PROJET Mastère ESI

Cybersécurité

SUP DE VINCI

Table des matières

I. INTRODUCTION

RÉSUMÉ DU PROJET

L'objet de ce projet est de développer une toolbox automatisée conçue pour simplifier et optimiser entre autres le processus de réalisation de tests d'intrusion pour «Le partenaire», une société de cybersécurité spécialisée dans les tests d'intrusion pour les entreprises et les organisations gouvernementales.

Objectifs principaux :

1. Automatisation des tâches : Créer des scripts et des modules pour automatiser les différentes étapes d'un test d'intrusion, de la découverte initiale à l'exploitation des vulnérabilités.
2. Amélioration de l'efficacité : Réduire significativement le temps nécessaire pour effectuer des tests d'intrusion et aux complets sans compromettre la qualité des résultats.
3. Intégration d'outils existants : Incorporer et adapter des outils open-source reconnus dans le domaine de la cybersécurité pour enrichir les fonctionnalités de la toolbox.
4. Développement de fonctionnalités spécifiques : Créer des modules sur mesure pour répondre aux besoins particuliers de «Le partenaire» et de ses clients.
5. Reporting automatisé : Générer des rapports détaillés et personnalisables sur les vulnérabilités découvertes et les recommandations de sécurité.
6. Interface utilisateur intuitive : Concevoir une interface permettant aux professionnels de la sécurité de configurer et d'exécuter facilement des tests d'intrusion complexes.
7. Évolutivité : Assurer que la toolbox soit modulaire et facilement extensible pour intégrer de nouvelles fonctionnalités à l'avenir.

Résultats attendus par «Le partenaire» :

- Réduction d'au moins 40% du temps nécessaire pour réaliser un test d'intrusion complet par un rapport à un test manuel.
- Augmentation de la couverture des tests en automatisant la découverte et l'analyse des vulnérabilités.
- Standardisation des processus de test d'intrusion au sein de «Le partenaire».
- Amélioration de la qualité et de la cohérence des rapports de tests d'intrusion.
- Capacité accrue à gérer simultanément plusieurs projets de tests d'intrusion.

Ce projet transformera la manière dont «Le partenaire» réalise ses tests d'intrusion, en fournissant une solution automatisée, efficace et évolutive. La toolbox permettra à «Le partenaire» de répondre plus rapidement aux demandes croissantes de ses clients, tout en maintenant un haut niveau de qualité dans ses services de cybersécurité. Elle contribuera également à renforcer la position de «Le partenaire» comme leader dans le domaine des tests d'intrusion et de la sécurité informatique.

Aspect	Description
Objectif principal	Développer une toolbox automatisée pour optimiser les tests d'intrusion chez «Le partenaire»
Entreprise	«Le partenaire», spécialisée en cybersécurité et tests d'intrusion
Défis actuels	- Délais serrés - Charge de travail importante - Demande croissante des clients
Technologies clés	- Python - Outils open-source de cybersécurité - Bibliothèques de visualisation des résultats
Fonctionnalités principales	- Automatisation des tâches de test d'intrusion - Découverte et analyse de vulnérabilités - Génération de rapports détaillés
Bénéfices attendus	- Réduction de 40% du temps pour réaliser les tests - Amélioration de la qualité et cohérence des rapports - Augmentation de la couverture des tests - Standardisation des processus
Impact sur l'entreprise	- Amélioration de la satisfaction client - Renforcement de la réputation dans le domaine - Capacité accrue à gérer plusieurs projets simultanément
Approche	- Conception modulaire et évolutive - Intégration d'outils open-source existants - Interface utilisateur intuitive

II. DESCRIPTION DU PROJET

FONCTIONNEMENT GÉNÉRAL :

Dans le cadre de sa stratégie de cybersécurité, la société «Le partenaire» souhaite développer une toolbox automatisée pour simplifier et optimiser le processus de réalisation de tests d'intrusion. Ce projet vise à établir une solution technologique efficace permettant d'identifier les vulnérabilités des systèmes informatiques et des réseaux, tout en assurant une qualité et une rapidité accrues dans l'exécution des tests.

La toolbox sera conçue pour fonctionner de manière autonome et automatisée, intégrant des outils open-source reconnus dans le domaine de la cybersécurité. Elle permettra aux professionnels de la sécurité d'effectuer des tests d'intrusion complets, allant de la découverte initiale à l'exploitation des vulnérabilités, tout en générant des rapports détaillés sur les résultats.

Principales améliorations et innovations :

1. **Configuration** : L'utilisateur configurera la toolbox pour répondre aux besoins spécifiques du test d'intrusion, en choisissant les cibles et les types de tests à réaliser.
2. **Exploration** : La toolbox explorera le système cible pour identifier les ports ouverts, les services en cours d'exécution et les vulnérabilités potentielles.
3. **Analyse** : La toolbox effectuera une analyse approfondie des vulnérabilités identifiées, en utilisant des techniques avancées d'analyse et des bases de données de vulnérabilités connues.
4. **Exploitation** : La toolbox tentera d'exploiter les vulnérabilités identifiées pour évaluer l'impact potentiel sur la sécurité du système cible.
5. **Reporting** : La toolbox produira des rapports détaillés sur les résultats des tests d'intrusion, incluant les vulnérabilités découvertes et des recommandations pour leur correction.
6. **Interface utilisateur intuitive** : Une interface graphique conviviale sera développée pour faciliter l'utilisation de la toolbox par les professionnels de la sécurité, même ceux ayant peu d'expérience avec les tests d'intrusion.
7. **Évolutivité** : La toolbox sera conçue de manière modulaire, permettant l'ajout futur de nouvelles fonctionnalités et l'intégration d'outils supplémentaires selon l'évolution des besoins de «Le partenaire» et des menaces émergentes.

FONCTIONS PRINCIPALES

- **Automatisation des processus** : Grâce à l'automatisation, la toolbox permettra d'accélérer considérablement le processus de test d'intrusion, réduisant ainsi le temps nécessaire pour réaliser un test complet.
- **Intégration d'outils open-source** : L'utilisation d'outils open-source reconnus garantira que la toolbox est à jour avec les dernières techniques et méthodes en matière de cybersécurité.
- **Reporting automatisé** : Les rapports générés par la toolbox seront personnalisables, permettant aux utilisateurs de présenter les résultats sous différents formats adaptés aux besoins spécifiques des clients.
- **Optimisation continue** : En intégrant un système de feedback basé sur les résultats des tests précédents, la toolbox pourra évoluer pour améliorer continuellement ses performances et ses capacités analytiques.

IMPACT SUR «LE PARTENAIRE» :

Ce projet transformera la manière dont «Le partenaire» réalise ses tests d'intrusion, fournissant une solution automatisée, efficace et évolutive. La toolbox permettra à «Le partenaire» de répondre plus rapidement aux demandes croissantes de ses clients tout en maintenant un haut niveau de qualité dans ses services. En renforçant sa capacité à identifier et corriger rapidement les vulnérabilités, «Le partenaire» consolidera sa position comme leader dans le domaine des tests d'intrusion et de la sécurité informatique. Cette version étayée fournit une description claire et précise du projet, mettant en avant son fonctionnement général ainsi que ses objectifs et impacts attendus pour «Le partenaire».

III. L'ORGANISATION DE LA SOCIÉTÉ

L'ORGANISATION DE LA SOCIÉTÉ

«Le partenaire» est organisée en plusieurs pôles opérationnels, chacun avec des besoins spécifiques en matière de tests d'intrusion et de sécurité.

La toolbox automatisée devra répondre aux exigences de chaque département. Voici les besoins en tests d'intrusion pour chaque pôle, accompagnés d'exemples d'outils qui pourraient être intégrés dans la toolbox :

1. **Pôle Sécurité (SOC, EDR, XDR)**
   Ce pôle fournit des services de sécurité tels que le SOC, l'EDR, et l'XDR. La toolbox doit permettre des tests approfondis sur ces systèmes de sécurité.
2. **Pôle Développement SaaS**
   Ce département développe la solution SaaS de gestion des infrastructures de sécurité. La toolbox doit inclure des fonctionnalités pour tester la sécurité des applications web et des API.
3. **Pôle Infrastructure**
   Ce pôle gère l'ensemble des ressources matérielles et logicielles. La toolbox doit permettre des tests d'intrusion sur l'infrastructure réseau et les systèmes.
4. **Pôle Support Client et Assistance**
   Ce pôle est en charge de l'assistance technique. La toolbox doit inclure des fonctionnalités pour tester la sécurité des outils de support et de communication.
5. **Pôle Ressources Humaines et Administration**
   Ce pôle gère les ressources humaines et l'administration. La toolbox doit permettre des tests sur les systèmes de gestion interne et les applications administratives.

La toolbox automatisée devra être conçue de manière modulaire pour intégrer ces différents outils et répondre aux besoins spécifiques de chaque pôle de «Le partenaire», tout en permettant une utilisation cohérente et efficace à travers l'ensemble de l'organisation.

```
[Toolbox Automatisée pour Tests d'Intrusion - «Le partenaire»]
|
|--- [Pôle Sécurité (SOC, EDR, XDR)]
|    |-- Nmap (découverte réseau et analyse de ports)
|    |-- Metasploit Framework (tests d'exploitation)
|    |-- Wireshark (analyse du trafic réseau)
|    |-- OWASP ZAP (tests de sécurité des applications web)
|
|--- [Pôle Développement SaaS]
|    |-- Burp Suite (tests de sécurité des applications web)
|    |-- Postman (tests d'API)
|    |-- SQLmap (détection d'injections SQL)
|    |-- OWASP Dependency-Check (analyse des dépendances)
|
|--- [Pôle Infrastructure]
|    |-- OpenVAS (scans de vulnérabilités)
```

```
|    |-- Nessus (évaluation des vulnérabilités)
|    |-- Hydra (tests de force brute)
|    |-- Aircrack-ng (tests de sécurité Wi-Fi)
|
|--- [Pôle Support Client et Assistance]
|    |-- Nikto (analyse des serveurs web)
|    |-- SSLyze (analyse de la configuration SSL/TLS)
|    |-- Ettercap (tests de sécurité réseau)
|    |-- Maltego (collecte d'informations et analyse de données)
|
|--- [Pôle Ressources Humaines et Administration]
     |-- John the Ripper (tests de force brute sur les mots de passe)
     |-- Cain & Abel (analyse de la sécurité des protocoles réseau)
     |-- Acunetix (scans de vulnérabilités web)
     |-- Nmap Scripting Engine (NSE) (scripts de test personnalisés)
```

IV. LA SOLUTION TECHNIQUE CONSEILLÉE

La solution technique proposée pour la toolbox automatisée de tests d'intrusion du client est conçue pour optimiser et sécuriser le processus de réalisation des tests d'intrusion, tout en assurant l'efficacité, la scalabilité et la flexibilité. Elle repose sur une architecture modulaire et hautement automatisée, structurée autour des composants suivants :

1. **Architecture modulaire basée sur Python**
   La toolbox sera développée en utilisant Python comme langage principal, offrant une grande flexibilité et une vaste bibliothèque de modules de cybersécurité. Flask sera utilisé comme framework pour créer une API RESTful, permettant une communication efficace entre les différents modules. Poetry sera employé pour gérer les dépendances et l'environnement virtuel, assurant ainsi une cohérence et une reproductibilité du projet.
2. **Modules d'automatisation des tâches**
   Plusieurs modules spécialisés seront développés pour automatiser les différentes étapes d'un test d'intrusion :
   - Module de découverte : Intégration de Nmap pour le scanning de réseau et l'énumération des services.
   - Module d'analyse de vulnérabilités : Utilisation d'OpenVAS ou Nessus via leurs API pour effectuer des scans de vulnérabilités approfondis.
   - Module d'exploitation : Intégration de Metasploit Framework via son API Ruby pour l'exploitation automatisée des vulnérabilités découvertes.
   - Module de post-exploitation : Développement de scripts personnalisés pour la collecte d'informations post-exploitation.
3. **Intégration d'outils open-source**
   La toolbox intégrera des outils open-source reconnus dans le domaine de la cybersécurité :
   - OWASP ZAP pour les tests de sécurité des applications web.
   - SQLmap pour la détection et l'exploitation des injections SQL.
   - Hydra pour les attaques par force brute sur divers protocoles.

- Aircrack-ng pour les tests de sécurité des réseaux Wi-Fi.

4. **Base de données et stockage**
   PostgreSQL sera utilisé comme base de données principale pour stocker les résultats des scans, les configurations et les rapports. MinIO, compatible avec l'API S3, sera employé pour le stockage des fichiers volumineux tels que les preuves et les captures d'écran.
5. **Reporting automatisé**
   Le système de reporting utilisera Jinja2 pour générer des rapports personnalisables. Les rapports seront exportables en formats PDF, HTML et CSV. D3.js sera intégré pour créer des graphiques interactifs dans les rapports, offrant une visualisation claire des résultats.
6. **Gestion des workflows et orchestration**
   Celery sera utilisé pour la gestion des tâches asynchrones et la parallélisation des scans, permettant une exécution efficace des tests d'intrusion. Redis servira de broker de messages pour Celery, assurant une communication rapide entre les composants.
7. **Sécurité de la toolbox**
   La sécurité de la toolbox elle-même sera assurée par :
   - Le chiffrement des données sensibles au repos avec Fernet.
   - La mise en place de HTTPS pour toutes les communications.
   - L'implémentation de contrôles d'accès basés sur les rôles (RBAC).
8. **Conteneurisation et déploiement**
   Docker sera utilisé pour la conteneurisation de la toolbox et de ses dépendances, assurant une portabilité et une reproductibilité de l'environnement. Docker Compose orchestrera le déploiement multi-conteneurs.
9. **Évolutivité et extensibilité**
   Une architecture de plugins sera mise en place pour faciliter l'ajout de nouveaux modules et fonctionnalités. L'API sera documentée avec Swagger, permettant une intégration facile avec d'autres outils ou systèmes.
10. **Module d'investigation forensique automatisée**
    Ce module spécialisé comprendra :
    - Des outils d'analyse statique comme strings, readelf, et UPX detector.
    - L'intégration de Cuckoo Sandbox pour l'analyse dynamique des binaires.
    - L'utilisation de Wireshark pour la capture et l'analyse du trafic réseau.
    - L'intégration de ClamAV et de l'API VirusTotal pour la détection de malwares.

Cette architecture modulaire et extensible permettra à «Le partenaire» de disposer d'une toolbox de tests d'intrusion puissante, automatisée et évolutive, capable de s'adapter aux futurs besoins en matière de cybersécurité.

Composant	Description
Architecture modulaire	- Python 3.x comme langage principal - Flask pour créer une API RESTful - Poetry pour la gestion des dépendances
Modules d'automatisation	- Nmap pour le scanning de réseau - OpenVAS ou Nessus pour les scans de vulnérabilités - Metasploit Framework pour l'exploitation automatisée - Scripts personnalisés pour la post-exploitation
Intégration d'outils open-source	- OWASP ZAP pour les tests d'applications web - SQLmap pour la détection d'injections SQL - Hydra pour les attaques par force brute - Aircrack-ng pour les tests Wi-Fi
Base de données et stockage	- PostgreSQL pour stocker les résultats et configurations - MinIO (compatible S3) pour stocker les preuves et captures
Reporting automatisé	- Jinja2 pour les templates de rapports personnalisables - Export en formats PDF, HTML et CSV - D3.js pour les graphiques interactifs
Gestion des workflows	- Celery pour les tâches asynchrones et la parallélisation - Redis comme broker de messages
Sécurité de la toolbox	- Chiffrement des données sensibles avec Fernet - HTTPS pour toutes les communications - Contrôles d'accès basés sur les rôles (RBAC)
Conteneurisation et déploiement	- Docker pour la conteneurisation - Docker Compose pour l'orchestration multi-conteneurs - GitLab CI/CD pour l'intégration et le déploiement continus
Évolutivité et extensibilité	- Architecture de plugins pour l'ajout de nouveaux modules - API documentée avec Swagger
Investigation forensique de binaires	- YARA pour la détection de patterns - Radare2 pour l'analyse statique - Cuckoo Sandbox pour l'analyse dynamique - ExifTool pour l'extraction de métadonnées
Analyse forensique avancée	- Extraction de chaînes avec strings - Analyse des en-têtes avec readelf ou objdump

Composant	Description
	- Détection de packing avec UPX detector - Exécution contrôlée dans Cuckoo Sandbox - Capture et analyse du trafic réseau avec Wireshark - Monitoring des appels système avec strace - Scan antivirus avec ClamAV - Comparaison avec des bases de signatures (VirusTotal API)

Conseil :

- Sécurité renforcée : Mettez davantage l'accent sur les aspects de sécurité de la toolbox elle-même. Considérez l'ajout de techniques comme l'obfuscation du code, la détection de tampering, et des mécanismes de protection contre les attaques par injection.
- Conformité légale : Incluez une section sur la conformité légale et éthique, expliquant comment la toolbox respecte les réglementations en vigueur (RGPD, etc.) et les bonnes pratiques éthiques en matière de tests d'intrusion.
- Gestion des mises à jour : Ajoutez des détails sur la stratégie de mise à jour des composants de la toolbox, en particulier pour les outils open-source intégrés.
- Personnalisation client : Expliquez comment la toolbox pourra être adaptée aux besoins spécifiques de différents clients de «Le partenaire».
- Intégration avec l'existant : Détaillez comment la toolbox s'intégrera avec les systèmes et processus existants de «Le partenaire».
- Performance et optimisation : Incluez des informations sur les stratégies d'optimisation des performances, notamment pour les scans à grande échelle.
- Formation et support : Ajoutez une section sur la formation des utilisateurs et le support technique pour la toolbox.
- Métriques et KPIs : Définissez des métriques clés pour évaluer l'efficacité et la performance de la toolbox.
- Plan de continuité : Expliquez comment la toolbox gérera les interruptions ou les pannes potentielles.
- Évolutivité future : Élaborez davantage sur la stratégie à long terme pour l'évolution de la toolbox, y compris l'intégration potentielle de technologies émergentes comme l'IA/ML pour l'analyse des vulnérabilités.

V. PREREQUIS DU PROJET

L'entreprise retenue pour ce projet devra fournir les prestations détaillées ci-dessous, afin d'assurer le développement optimal de la toolbox automatisée pour les tests d'intrusion de «Le partenaire» :

1. Étude détaillée des besoins et de l'architecture de la toolbox L'étude devra inclure une analyse complète de l'architecture technique proposée, englobant :

    - La conception modulaire de la toolbox et l'intégration des différents composants;
    - Les méthodes d'automatisation des tâches de test d'intrusion ;
    - Les solutions pour l'analyse de vulnérabilités et l'exploitation automatisée ;
    - Les mesures de sécurité pour la toolbox elle-même (chiffrement, contrôle d'accès).

2. Conditions d'utilisation et gestion de la toolbox. L'entreprise devra définir les conditions d'utilisation pour garantir l'efficacité et la fiabilité, incluant :

    - Les bonnes pratiques et normes à respecter pour l'utilisation de la toolbox
    - Les procédures de mise à jour et de maintenance de la toolbox
    - La gestion des rapports générés et le stockage sécurisé des résultats des tests
    - La formation des utilisateurs et la documentation d'utilisation.

3. Planning prévisionnel du développement Le planning prévisionnel devra inclure toutes les étapes du projet, comprenant :

    - Les phases de conception et de validation de l'architecture de la toolbox
    - Le développement des différents modules et leur intégration
    - Les phases de tests unitaires, d'intégration et de validation globale
    - Le déploiement de la toolbox dans l'environnement de «Le partenaire»
    - La formation des équipes et le suivi post-déploiement.

4. Plan de test et validation des performances. Le plan de test devra inclure :

    - Des tests fonctionnels pour chaque module de la toolbox
    - Des tests de performance pour évaluer l'efficacité de l'automatisation
    - Des tests de sécurité pour s'assurer que la toolbox elle-même est sécurisée
    - Une période de test en conditions réelles avec les équipes de «Le partenaire».

- Étude détaillée de l'architecture : Analyse de la conception modulaire, des méthodes d'automatisation, des solutions d'analyse de vulnérabilités et des mesures de sécurité pour la toolbox.
- Conditions d'utilisation : Définition des bonnes pratiques, procédures de maintenance, gestion des rapports et formation des utilisateurs.
- Planning prévisionnel : Phases de conception, développement, tests, déploiement et formation.
- Plan de test : Tests fonctionnels, de performance, de sécurité et période d'essai en conditions réelles.

VI. CONSEIL D'ORGANISATION DE PROJET

ETAPE CLE CONSEILLÉE

Étape	Description	Livrables
1. Étude détaillée des besoins	- Analyse des exigences de «Le partenaire» - Définition des fonctionnalités de la toolbox - Identification des outils open-source à intégrer	- Document de spécifications fonctionnelles - Liste des outils à intégrer
2. Architecture de la toolbox	- Conception de l'architecture modulaire - Définition des interfaces entre modules - Planification de l'intégration des outils	- Schéma d'architecture - Document de conception technique

Étape	Description	Livrables
3. Développement du core	- Implémentation du framework de base - Développement des modules principaux - Intégration des outils open-source	- Code source du core - Documentation technique
4. Modules de test d'intrusion	- Développement des modules de découverte - Implémentation des modules d'analyse de vulnérabilités - Création des modules d'exploitation	- Modules de test d'intrusion - Documentation des modules
5. Module forensique	- Développement des fonctionnalités d'analyse statique - Implémentation de l'analyse dynamique - Intégration des outils forensiques (ex: Volatility)	- Module forensique - Guide d'utilisation forensique
6. Automatisation et orchestration	- Développement des scripts d'automatisation - Mise en place des workflows de test	- Scripts d'automatisation - Documentation des workflows
7. Interface utilisateur	- Conception de l'interface web - Développement du frontend - Intégration avec le backend	- Interface utilisateur - Guide d'utilisation
8. Système de reporting	- Développement du module de génération de rapports - Création de templates de rapports - Intégration de visualisations (ex: D3.js)	- Module de reporting - Templates de rapports
9. Sécurisation de la toolbox	- Implémentation du chiffrement des données - Mise en place de l'authentification et des contrôles d'accès - Configuration HTTPS	- Documentation de sécurité - Rapport d'audit de sécurité
10. Tests et validation	- Élaboration du plan de tests - Exécution des tests unitaires et d'intégration - Tests de performance et de sécurité	- Plan de tests - Rapport de tests
11. Déploiement et conteneurisation	- Configuration de l'environnement Docker - Création des Dockerfiles - Mise en place de Docker Compose	- Fichiers Docker - Guide de déploiement
12. Documentation et formation	- Rédaction de la documentation technique - Création du manuel utilisateur - Préparation du matériel de formation	- Documentation complète - Manuel utilisateur - Support de formation

Méthodologie de travail conseillé Agile Scrum adapté:

1. Durée du sprint : 2 semaines
2. Événements Scrum :
   - Sprint Planning : Début de chaque sprint
   - Sprint Review : Fin de chaque sprint
   - Sprint Retrospective : Après chaque Sprint Review
3. Rôles Scrum :
   - Product Owner : Représentant de «Le partenaire»
   - Scrum Master : À désigner parmi les étudiants (rotation possible)
   - Development Team : Les 2 étudiants
4. Livrables intermédiaires par sprint :
   - Sprint 1 : Architecture globale et plan de développement
   - Sprint 2 : Développement du core et intégration des premiers outils open-source
   - Sprint 3 : Implémentation des modules de test d'intrusion de base
   - Sprint 4 : Développement du module forensique et automatisation des tâches

- Sprint 5 : Mise en place du système de reporting et de l'interface utilisateur
- Sprint 6 : Tests d'intégration, de sécurité et finalisation de la documentation

5. Outils de gestion de projet :
   - Jira ou Trello pour le suivi des tâches et du backlog
   - Git pour le versioning du code et de la documentation
   - Slack ou Microsoft Teams pour la communication d'équipe
   - Jenkins ou GitLab CI pour l'intégration continue et les tests automatisés
6. Pratiques de développement :
   - Code review : Chaque fonctionnalité doit être revue par l'autre membre de l'équipe
   - Test-Driven Development (TDD) : Écriture des tests avant l'implémentation
   - Pair programming : Pour les tâches complexes ou critiques
   - Documentation continue : Mise à jour de la documentation technique à chaque sprint
7. Gestion de la sécurité :
   - Analyse de code statique à chaque commit avec des outils comme SonarQube
   - Tests de pénétration réguliers sur la toolbox elle-même
   - Revue de sécurité à la fin de chaque sprint
8. Métriques de suivi :
   - Vélocité de l'équipe
   - Taux de couverture des tests
   - Nombre de vulnérabilités détectées et corrigées
   - Temps moyen pour compléter un test d'intrusion avec la toolbox

ETAPE PAR METIER CONSEILLEE

Rôle	Responsabilités principales	Livrables
Architecte Sécurité	- Conception de l'architecture globale de la toolbox - Définition des modules de test d'intrusion - Élaboration du workflow d'automatisation - Conception des mécanismes de sécurité de la toolbox	- Étude détaillée de l'architecture de la toolbox - Schémas d'architecture des modules - Documentation du workflow d'automatisation - Plan de sécurisation de la toolbox
Ingénieur Développement	- Développement des modules de test d'intrusion - Intégration des outils open-source - Implémentation de l'automatisation - Développement de l'interface utilisateur	- Code source des modules - Documentation d'intégration des outils - Scripts d'automatisation - Interface utilisateur fonctionnelle
Analyste Forensique	- Conception du module d'analyse forensique - Définition des procédures d'investigation automatisée - Intégration des outils d'analyse statique et dynamique - Développement des fonctionnalités de reporting forensique	- Module d'analyse forensique - Procédures d'investigation automatisée - Documentation des outils forensiques intégrés - Modèles de rapports forensiques
Chef de Projet (pour tous les étudiants)	- Élaboration du planning du projet - Coordination des phases de développement - Suivi de l'avancement et gestion des risques - Planification des tests et de la formation	- Planning détaillé du projet - Rapports d'avancement - Plan de gestion des risques - Programme de formation des utilisateurs
Ingénieur QA / Test	- Conception des plans de test - Vérification de la fiabilité des modules	- Plans de test détaillés - Rapports de tests fonctionnels

	- Évaluation de l'ergonomie de l'interface - Réalisation des tests de performance et de sécurité	- Évaluations d'ergonomie - Rapports de tests de performance et de sécurité
Analyse Financiére du Projet (pour tous les étudiants	- Estimation des coûts de développement - Calcul des coûts des licences logicielles - Évaluation du retour sur investissement - Analyse comparative des options technologiques	- Estimation financière détaillée - Analyse des coûts de licences - Analyse comparative des solutions

VII. EXEMPLE DE REPARTITION DES TACHES MINIMUN PAR ETUDIANT

Le tableau ci-dessous présente un backlog de 50 tâches qui serviront de base pour le projet de développement d'une toolbox automatisée pour les tests d'intrusion de «Le partenaire». Ce backlog est fourni à titre d'exemple pour orienter le travail des étudiants et illustrer l'étendue et la diversité des tâches à accomplir dans le cadre de ce projet de cybersécurité. Il couvre les différents aspects du développement de la toolbox, de l'architecture à l'implémentation des fonctionnalités de test d'intrusion, en passant par la sécurisation de l'outil lui-même et son intégration dans l'environnement de «Le partenaire».

ID	Catégorie	Tâche
1	Architecture	Concevoir l'architecture globale de la toolbox
2	Architecture	Définir les interfaces entre les différents modules
3	Architecture	Élaborer le schéma de flux de données
4	Architecture	Concevoir le système de stockage des résultats
5	Architecture	Définir la stratégie de sécurisation de la toolbox
6	Développement	Implémenter le module de découverte réseau
7	Développement	Développer le module de scan de ports
8	Développement	Créer le module d'énumération des services
9	Développement	Implémenter le module de détection de vulnérabilités
10	Développement	Développer le module d'exploitation des vulnérabilités
11	Développement	Créer le module de post-exploitation
12	Développement	Implémenter le système de reporting
13	Développement	Développer l'interface utilisateur

ID	Catégorie	Tâche
14	Développement	Créer le module de gestion des plugins
15	Développement	Implémenter le système de logging
16	Intégration	Intégrer Nmap pour la découverte réseau
17	Intégration	Intégrer Metasploit pour l'exploitation
18	Intégration	Intégrer OpenVAS pour le scan de vulnérabilités
19	Intégration	Intégrer Hydra pour les attaques par force brute
20	Intégration	Intégrer Wireshark pour l'analyse de trafic
21	Sécurité	Implémenter le chiffrement des données sensibles
22	Sécurité	Configurer l'authentification et l'autorisation des utilisateurs
23	Sécurité	Mettre en place un système de gestion des clés
24	Sécurité	Implémenter des mécanismes de détection d'intrusion
25	Sécurité	Configurer les pares-feux et les règles de filtrage
26	Tests	Développer des tests unitaires pour chaque module
27	Tests	Créer des scénarios de tests d'intégration
28	Tests	Mettre en place des tests de performance
29	Tests	Réaliser des tests de sécurité sur la toolbox
30	Tests	Effectuer des tests d'acceptation utilisateur
31	Documentation	Rédiger la documentation technique de la toolbox
32	Documentation	Créer un guide d'utilisation pour les pentesters
33	Documentation	Documenter les procédures d'installation et de configuration
34	Documentation	Élaborer un guide de dépannage
35	Documentation	Rédiger un manuel de bonnes pratiques pour les tests d'intrusion
36	Déploiement	Configurer l'environnement de production

ID	Catégorie	Tâche
37	Déploiement	Mettre en place un système de déploiement continu
38	Déploiement	Créer des scripts de déploiement automatisé
39	Déploiement	Configurer le monitoring de la toolbox en production
40	Déploiement	Mettre en place un système de sauvegarde et restauration
41	Forensique	Développer le module d'analyse forensique
42	Forensique	Intégrer des outils d'analyse de mémoire (ex: Volatility)
43	Forensique	Implémenter l'analyse de fichiers malveillants
44	Forensique	Créer un module de timeline des événements
45	Forensique	Intégrer des capacités d'analyse de trafic réseau
46	Optimisation	Optimiser les performances des scans de vulnérabilités
47	Optimisation	Améliorer l'efficacité des modules d'exploitation
48	Optimisation	Optimiser la génération de rapports
49	Optimisation	Implémenter le multithreading pour les tâches parallèles
50	Optimisation	Optimiser l'utilisation des ressources système

Répartition équitable des tâches :

Les 50 tâches listées dans ce backlog doivent être réparties de manière équitable entre les trois étudiants impliqués dans le projet. Chaque étudiant sera responsable d'environ 17 tâches, couvrant différentes catégories (Architecture, Développement, Intégration, Sécurité, Tests, et Documentation). Cette répartition équilibrée permettra à chaque étudiant de développer des compétences variées en cybersécurité tout en contribuant de manière significative au projet.

Les étudiants devront collaborer étroitement pour assurer la cohérence globale de la toolbox et partager leurs connaissances en matière de tests d'intrusion et d'automatisation. La répartition précise des tâches pourra être ajustée en fonction des compétences et des intérêts de chaque étudiant, tout en veillant à maintenir un équilibre dans la charge de travail et la complexité des tâches attribuées.

Il est important de noter que ce backlog est un point de départ et que les étudiants sont encouragés à l'adapter et à le compléter en fonction des besoins spécifiques du projet. L'objectif est de fournir une structure initiale tout en laissant la flexibilité nécessaire pour que les étudiants puissent démontrer leur créativité et leur capacité à résoudre des problèmes complexes dans le domaine de la cybersécurité et de l'automatisation des tests d'intrusion.

VIII. GLOSSAIRE DES TECHNOLOGIES

Partie	Technologies Recommandées	Détails Supplémentaires
Langage de programmation	- Python - Ruby - Go - Rust - JavaScript/Node.js	- Python : Langage principal, riche en bibliothèques de sécurité - Ruby : Pour l'intégration avec Metasploit - Go : Performance et concurrence - Rust : Sécurité mémoire et performance - JavaScript : Pour les modules web et l'interface utilisateur
Framework web	- Flask - Django - FastAPI - Ruby on Rails - Express.js	- Flask : Léger et flexible - Django : Robuste avec ORM intégré - FastAPI : Rapide et asynchrone - Ruby on Rails : Full-stack pour Ruby - Express.js : Minimaliste pour Node.js
Outils de scan réseau	- Nmap - Masscan - Zmap - Angry IP Scanner - Unicornscan	- Nmap : Polyvalent et extensible - Masscan : Scans rapides à grande échelle - Zmap : Scan de l'espace IP à haute vitesse - Angry IP Scanner : Interface graphique conviviale - Unicornscan : Scanner asynchrone
Analyse de vulnérabilités	- OpenVAS - Nessus - Nexpose - Qualys - Acunetix	- OpenVAS : Scanner open-source - Nessus : Solution commerciale populaire - Nexpose : Analyse avancée - Qualys : Solution cloud - Acunetix : Spécialisé dans les applications web
Exploitation	- Metasploit Framework - ExploitDB - Core Impact - Canvas - Cobalt Strike	- Metasploit : Framework d'exploitation le plus utilisé - ExploitDB : Base de données d'exploits - Core Impact : Plateforme commerciale avancée - Canvas : Framework d'exploitation alternatif - Cobalt Strike : Pour les tests d'intrusion avancés
Web Application Testing	- OWASP ZAP - Burp Suite - Nikto - w3af - Arachni	- OWASP ZAP : Outil open-source complet - Burp Suite : Suite commerciale populaire - Nikto : Scanner de serveur web - w3af : Framework de test d'applications web - Arachni : Scanner modulaire et distribué
Analyse forensique	- Volatility - Autopsy - The Sleuth Kit - EnCase - FTK (Forensic Toolkit)	- Volatility : Analyse de mémoire volatile - Autopsy : Plateforme d'analyse digitale - The Sleuth Kit : Outils pour l'analyse de systèmes de fichiers - EnCase : Solution commerciale complète - FTK : Suite d'outils forensiques avancée
Base de données	- PostgreSQL - MongoDB - SQLite - Elasticsearch - Neo4j	- PostgreSQL : Base relationnelle robuste - MongoDB : Base NoSQL flexible - SQLite : Base légère et embarquée - Elasticsearch : Recherche et analyse de données - Neo4j : Base de données graphe

Partie	Technologies Recommandées	Détails Supplémentaires
Reporting	- Jinja2 - ReportLab - Matplotlib - D3.js - Plotly	- Jinja2 : Moteur de template pour HTML - ReportLab : Génération de PDF - Matplotlib : Bibliothèque de visualisation Python - D3.js : Visualisations interactives en JavaScript - Plotly : Graphiques interactifs
Authentification	- JWT - OAuth 2.0 - LDAP - SAML - Keycloak	- JWT : Jetons d'authentification stateless - OAuth 2.0 : Protocole d'autorisation - LDAP : Protocole d'annuaire léger - SAML : Authentification fédérée - Keycloak : Gestion d'identité et d'accès
Conteneurisation	- Docker - Podman - LXC - rkt - containerd	- Docker : Plateforme de conteneurisation populaire - Podman : Alternative sans démon - LXC : Conteneurs au niveau du système - rkt : Runtime de conteneur sécurisé - containerd : Runtime de conteneur de l'OCI
Orchestration	- Kubernetes - Docker Swarm - Nomad - OpenShift - Rancher	- Kubernetes : Orchestrateur leader du marché - Docker Swarm : Orchestration native Docker - Nomad : Orchestrateur léger de HashiCorp - OpenShift : PaaS basé sur Kubernetes - Rancher : Plateforme de gestion de conteneurs
CI/CD	- GitLab CI - Jenkins - GitHub Actions - CircleCI - Travis CI	- GitLab CI : Intégré à GitLab - Jenkins : Serveur d'automatisation open-source - GitHub Actions : Intégré à GitHub - CircleCI : Plateforme cloud native - Travis CI : Service d'intégration continue
Gestion de configuration	- Ansible - Puppet - Chef - SaltStack - Terraform	- Ansible : Outil d'automatisation agentless - Puppet : Gestion de configuration à grande échelle - Chef : Automatisation d'infrastructure - SaltStack : Gestion de configuration événementielle - Terraform : Outil d'Infrastructure as Code

IX. CALENDRIER DES RENDU

Étape	Calendrier	Livrable attendu	Points clés	Évaluation	Ce qui doit être fait par rapport au projet
Analyse du sujet	N+1 mois	Questions d'éclaircissement au client via un PDF	Minimum 3 questions par étudiant, pour valider la compréhension du sujet.	50 % du Management	Identifier les zones d'incertitude, clarifier les attentes du client, et établir une base solide pour les livrables futurs.
Backlog & Timeline	N+2/3 mois	Document PDF comprenant backlog et timeline	Planification détaillée, timeline, ressources nécessaires, estimation des risques.	50 % du Management	Créer un backlog structuré et une timeline réaliste en priorisant les tâches, identifier les risques, et définir les ressources nécessaires.
Vidéo & MVP	N+6 mois	Soutenance orale avec démonstration vidéo du MVP	Présentation structurée : contexte, objectifs, démonstration, perspectives d'évolution.	30 % de la note du Bloc technique	Développer et finaliser un produit minimum viable, préparer une démonstration convaincante et créer une vidéo illustrant les fonctionnalités clés.
Document technique final	N+6 mois	Document technique détaillé	Introduction, architecture, fonctionnalités, tests, documentation utilisateur.	70 % de la note du Bloc technique	Documenter en détail tout le projet (architecture, tests, implémentation) et fournir un guide utilisateur complet et professionnel.

X. AIDE SUPLÉMENTAIRE

Les étudiants sont encouragés à s'inscrire au GitHub Student Pack pour bénéficier de ressources utiles à la réalisation de leur projet d'études. Pour accéder à ce pack, ils devront utiliser leur adresse e-mail académique fournie par l'école ainsi qu'un certificat de scolarité en cours de validité. Cette inscription leur permettra de bénéficier d'avantages dédiés aux étudiants, facilitant ainsi leur travail tout au long du projet. Il est recommandé d'effectuer cette démarche dès le début afin de tirer pleinement parti des outils mis à leur disposition.