement listé, mais peut être lié à A5:2021-Security Misconfiguration.
- Description : Commentaires suspects dans le code pouvant révéler des informations internes.
- Recommandation : Nettoyez le code de tous les commentaires sensibles ou informatifs avant la mise en production.

16. Modern Web Application
- OWASP Top 10 : Non spécifiquement listé, mais pourrait être lié aux meilleures pratiques de développement de logiciels modernes.

- Description : Identification générale d'une application web moderne.
- Recommandations : Aucune action spécifique nécessaire, mais gardez l’application à jour.

17. User Controllable Charset
- OWASP Top 10 : Non spécifiquement listé, mais peut être lié à A3:2021-Injection.
- Description : Permet aux utilisateurs de contrôler le jeu de caractères, ce qui pourrait conduire à des vulnérabilités.
- Recommandation : Validez et sanitisez toutes les entrées contrôlant le jeu de caractères.

18. User Controllable HTML Element Attribute (Potential XSS)
- OWASP Top 10 : Relatif à A3:2021-Injection.
- Description : Les utilisateurs peuvent contrôler les attributs des éléments HTML, augmentant le risque de XSS.
- Recommandation : Validez et sanitisez strictement toutes les entrées utilisateur influençant les attributs HTML.

Comparaison des résultats entre SonarCloud et ZAP de DSVW

Pour comparer les résultats de SonarCloud à ceux de Zap, on a tout d'abord trouvé des correspondances entre les vulnérabilités de SonarCould et les vulnérabilités qui sont dans le catalogue CWE. Ces correspondances sont résumées dans le tableau ci-dessous :

Tableau 3. Vulnérabilités de SonarCloud pour DSVW

Vulnérabilités SonarCloud (tableau 1)		Catalogue CWE	
id	Description	CWE	Relation OWASP
1	Le fichier contient un mot de passe et est directement accessible sur le repository et le mot de passe n'est pas sécurisé.	CWE-256: Plaintext Storage of a Password [5]	OWASP Top Ten 2021 Category A04:2021 - Insecure Design [5]
2	L'application utilise un générateur pseudoaléatoire de lettres et chiffres pour générer des données sensibles. Un individu mal intentionné pourrait prédire les séquences résultantes.	CWE-338: Use of Cryptographically Weak Pseudo-Random Number Generator (PRNG) [6]	OWASP Top Ten 2021 Category A02:2021 - Cryptographic Failures [6]
3	L'application utilise à plusieurs reprises le protocole HTTP au lieu du protocole HTTPS. Ce dernier est déprécié, car il ne crypte pas les paquets lors de la communication, rendant ainsi possible l'interception de ces derniers.	CWE-319: Cleartext Transmission of Sensitive Information [7]	OWASP Top Ten 2021 Category A02:2021 - Cryptographic Failures [7]
4	L'application construit du code SQL directement depuis des données contrôlées par l'utilisateur.	CWE-20: Improper Input Validation [8]	OWASP Top Ten 2021 Category A03:2021 - Injection [8]
5	Le code désérialise des données contrôlées par le client, ce qui expose le serveur à des attaques	CWE-502: Deserialization of Untrusted Data [9]	OWASP Top Ten 2021 Category A08:2021 - Software and Data Integrity Failures [9]
6	L'application charge des fichiers en construisant un chemin directement depuis des données contrôlées par l'utilisateur.	CWE-73: External Control of File Name or Path [10]	OWASP Top Ten 2021 Category A04:2021 - Insecure Design [10]

7	Le code permet d'accéder à des entités externes lors du "parsing" de XML, ce qui peut mener à des attaques XML External Entity (XXE).	CWE-611: Improper Restriction of XML External Entity Reference [11]	OWASP Top Ten 2021 Category A05:2021 - Security Misconfiguration [11]
8	Le code exploite du XML non validé.	CWE-112: Missing XML Validation [12]	OWASP Top Ten 2021 Category A03:2021 - Injection [12]
9	Le code construit des URL directement depuis des données contrôlées par le client sans les valider.	CWE-601: URL Redirection to Untrusted Site ('Open Redirect') [13]	OWASP Top Ten 2021 Category A01:2021 - Broken Access Control [13]
10	Le code a des vulnérabilités permettant d'accéder aux données d'autres utilisateurs à l'aide d'attaques Reflected cross-site scripting	CWE-79: Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting') [14]	OWASP Top Ten 2021 Category A03:2021 - Injection [14]

Ensuite, on a comparé les vulnérabilités du catalogue CWE identifiées à celles identifiées par ZAP. Les vulnérabilités de ZAP qui ont une correspondance avec celles de SonarCloud sont énumérées ci-dessous :

Tableau 4. Vulnérabilités identifiées par ZAP et SonarCloud

Vulnérabilités SonarCloud	Vulnérabilités Zap		
id	no.	Description	Relation OWASP
4	2	Permet à un attaquant d'injecter et d'exécuter des commandes SQL arbitraires sur la base de données.	OWASP Top Ten 2021 Category A03:2021 - Injection
10	1	Permet l'insertion de scripts malveillants dans des pages Web qui sont ensuite visualisées par d'autres utilisateurs.	OWASP Top Ten 2021 Category A03:2021 - Injection

Dans ce tableau, on a établi que la 4e vulnérabilité de SonarCloud a été identifiée par ZAP et elle se réfère à la 2e vulnérabilité identifiée par ZAP.

La 4e vulnérabilité identifiée par SonarCloud a un niveau de criticité ‘élevé’ comme pour sa correspondance avec la 2e vulnérabilité ZAP qui lui attribue le niveau de risque ‘Haut’.

La 10àme vulnérabilité identifiée par SonarCloud a un niveau de criticité ‘élevé’ comme pour sa correspondance avec la première vulnérabilité ZAP qui lui attribue le niveau de risque ‘Haut’.

En d’autres mots, toutes les vulnérabilités communes trouvées par SonarCloud et ZAP ont le même niveau de criticité.

Aussi, en comparant le tableau 3 et le tableau 4, on remarque que les vulnérabilités communes ont la même relation OWASP.

À partir de cette analyse, on en déduit que la majorité des vulnérabilités identifiées par SonarCloud ne sont pas détectées par Zap et vice-versa. Ceci peut s’expliquer par le fait que les deux outils ne détectent pas les vulnérabilités de la même manière. En effet, Sonar Cloud est un outil d’analyse statique du code alors il peut être utilisé pour une application qui n’est pas en cours d'exécution, tandis que Zap simule des attaques sur une application en cours d'exécution afin de détecter des vulnérabilités et des erreurs qui ne sont pas détectables quand l’application n’est pas en cours d'exécution (contrairement a l’analyse statique) [15]. Aussi, certaines vulnérabilités comme avoir un mot de passe écrit en clair dans le code est détectable uniquement en regardant le code comme en effectuant une analyse statique du code avec SonarCloud. Par conséquent, ces deux outils se complémentent. Utiliser ces deux outils permet donc de maximiser la quantité de vulnérabilités trouvées et cela peut aider les développeurs à corriger leurs applications pour qu’elles puissent assurer l'intégrité, la confidentialité et la disponibilité des données des utilisateurs.

Vulnerable SAML App

Introduction de Vulnerable SAML App

Vulnerable SAML (Security Assertion Markup Language) App est une application conçue pour illustrer comment certaines configurations vulnérables peuvent être exploitées pour permettre à un utilisateur de modifier ses permissions, son nom, etc. au sein d'une application. Cette application utilise la bibliothèque SAML en Python de OneLogin, modifiée de manière significative pour permettre le fonctionnement des configurations vulnérables. L'infrastructure comprend deux images Docker : vulnerableidp, qui agit en tant que fournisseur d'identité, et une application web.

Pour déployer cette infrastructure, il vous suffit d'exécuter la commande "docker-compose up". Les images Docker seront construites, et l'application web sera hébergée à l'adresse http://127.0.0.1:8000. Des identifiants de connexion sont fournis pour différents types d'utilisateurs, tels que les utilisateurs non privilégiés, les administrateurs, et les instructeurs. Des instructions sont également fournies pour personnaliser les comptes utilisateurs et les groupes, ainsi que pour déployer l'application sur plusieurs hôtes.

Cette application, développée en Python et utilisant Docker pour le déploiement, offre une manière pratique et sécurisée d'explorer les vulnérabilités liées à SAML et d'apprendre comment les prévenir. Son architecture modulaire et ses instructions détaillées en font un outil précieux pour les développeurs, les étudiants et les professionnels de la sécurité cherchant à mieux comprendre les risques de sécurité associés à SAML.

Résultats d’Analyse Statique de Vulnerable SAML App

Sommaire des résultats par SonarCloud

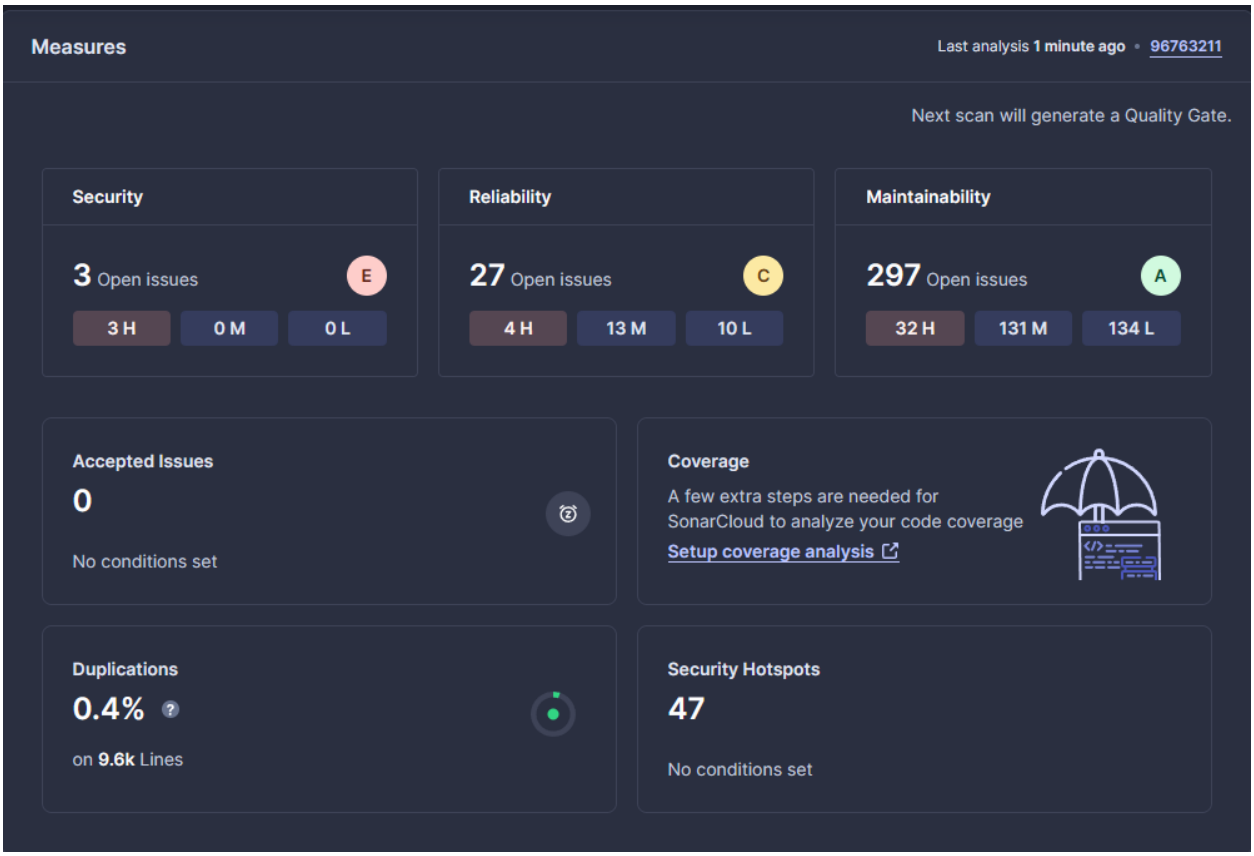

Figure 3. Sommaire des résultats de SonarCloud de Vulnerable SAML App

Commentaires sur des vulnérabilités ou des hotspots de sécurités

Tableau 5. Vulnérabilités (ou hotspots) identifiées pour Vulnerable SAML App

id	Nom du fichier	Criticité	Type de vulnérabilité*	Description	Recommendation
1	vulnerable_saml_sp/yogi SP/vulnsp.py	Élevée	Password Plaintext Storage	Une clef secréte est presente dans un fichier, accompagnée d’un identificateur “SECRET_KEY”.	Mettre en place un fichier de configuration/environnement pour ne pas rendre vulnérable les clefs secrètes.

2	vulnerablesp/yogiSP/vulnsp.py	Élevée	Process Control	Le code désactive les protections contre les attaques CSRF.	Vérifier si désactiver les mesures de protection contre les CSRF est sécuritaire dans ce contexte.
3	vulnerablesp/yogiSP/vulnsp.py	Élevée	Process Control	Le code autorise à la fois les requêtes HTTP sécuritaire et non sécuritaire.	Uniquement permettre les requêtes HTTP sécuritaires.
4	vulnerablesp/Dockerfile	Moyenne	Privacy Violation	Le docker file contient une instruction de copie récursive qui pourrait engendré l'exposition de données sensibles de certains utilisateurs.	Limitez l'utilisation du globbing dans la définition des sources de COPY et ADD.
5	vulnerableidp/Dockerfile	Moyenne	Least Privilege Violation	L'image Ubuntu spécifiée roule avec les permissions "root".	Investiguer si ce super rôle est bel et bien le rôle nécessaire.
6	vulnerablesp/yogiSP/vulnsp.py	Moyenne	Using a broken or risky cryptographic algorithm	Un générateur de séquence de lettres aléatoires à taille fixe est utilisé pour générer des clefs.	Utiliser un algorithme de cryptographie moins prédictible.
7	configure_platform.py	Moyenne	Use of Obsolete Methods	L'application utilise à plusieurs reprises le protocole HTTP aulieu du protocole HTTPS. Ce dernier est déprécié, car il ne crypte pas les paquets lors de la communication, rendant ainsi possible l'interception de ces derniers.	Passer au protocole HTTPS.
8	vulnerablesp/yogiSP/vulnsp.py	Faible	Vulnerability template	Du code de production fait appel à un mode debug.	Enlever le mode debug du code de production.
9	configure_platform.py	Moyenne	Use of Obsolete Methods	L'application utilise à plusieurs reprises le protocole HTTP aulieu du	Passer au protocole HTTPS.

				protocole HTTPS. Ce dernier est déprécié, car il ne crypte pas les paquets lors de la communication, rendant ainsi possible l'interception de ces derniers.	
10	vulnerabl esp/Dock erfile	Élevée	Least Privilege Violation	L'image Ubuntu spécifiée roule avec les permissions "root".	Investiguer si ce super rôle est bel et bien le rôle nécessaire.

Résultats des tests d’intrusion de Vulnerable SAML App

Alert type	Risk	Count
Absence de Jetons Anti-CSRF	Moyen	1 (7,1 %)
Content Security Policy (CSP) Header Not Set	Moyen	9 (64,3 %)
Missing Anti-clickjacking Header	Moyen	7 (50,0 %)
Vulnerable JS Library	Moyen	1 (7,1 %)
Big Redirect Detected (Potential Sensitive Information Leak)	Faible	2 (14,3 %)
Cookie without SameSite Attribute	Faible	1 (7,1 %)
Server Leaks Version Information via "Server" HTTP Response Header Field	Faible	4 (28,6 %)
Timestamp Disclosure - Unix	Faible	2 (14,3 %)
X-Content-Type-Options Header Missing	Faible	11 (78,6 %)
Authentication Request Identified	Pour information	1 (7,1 %)
Information Disclosure - Suspicious Comments	Pour information	1 (7,1 %)
Session Management Response Identified	Pour information	2 (14,3 %)
User Agent Fuzzer	Pour information	69 (492,9 %)
User Controllable HTML Element Attribute (Potential XSS)	Pour information	2 (14,3 %)
Total		14

Figure 4. Résultat de test d’intrusion de ZAP pour Vulnerable SAML App

Rapport d'Analyse de Sécurité de Vulnerable SAML App

Date de Génération: jeu. 4 avr. 2024
Version ZAP: 2.14.0
Sites Web Testés: http://127.0.0.1:8000

Identification des Vulnérabilités

Les vulnérabilités détectées par ZAP incluent certaines énumérées dans l'OWASP Top 10. Voici les plus remarquables [1][2][3][4] :

Tableau 6. Vulnérabilités remarquables détectées par ZAP pour Vulnerable SAML App

Vulnérabilité	Risque	Impact Potentiel
Absence de Jetons Anti-CSRF	Moyen	Risque de Cross-Site Request Forgery, permettant à des sites malveillants de réaliser des actions indésirables en utilisant les droits de l'utilisateur authentifié.
Content Security Policy (CSP) Header Not Set	Moyen	Absence de CSP qui peut augmenter le risque d'attaques XSS et de données injectées.
Missing Anti-clickjacking Header	Moyen	Absence de CSP qui peut augmenter le risque d'attaques XSS et de données injectées.
Server Leaks Version Information via "Server" HTTP Response Header Field	Faible	Révèle des informations sur la version du serveur qui pourraient aider les attaquants à exploiter des vulnérabilités spécifiques
X-Content-Type-Options Header Missing	Faible	Risque de MIME type sniffing où le navigateur peut interpréter incorrectement les

		types de contenu, menant à des attaques XSS.

Description des vulnérabilités

Dans cette partie du rapport, nous abordons les diverses vulnérabilités mises en évidence par ZAP au cours des tests d'intrusion. Notre système VulnerableSAMApp a révélé 14 vulnérabilités, détaillées dans le graphique susmentionné. Nous allons procéder à l'analyse et à la classification de ces vulnérabilités, en nous basant sur les normes établies par l'OWASP Top 10 pour les années 2021, 2017 et 2013.

1. Absence de Jetons Anti-CSRF (Risque Moyen)
   - OWASP Top 10 : Proche de 2013 (A8-Cross-Site Request Forgery (CSRF)).
   - Description : Sans jetons anti-CSRF, un attaquant pourrait réaliser une attaque de type Cross-Site Request Forgery (CSRF), en forçant un utilisateur authentifié à exécuter des actions indésirables sur votre site.
   - Recommandation : Implémentez des jetons CSRF dans toutes les formes et requêtes sensibles.

2. Content Security Policy (CSP) Header Not Set (Risque Moyen)
   - OWASP Top 10 : Associé à 2021 (A5:2021-Security Misconfiguration).
   - Description : L'absence de CSP augmente le risque d'attaques comme le Cross-Site Scripting (XSS), permettant à des scripts malveillants d'être injectés dans votre page web.
   - Recommandation : Définir une politique CSP stricte pour contrôler les ressources autorisées à s'exécuter ou à être chargées sur votre site web.

3. Missing Anti-clickjacking Header (Risque Moyen)
   - OWASP Top 10 : Non spécifiquement listé, mais relatif à la sécurité du client.
   - Description : Sans en-tête anti-clickjacking, votre site pourrait être vulnérable aux attaques de clickjacking, où un utilisateur est trompé pour cliquer sur un élément différent de celui qu'il voit.
   - Recommandation : Utilisation de l'en-tête 'X-Frame-Options' pour empêcher le site d'être embarqué dans des iframes de sites tiers.

4. Vulnerable JS Library (Risque Moyen)
   - OWASP Top 10 : Non spécifiquement listé, mais les vulnérabilités des composants avec des versions connues sont couvertes en 2021 (A6:2021-Vulnerable and Outdated Components).
   - Description : Utiliser une bibliothèque JavaScript vulnérable peut ouvrir des failles de sécurité permettant diverses attaques.
   - Recommandation : Mettre à jour les bibliothèques JavaScript à la dernière version sécurisée.

5. Big Redirect Detected (Potential Sensitive Information Leak) (Risque Faible)
- OWASP Top 10 : Non spécifiquement listé, mais peut être lié à A3:2021-Injection en 2021 pour les fuites d'informations potentielles.
- Description : De grandes redirections peuvent indiquer des fuites d'informations sensibles par des paramètres dans l'URL.
- Recommandation : Vérification et sécurisation des mécanismes de redirection pour éviter toute fuite d'informations.

6. Cookie without SameSite Attribute (Risque Faible)
- OWASP Top 10 : Relatif à A3:2021-Injection en 2021 pour la sécurité des témoins.
- Description : Les témoins sans l'attribut SameSite peuvent être envoyés lors de requêtes cross-site, ce qui peut entraîner des attaques CSRF.
- Recommandation : S'assurer que tous les témoins définissent l'attribut SameSite pour limiter leur envoi lors de requêtes cross-site.

7. Server Leaks Version Information via "Server" HTTP Response Header Field (Risque Faible)
- OWASP Top 10 : Associé à 2021 (A5:2021-Security Misconfiguration).
- Description : Si le serveur divulgue des informations de version, cela pourrait aider un attaquant à trouver des vulnérabilités spécifiques à la version utilisée.
- Recommandation : Configuration du serveur pour qu'il ne divulgue pas d'informations sur sa version dans les en-têtes HTTP.

8. Timestamp Disclosure - Unix (Risque Faible)
- OWASP Top 10 : Relatif à A5:2021-Security Misconfiguration.
- Description : La divulgation de timestamps Unix peut aider un attaquant à déterminer des informations sensibles liées au timing des processus du serveur.
- Recommandation : Éviter de révéler des timestamps Unix dans les réponses de votre serveur.

9. X-Content-Type-Options Header Missing (Risque Faible)
- OWASP Top 10 : Lié à la sécurité des clients, bien que non spécifiquement listé.
- Description : Sans cet en-tête, les navigateurs peuvent interpréter le contenu MIME d'une manière qui pourrait faciliter des attaques XSS.
- Recommandation : Utilisation de l'en-tête 'X-Content-Type-Options: nosniff' pour empêcher le navigateur d'interpréter différemment les types MIME.

10. Authentication Request Identified
- OWASP Top 10 : Non spécifiquement listé, mais relatif à A7:2021-Identification and Authentication Failures.
- Description : Cette alerte indique des requêtes liées à l'authentification, qui peuvent être normales, mais méritent une vérification.
- Recommandations : Revoir ces requêtes pour s'assurer qu'elles sont sécurisées et traitées correctement.

11. Information Disclosure - Suspicious Comments
- OWASP Top 10 : Non spécifiquement listé, mais peut être lié à A5:2021-Security Misconfiguration.
- Description : Des commentaires dans le code source peuvent révéler des informations sensibles.
- Recommandations : Passer en revue et supprimer les commentaires inutiles du code source déployé.

12. Session Management Response Identified
- OWASP Top 10 : Relatif à A7:2021-Identification and Authentication Failures.
- Description : Indique la gestion des sessions, pouvant révéler des pratiques de gestion des sessions.
- Recommandation : S'assurer que les méthodes de gestion des sessions sont sécurisées et respectent les meilleures pratiques.

13. User Agent Fuzzer
- OWASP Top 10 : Non spécifiquement listé, mais peut être lié à A5:2021-Security Misconfiguration pour les problèmes de configuration de sécurité.
- Description : Révèle comment votre application répond à diverses chaînes d'agents utilisateurs, pouvant indiquer des incohérences ou des vulnérabilités.
- Recommandation : Tester et assurer une réponse cohérente et sécurisée à une gamme d'agents utilisateurs.

14. User Controllable HTML Element Attribute (Potential XSS)
- OWASP Top 10 : Relatif à A3:2021-Injection pour les problèmes XSS.
- Description: Permet aux utilisateurs de contrôler les attributs des éléments HTML, risquant des attaques XSS.
- Recommandation: Sanitiser et valider toutes les entrées utilisateur qui peuvent contrôler les attributs HTML.

Comparaison des résultats entre SonarCloud et ZAP de Vulnerable SAML App

Pour comparer les résultats de SonarCloud à ceux de Zap, on a tout d'abord trouvé des correspondances entre les vulnérabilités de SonarCould et les vulnérabilités qui sont dans le catalogue CWE. Ces correspondances sont résumées dans le tableau ci-dessous :

Tableau 7: Vulnérabilités de SonarCloud pour Vulnerable SAML App

Vulnérabilités SonarCloud (tableau X)		Catalogue CWE	
id	Description	CWE	Relation OWASP
1	Une clef secrète est présente dans un fichier, accompagnée d'un identificateur "SECRET_KEY".	CWE-256: Plaintext Storage of a Password [16]	OWASP Top Ten 2021 Category A04:2021 - Insecure Design [16]
2	Le code désactive les protections contre les attaques CSRF.	CWE-352: Cross-Site Request Forgery (CSRF) [17]	OWASP Top Ten 2021 Category A01:2021 - Broken Access Control [17]
3	Le code autorise à la fois les requêtes HTTP sécuritaire et non sécuritaire.	CWE-319: Cleartext Transmission of Sensitive Information [18]	OWASP Top Ten 2021 Category A02:2021 - Cryptographic Failures [18]
4	Le docker file contient une instruction de copie récursive qui pourrait engendrer l'exposition de données sensibles de certains utilisateurs.	CWE-200: Exposure of Sensitive Information to an Unauthorized Actor [19]	OWASP Top Ten 2021 Category A01:2021 - Broken Access Control [19]
5	L'image Ubuntu spécifiée roule avec les permissions "root".	CWE-250: Execution with Unnecessary Privileges [20]	OWASP Top Ten 2021 Category A06:2021 - Security Misconfiguration [20]
6	Un générateur de séquence de lettres aléatoires à taille fixe est utilisé pour générer des clefs.	CWE-338: Use of Cryptographically Weak Pseudo-Random Number Generator (PRNG) [21]	OWASP Top Ten 2021 Category A02:2021 - Cryptographic Failures [21]

7	L’application utilise à plusieurs reprises le protocole HTTP au lieu du protocole HTTPS. Ce dernier est déprécié, car il ne crypte pas les paquets lors de la communication, rendant ainsi possible l’interception de ces derniers.	CWE-319: Cleartext Transmission of Sensitive Information [22]	OWASP Top Ten 2021 Category A02:2021 - Cryptographic Failures [22]
8	Du code de production fait appel à un mode debug.	CWE-489: Active Debug Code [23]	OWASP Top Ten 10 Category A10:2004 - Insecure Configuration Management [23]
9	L’application utilise à plusieurs reprises le protocole HTTP au lieu du protocole HTTPS. Ce dernier est déprécié, car il ne crypte pas les paquets lors de la communication, rendant ainsi possible l’interception de ces derniers.	CWE-319: Cleartext Transmission of Sensitive Information [24]	OWASP Top Ten 2021 Category A02:2021 - Cryptographic Failures [24]
10	L’image Ubuntu spécifiée roule avec les permissions “root”.	CWE-250: Execution with Unnecessary Privileges [25]	OWASP Top Ten 2021 Category A06:2021 - Security Misconfiguration [25]

Ensuite, on a comparé les vulnérabilités du catalogue CWE identifiées à celles identifiées par ZAP. Les vulnérabilités de ZAP qui ont une correspondance avec celles de SonarCloud sont énumérés ci-dessous :

Tableau 8. Vulnérabilités identifiées par ZAP et SonarCloud pour Vulnerable SAML App

Vulnérabilités SonarCloud	Vulnérabilités Zap		
id	no.	Description	Relation OWASP
2	1	Sans jetons anti-CSRF, un attaquant pourrait réaliser une attaque de type Cross-Site Request Forgery (CSRF), en forçant un utilisateur authentifié à exécuter des actions indésirables sur votre site.	OWASP Top Ten 2021 Category A01:2021 - Broken Access Control

Dans ce tableau, on a établi que la 2e vulnérabilité de SonarCloud a été identifiée par ZAP et elle se réfère à la première vulnérabilité identifiée par ZAP.

La 2e vulnérabilité identifiée par SonarCloud a un niveau de criticité ‘élevé’ tandis que la première vulnérabilité ZAP qui lui attribue le niveau de risque ‘Moyen’. Le niveau de criticité n’est donc pas le même. Aussi, en comparant le tableau 7 et le tableau 8, on remarque que les vulnérabilités communes ont la même relation OWASP.

À partir de ces résultats, on arrive à la même conclusion qu’avec l’analyse de la première application. En effet, la majorité des vulnérabilités identifiées par SonarCloud ne sont pas détectées par Zap et vice-versa.

Références

[1] OWASP. (2021) OWASP Top 10 - 2021, Open Web Application Security Project. [En ligne]. Disponible: https://owasp.org/Top10/

[2] Splunk. (2023) OWASP Explained: Today's OWASP Top 10. [En ligne]. Disponible: https://www.splunk.com/en_us/blog/learn/owasp-top-10.html

[3] Indusface. (2013) OWASP Top 10 Vulnerabilities 2013. [En ligne]. Disponible: https://www.indusface.com/blog/owasp-top-10-vulnerabilities-2013/

[4] OWASP. (2017) OWASP Top Ten 2017, Open Web Application Security Project. [En ligne]. Disponible: https://owasp.org/www-project-top-ten/2017/Top_10

[5] Common Weakness Enumeration. (2024) CWE-256: Plaintext Storage of a Password. [En ligne]. Disponible : https://cwe.mitre.org/data/definitions/256.html

[6] Common Weakness Enumeration. (2024) CWE-338: Use of Cryptographically Weak Pseudo-Random Number Generator (PRNG). [En ligne]. Disponible : https://cwe.mitre.org/data/definitions/338.html

[7] Common Weakness Enumeration. (2024) CWE-319: Cleartext Transmission of Sensitive Information. [En ligne]. Disponible : https://cwe.mitre.org/data/definitions/319.html

[8] Common Weakness Enumeration. (2024) CWE-20: Improper Input Validation. [En ligne]. Disponible : https://cwe.mitre.org/data/definitions/20.html

[9] Common Weakness Enumeration. (2024) CWE-502: Deserialization of Untrusted Data. [En ligne]. Disponible : https://cwe.mitre.org/data/definitions/502.html

[10] Common Weakness Enumeration. (2024) CWE-73: External Control of File Name or Path. [En ligne]. Disponible : https://cwe.mitre.org/data/definitions/73.html

[11] Common Weakness Enumeration. (2024) CWE-611: Improper Restriction of XML External Entity Reference. [En ligne]. Disponible : https://cwe.mitre.org/data/definitions/611.html

[12] Common Weakness Enumeration. (2024) CWE-112: Missing XML Validation. [En ligne]. Disponible : https://cwe.mitre.org/data/definitions/112.html

[13] Common Weakness Enumeration. (2024) CWE-601: URL Redirection to Untrusted Site ('Open Redirect'). [En ligne]. Disponible : https://cwe.mitre.org/data/definitions/601.html

[14] Common Weakness Enumeration. (2024) CWE-79: Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting').
[En ligne]. Disponible : https://cwe.mitre.org/data/definitions/79.html

[15] SolDevelo. (2023) SAST/DAST: Bridging Security Gap. [En ligne]. Disponible: https://soldevelo.com/blog/sast-dast-security-testing/

[16] Common Weakness Enumeration. (2024) CWE-256: Plaintext Storage of a Password.
[En ligne]. Disponible : https://cwe.mitre.org/data/definitions/256.html

[17] Common Weakness Enumeration. (2024) CWE-352: Cross-Site Request Forgery (CSRF).
[En ligne]. Disponible : https://cwe.mitre.org/data/definitions/352.html

[18] Common Weakness Enumeration. (2024) CWE-319: Cleartext Transmission of Sensitive Information.
[En ligne]. Disponible : https://cwe.mitre.org/data/definitions/319.html

[19] Common Weakness Enumeration. (2024) CWE-200: Exposure of Sensitive Information to an Unauthorized Actor.
[En ligne]. Disponible : https://cwe.mitre.org/data/definitions/200.html

[20] Common Weakness Enumeration. (2024) CWE-250: Execution with Unnecessary Privileges.
[En ligne]. Disponible : https://cwe.mitre.org/data/definitions/250.html

[21] Common Weakness Enumeration. (2024) CWE-338: Use of Cryptographically Weak Pseudo-Random Number Generator (PRNG).
[En ligne]. Disponible : https://cwe.mitre.org/data/definitions/338.html

[22] Common Weakness Enumeration. (2024) CWE-319: Cleartext Transmission of Sensitive Information.
[En ligne]. Disponible : https://cwe.mitre.org/data/definitions/319.html

[23] Common Weakness Enumeration. (2024) CWE-489: Active Debug Code.
[En ligne]. Disponible : https://cwe.mitre.org/data/definitions/489.html

[24] Common Weakness Enumeration. (2024) CWE-319: Cleartext Transmission of Sensitive Information.
[En ligne]. Disponible : https://cwe.mitre.org/data/definitions/319.html

[25] Common Weakness Enumeration. (2024) CWE-250: Execution with Unnecessary Privileges.
[En ligne]. Disponible : https://cwe.mitre.org/data/definitions/250.html