مدرسة علوم المعلومات
ⵜⵉⵣⵔⴰⵡⵉⵏ ⵏ ⵜⵓⵙⵙⵏⵉ ⵏ ⵜⵣⵔⴼⵜ
ESI ECOLE DES SCIENCES DE L'INFORMATION

CBI producteur de productivité®

Rapport de Stage

2ème ACI

Annee Universitaire: 2024/2025

Comprendre et Simuler les Cyberattaques : de La Théorie (Cyber Kill Chain & MITRE ATT&CK) à La Pratique (Pentest)

Réalisé par **Salma HERMAK**

Encadré par **Mr. Driss BENATTOU**

Supervisé par **Mr. Yassine MEDOUAR**

Table des matières

Remerciements

C'est avec une gratitude profonde que je commence par exprimer ma reconnaissance envers Allah le Tout-Puissant pour m'avoir accordé la santé et la volonté nécessaires pour mener à bien ce projet, qui a insufflé en moi la foi, le courage et la détermination indispensables à la réussite de ce stage.

Mes remerciements vont à Monsieur Driss Benattou, mon encadrant de stage au sein de CBI, pour sa confiance, son encadrement technique et ses précieux conseils. Sa disponibilité, sa pédagogie et son approche professionnelle ont rendu cette expérience particulièrement enrichissante.

Je souhaite également exprimer ma gratitude envers Monsieur Yassine Medouar, consultant au sein de CBI, pour son accompagnement opérationnel, son expertise et ses conseils pratiques, qui ont grandement contribué à la réussite de ce projet.

Enfin, je tiens à remercier chaleureusement toute l'équipe du SOC de CBI, pour leur accueil, leur collaboration, ainsi que pour l'ambiance conviviale et motivante qui a marqué l'ensemble de mon parcours. Leur soutien, direct ou indirect, a été un atout majeur pour mon épanouissement professionnel et personnel

Liste des figures

Résumé

Dans le cadre de ce stage, nous avons mené un projet visant à explorer les principales étapes d'un cycle de cyberattaque ainsi que les méthodes de défense associées. La première phase a consisté en une étude documentaire approfondie de la Cyber Kill Chain et du cadre MITRE ATT&CK, qui offrent une structuration des techniques offensives et des mesures de détection. Ensuite, nous avons réalisé des présentations sur des concepts fondamentaux en sécurité informatique, notamment le DNS, le modèle OSI ainsi que les rôles respectifs de la Blue Team et de la Red Team.

Une phase pratique a suivi, avec l'étude de trois vulnérabilités issues de l'OWASP Top 10 et la mise en place de labs dédiés permettant leur exploitation contrôlée et l'analyse des impacts. Enfin, afin de consolider l'ensemble des acquis, nous avons conçu une machine d'entraînement (VM vulnérable) intégrant différents services et failles. Cette machine a permis de simuler des scénarios d'attaque complets, en suivant les étapes de la Cyber Kill Chain et en mappant chaque action aux techniques du MITRE ATT&CK.

1. Introduction

1.1 Contexte générale du stage

Dans le cadre de mon **Projet de Fin d'Année** (**PFA**), j'ai eu l'opportunité d'effectuer un stage d'une durée de deux mois, du 3 juillet au 29 août 2025, au sein de l'entreprise **CBI Maroc**, acteur majeur dans l'intégration de solutions IT et la cybersécurité.
Ce stage s'inscrit dans la continuité de ma formation académique et avait pour objectif principal de mettre en pratique les connaissances théoriques acquises, tout en développant de nouvelles compétences techniques et professionnelles.

Intégrée à l'équipe du **Security Operations Center** (**SOC**), j'ai été assignée à la cellule pentesting, où ma mission consistait à réaliser une simulation de test d'intrusion (**pentesting**) sur un environnement virtuel. Cette mission avait pour finalité d'identifier et d'exploiter des vulnérabilités critiques en se basant sur les standards de l'**OWASP Top 10**.

Le stage s'est déroulé en deux grandes phases complémentaires. La première, d'ordre documentaire, m'a permis d'étudier et d'approfondir des cadres méthodologiques tels que la **Cyber Kill Chain** et le **framework MITRE ATT&CK**, ainsi que de réviser les fondamentaux en cybersécurité, en réseaux et en systèmes. La seconde phase a été consacrée à l'application pratique, avec la préparation de l'environnement virtuel, l'exploitation des vulnérabilités sélectionnées et la rédaction d'un rapport technique détaillant les résultats et les recommandations correctives.

Ainsi, ce stage s'inscrit dans une démarche académique et professionnelle visant à :

- Développer une compréhension approfondie des **méthodes offensives** utilisées en cybersécurité,
- Expérimenter le processus complet d'un **test d'intrusion**, depuis la préparation documentaire jusqu'à l'exploitation des failles,
- Contribuer à la consolidation de la posture de sécurité de l'entreprise à travers **un reporting détaillé et des recommandations**.

1.2 Objectifs du rapport et pertinence de la mission

Ce rapport a pour objectif de présenter de manière détaillée le déroulement et les apports de cette expérience, en mettant en évidence l'acquisition de nouvelles compétences techniques et méthodologiques, tout en soulignant la pertinence du pentesting dans la consolidation de la sécurité des systèmes d'information. Il s'agit à la fois d'un compte rendu professionnel illustrant la mission confiée et d'un document académique démontrant l'intégration harmonieuse entre théorie et pratique dans le domaine de la cybersécurité.

2. Présentation de l'entreprise CBI

La Compagnie Bancaire Informatique, plus connue sous le nom de **CBI Maroc**, est un acteur majeur dans le domaine des technologies de l'information au Maroc et en Afrique francophone. Forte de **plus de 55** années d'expérience, l'entreprise s'est imposée comme un intégrateur **IT global**, accompagnant aussi bien les grandes entreprises que les institutions publiques dans leur transformation numérique et la sécurisation de leurs infrastructures.

2.1 Historique et positionnement

Créée dans le but initial de répondre aux besoins informatiques du secteur bancaire, **CBI** a progressivement diversifié ses activités pour devenir un partenaire incontournable dans l'intégration de solutions IT. Son expertise repose sur un solide réseau de partenariats stratégiques avec des éditeurs et constructeurs de renommée mondiale, ainsi que sur plus de **600 certifications** détenues par ses ingénieurs et consultants.
Aujourd'hui, CBI est présente au Maroc, confirmant ainsi son rôle de leader régional dans le domaine des technologies et de la cybersécurité.

2.2 Secteurs d'activité

L'entreprise intervient dans plusieurs domaines clés, parmi lesquels : la **transformation digitale**, le déploiement et la gestion des **infrastructures hybrides**, les **services managés**, le **cloud computing**, la **digitalisation des processus métiers** et la **gestion de la donnée**.
Ces activités s'articulent autour d'une vision globale qui place l'innovation et la sécurité au cœur des solutions proposées aux clients.

2.3 Organigramme de l'entreprise

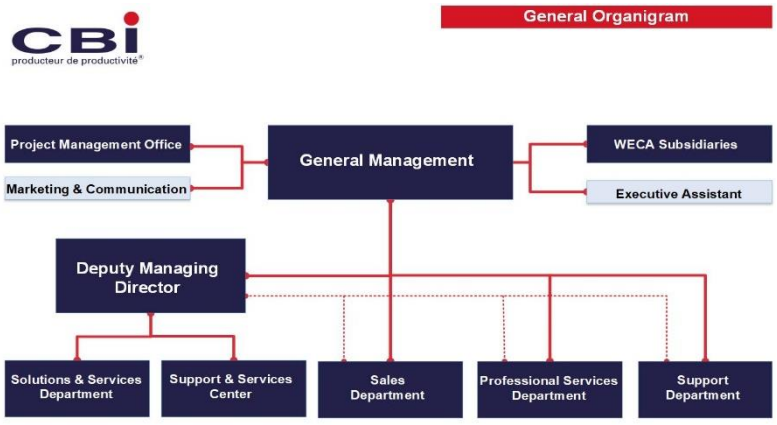

Figure 1 – Schéma d'organigramme de l'entreprise

3. Documentation & Perfectionnement théorique

La première étape de mon stage a été consacrée à la documentation et à l'appropriation des connaissances théoriques nécessaires à la réalisation de la mission. Elle s'est déroulée sur environ un mois et a couvert des notions fondamentales de cybersécurité, des frameworks méthodologiques, ainsi que des révisions techniques en réseaux et systèmes.

3.1 Cadres méthodologiques étudiés

3.1.1 Cybersécurité

La **cybersécurité** regroupe l'ensemble des méthodes, technologies et processus mis en place pour protéger les systèmes informatiques, les réseaux et les données contre les cyberattaques.

Ses trois objectifs principaux sont la **confidentialité** (empêcher l'accès non autorisé aux informations), l'**intégrité** (éviter les modifications non autorisées des données) et la **disponibilité** (garantir que les systèmes et données restent accessibles aux utilisateurs légitimes).

J'ai étudié les menaces les plus répandues, comme les malwares, ransomwares, attaques par phishing, ou encore les menaces internes.

Les menaces en cybersécurité sont multiples et évolutives :

Les **malwares** (virus, vers, chevaux de Troie, ransomwares) capables de corrompre ou bloquer un système.

Les **attaques par phishing** visant à tromper les utilisateurs pour obtenir des informations sensibles.

Les **attaques par déni de service (DDoS)**, saturant un serveur ou réseau pour le rendre indisponible.

3.1.2 Rôles en cybersécurité : Blue Team, Red Team & Purple Team

La Blue Team : le pilier défensif

La **Blue Team** incarne la dimension **défensive** de la cybersécurité. Elle est responsable de la **protection proactive et réactive** des systèmes d'information contre toute tentative d'intrus ion, de compromission ou de sabotage. Sa mission consiste non seulement à **ériger des barrières de sécurité**, mais aussi à assurer une **surveillance continue** et une **réponse efficace** en cas d'incident.

Missions principales :

- **Prévention et protection** : mise en place de contrôles de sécurité (pare-feu, IDS/IPS, EDR, segmentation réseau, règles d'authentification forte, politiques de mots de passe, etc.).
- **Surveillance continue** : utilisation de SOC (Security Operations Centers) pour collecter et corréler les journaux via des solutions SIEM (Security Information and Event Management).
- **Détection d'incidents** : identification d'anomalies, de comportements suspects ou d'indicateurs de compromission (IoC).
- **Réponse et remédiation** : isolement des machines compromises, neutralisation des malwares, restauration des services critiques, mise en place de correctifs.
- **Gestion des vulnérabilités** : scan régulier des systèmes avec des outils comme **Nessus, Qualys, OpenVAS** afin de détecter et corriger les failles.
- **Sensibilisation** : formation des utilisateurs pour réduire le risque humain (phishing, ingénierie sociale).

La Red Team : l'adversaire simulé

La **Red Team** représente la composante **offensive** de la cybersécurité. Contrairement aux attaquants malveillants, elle agit dans un cadre légal, avec l'accord de l'organisation, afin de **simuler des attaques réelles** et tester la robustesse des défenses.

Missions principales :

- **Pentesting** (tests d'intrusion) : identification et exploitation des vulnérabilités dans les applications, réseaux, bases de données ou systèmes.
- **Exercices de Red Teaming** : campagnes d'attaques simulées imitant fidèlement les tactiques,

techniques et procédures (TTP) des cybercriminels.

- **Ingénierie sociale** : tests de phishing, prétexting, baiting, tailgating pour mesurer la vigilance des collaborateurs.
- **Évaluation de la résilience** : mesurer la rapidité et l'efficacité de la Blue Team face à une intrusion simulée.

Outils utilisés :

La Red Team s'appuie sur des outils offensifs variés :

- **Scanning & reconnaissance** : Nmap, Maltego.
- **Exploitation** : Metasploit, SQLmap, Burp Suite.
- **Post-exploitation** : Cobalt Strike, Empire.
- **Attaques par force brute / mot de passe** : Hydra, John the Ripper.
- **Frameworks personnalisés** : scripts Python, PowerShell pour adapter les attaques.

La Purple Team : l'intermédiaire collaboratif

La **Purple Team** est née de la nécessité de combler le fossé entre la Red Team et la Blue Team. Plutôt que de fonctionner en opposition (attaquant/défenseur), la Purple Team favorise une **collaboration continue**, transformant chaque attaque simulée en un **apprentissage immédiat** pour les défenseurs.

Missions principales :

- **Coordination et communication** : faciliter le partage des résultats entre Red et Blue Team.
- **Exercices conjoints** (*Purple Teaming*) : simuler une attaque en temps réel et évaluer la capacité de détection et de réponse.
- **Amélioration continue** : transformer les failles découvertes par la Red Team en règles de détection pour la Blue Team.
- **Optimisation des défenses** : aligner les tactiques de détection avec les TTP décrits dans le **MITRE ATT&CK**..

Complémentarité et synergie :

Ces trois équipes forment un **écosystème complet** de cybersécurité :

- La **Red Team** agit comme l'attaquant, révélant les vulnérabilités.
- La **Blue Team** joue le rôle du défenseur, détectant et neutralisant les menaces.
- La **Purple Team** assure la boucle de retour, garantissant que les leçons apprises deviennent des mécanismes défensifs efficaces.

Ainsi, ce triptyque permet d'instaurer une logique de **cyber-résilience** : anticiper, protéger, détecter, répondre et améliorer.

3.1.3 Pentesting

Le **Pentesting**, ou **test d'intrusion**, est une pratique de cybersécurité offensive qui consiste à simuler des attaques réelles contre un système d'information, une application ou un réseau, dans le but d'identifier et d'exploiter des vulnérabilités avant qu'elles ne puissent être utilisées par de véritables cybercriminels. Contrairement à un audit de sécurité classique, qui se limite souvent à une analyse théorique ou automatisée, le pentest inclut une **phase active d'exploitation**, permettant d'évaluer concrètement le niveau de résistance des défenses mises en place.

Le pentesting repose donc sur une logique préventive : **penser et agir comme un attaquant** pour renforcer la sécurité organisationnelle.

Étapes d'un pentesting :

Un pentest suit généralement une méthodologie bien définie, inspirée de standards internationaux comme l'**OWASP Testing Guide**, l'**OSSTMM (Open Source Security Testing Methodology Manual)** ou encore le **PTES (Penetration Testing Execution Standard)**.
Les étapes typiques sont :

① **Planification et cadrage** : définition du périmètre, des objectifs, des règles d'engagement (black-box, white-box, grey-box).

② **Collecte d'informations (Reconnaissance)** : recherche passive et active d'informations sur la cible (OSINT, scans, fingerprinting).

③ **Analyse et énumération** : identification des services, systèmes d'exploitation, applications et ports ouverts.

④ **Exploitation** : tentative d'intrusion en exploitant les vulnérabilités détectées (SQLi, XSS, buffer overflow, mauvaises configurations, etc.).

⑤ **Escalade de privilèges** : recherche d'un accès administrateur ou d'un contrôle total sur le système.

⑥ **Mouvement latéral** : propagation dans le réseau interne pour atteindre des cibles stratégiques.

⑦ **Maintien de l'accès (Persistence)** : mise en place de backdoors ou d'autres mécanismes.

⑧ **Rédaction du rapport final** : description des vulnérabilités découvertes, preuves d'exploitation (PoC) et recommandations de remédiation.

Types de pentesting :

On distingue également des **catégories spécifiques** :

- **Pentesting réseau** : évaluation des pare-feu, routeurs, protocoles, services exposés.
- **Pentesting applicatif** : recherche de failles logicielles (injections, XSS, CSRF, etc.).
- **Pentesting physique** : simulation d'intrusion dans des locaux pour tester la sécurité physique.
- **Pentesting social** : utilisation de techniques d'ingénierie sociale (phishing, prétexting).

3.1.4 Réseau Informatique

Le modèle OSI (Open Systems Interconnection) :

Le **modèle OSI** est un cadre théorique à 7 couches qui décrit comment les données circulent dans un réseau :

1. **Couche physique** : câblage, signaux électriques, fibre optique, Wi-Fi.
2. **Couche liaison de données** : adressage MAC, détection/correction d'erreurs (Ethernet).
3. **Couche réseau** : adressage IP, routage (IPv4, IPv6).
4. **Couche transport** : transmission de bout en bout (TCP, UDP).
5. **Couche session** : gestion des connexions entre applications.
6. **Couche présentation** : traduction des données (chiffrement, compression, encodage).
7. **Couche application** : interface utilisateur (HTTP, FTP, DNS, SMTP).

En cybersécurité, ce modèle est une **référence analytique** : il permet de localiser et comprendre où une attaque agit (ex. attaque DDoS couche 3, injection SQL couche 7).

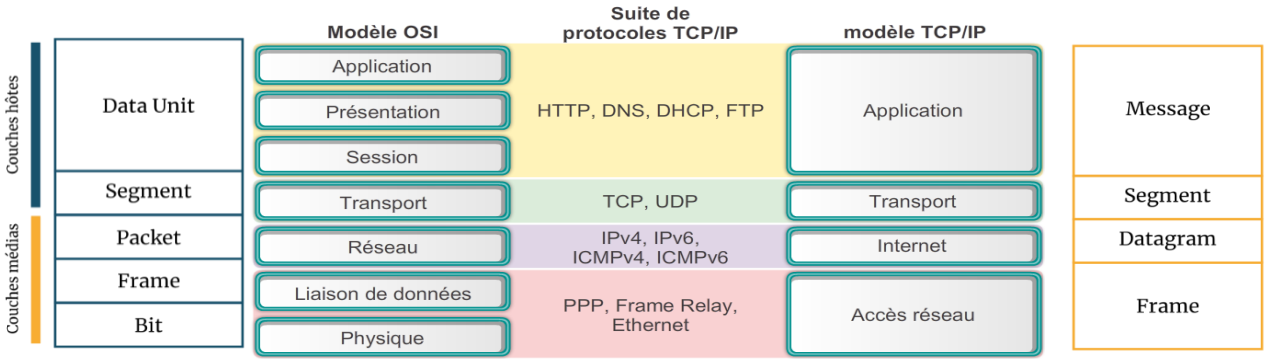

Figure 2 – Les couches du modèle OSI

Importance de la maîtrise des réseaux en cybersécurité :

Un professionnel en cybersécurité doit maîtriser les réseaux afin de :

- **Identifier les failles structurelles** (ex. absence de segmentation).
- **Comprendre les vecteurs d'attaque réseau** (scan de ports, sniffing, spoofing, MITM).
- **Mettre en place des contre-mesures efficaces** (IDS, segmentation VLAN, durcissement des protocoles).
- **Analyser un incident** en retraçant la source d'un trafic malveillant.

Ainsi, la connaissance approfondie des réseaux constitue une **base incontournable** avant de passer à la pratique du pentesting.

3.2 Cyber Kill Chain

Le **Cyber Kill Chain**, développé par Lockheed Martin en 2011, est un modèle méthodologique qui décrit les différentes phases d'une cyberattaque, depuis la préparation initiale par l'attaquant jusqu'à l'atteinte des objectifs finaux (exfiltration de données, sabotage, espionnage, etc.). Inspiré du concept militaire de kill chain (chaîne de destruction), ce modèle met en évidence **la progression logique et séquentielle d'une attaque**, permettant ainsi aux défenseurs d'identifier à quel moment intervenir pour la stopper.
L'intérêt du Cyber Kill Chain réside dans **sa vision structurée des menaces** : en décomposant l'attaque en étapes, il devient possible de renforcer la détection et la prévention à chaque phase. Ce modèle est largement utilisé dans les **Security Operations Centers (SOC)** pour l'analyse, la corrélation d'événements et la mise en place de mécanismes de défense proactive.

Les 7 étapes du Cyber Kill Chain :

① Reconnaissance (Reconnaissance)

L'attaquant collecte des informations sur la cible : noms de domaine, adresses IP, technologies utilisées, profils d'employés. Des outils comme whois, nmap, Maltego ou l'OSINT (Open Source Intelligence) sont utilisés.

② Armement (Weaponization)

Sur la base des informations collectées, l'attaquant conçoit son arme : un malware, un exploit, un document piégé, etc. C'est la préparation technique de l'attaque.

③ Livraison (Delivery)

Le vecteur d'attaque est transmis à la cible. Cela peut se faire par e-mail (phishing), lien malveillant, clé USB infectée ou injection sur un site vulnérable.

④ Exploitation (Exploitation)

L'attaque s'exécute lorsque la victime interagit avec le vecteur (ouvre une pièce jointe, clique sur un lien). Le malware exploite alors une vulnérabilité du système ou de l'application.

⑤ Installation (Installation)

Le malware s'installe sur la machine cible pour établir une persistance (ex. backdoor, rootkit). Cela permet à l'attaquant de conserver l'accès même après redémarrage.

⑥ Command & Control (C2)

Le système compromis entre en communication avec le serveur de l'attaquant, permettant le contrôle à distance. Cette étape transforme la machine en bot contrôlé.

⑦ Actions sur les objectifs (Actions on Objectives)

L'attaquant réalise sa finalité : vol de données, sabotage, espionnage, chiffrement de fichiers (ransomware), etc.

Figure 3 – Les étapes de Cyber Kill Chain

Intérêts et apports :

- Fournit une **cartographie claire des attaques**, utile pour les SOC.
- Permet une **détection proactive** : chaque étape est un point d'arrêt potentiel.

- Sert de **base à la Threat Intelligence** pour comprendre les tactiques, techniques et procédures (TTP) des attaquants.
- Facilite la **mise en place de défenses multicouches** (défense en profondeur).

Limites du modèle :

✗ Le Cyber Kill Chain est centré sur les attaques linéaires de type APT (Advanced Persistent Threat), mais il s'adapte moins bien aux attaques modernes qui sont itératives et multi-vecteurs.

✗ Les menaces internes (insiders) ou attaques purement sociales (ex. phishing sans malware) ne sont pas toujours bien couvertes.

✗ Les attaques « fileless » et rapides, qui n'ont pas toutes les étapes classiques, peuvent contourner ce modèle.

Le **Cyber Kill** Chain est une méthodologie essentielle en cybersécurité, car elle permet de transformer une attaque complexe en une série d'étapes compréhensibles et exploitables par les défenseurs. Son utilisation conjointe avec d'autres cadres comme **MITRE ATT&CK** renforce son efficacité, offrant une vision globale et opérationnelle de la menace.

3.3 MITRE ATT&CK

Le MITRE ATT&CK (*Adversarial Tactics, Techniques & Common Knowledge*) est **une base de connaissances mondiale** développée par l'organisation à but non lucratif **MITRE**. Elle recense et documente de manière systématique les **tactiques, techniques et procédures** (**TTPs**) utilisées par les cybercriminels et les groupes APT (Advanced Persistent Threats). Contrairement au modèle linéaire de la **Cyber Kill Chain**, ATT&CK propose une vision plus granulaire, dynamique et continue des comportements adverses, en se basant sur des observations réelles d'attaques.
Ce référentiel est aujourd'hui l'un des standards les plus utilisés par les entreprises et les **Security Operations Centers (SOC)** pour l'analyse des menaces, la détection d'intrusions, la chasse aux menaces (threat hunting) et la construction de défenses proactives.

Figure 4 – La plateforme de Mitre ATT&CK

Fonctionnement et structure :

Le framework MITRE ATT&CK est organisé sous forme de **matrices**, où chaque colonne représente une **tactique** (le but poursuivi par l'attaquant à une étape donnée) et chaque ligne détaille les **techniques** et **sous-techniques** permettant de l'atteindre.

- **Tactique** = *le pourquoi* (objectif de l'attaquant à une étape, ex. persistance).
- **Technique** = *le comment* (méthode utilisée, ex. création d'une clé de registre).
- **Sous-technique** = *la précision* (ex. clé Run de Windows pour exécuter un programme au démarrage).

Les tactiques principales (dans ATT&CK Enterprise Matrix)

① **Reconnaissance** : collecte d'informations sur la cible (scans, OSINT).

② **Resource Development** : préparation des ressources d'attaque (création d'infrastructures malveillantes, faux comptes).

③ **Initial Access** : obtention d'un premier point d'entrée (phishing, exploitation de vulnérabilités).

④**Execution** : exécution de code malveillant sur le système.

⑤ **Persistence** : maintien d'un accès durable (backdoor, comptes cachés).

⑥ **Privilege Escalation** : élévation de privilèges pour obtenir plus de droits.

⑦ **Defense Evasion** : techniques pour contourner les antivirus, EDR ou logs.

⑧ **Credential Access** : vol de mots de passe et tokens d'authentification.

⑨ **Discovery** : exploration du système et du réseau (cartographie interne).

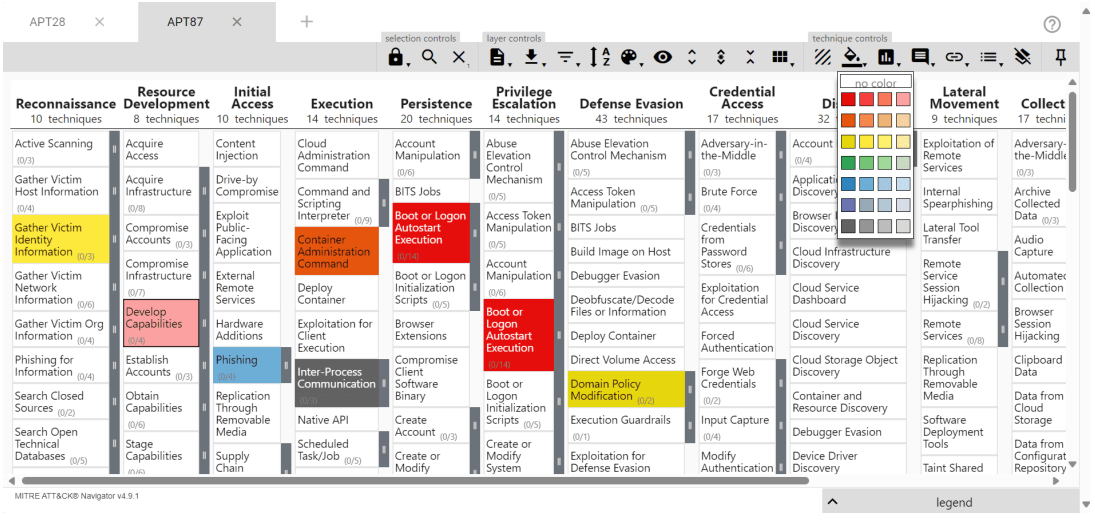

Figure 5 – Le navigateur ATT&CK

Apports du MITRE ATT&CK en cybersécurité :

- **Standardisation mondiale** : permet aux entreprises et SOC de parler le même langage en matière de menaces.
- **Détection et monitoring** : fournit des indications précises sur les indicateurs de compromission (IoCs) et les journaux à surveiller.
- **Threat Hunting** : aide les analystes à rechercher activement des signes d'intrusion avancée.
- **Simulation d'attaques (Red Teaming)** : sert de base aux tests d'intrusion pour reproduire fidèlement les comportements adverses.
- **Renforcement défensif** : permet d'identifier les lacunes dans les mécanismes de détection et d'améliorer la posture de sécurité globale.

Différences avec la Cyber Kill Chain :

- **Cyber Kill Chain :** modèle linéaire et stratégique, utile pour comprendre la progression globale d'une attaque.
- **MITRE ATT&CK :** modèle détaillé et dynamique, utile pour analyser comment les attaquants procèdent à chaque étape, avec des cas réels.

En pratique**, les deux sont complémentaires :** la Kill Chain donne une vue macro du cycle d'attaque, tandis qu'ATT&CK fournit une vue micro, technique et exploitable en temps réel.

Limites du MITRE ATT&CK :

✗ Sa richesse peut devenir une contrainte : des centaines de techniques sont répertoriées, ce qui peut rendre difficile leur exploitation sans outils adaptés.

✗ ATT&CK décrit surtout les techniques connues ; il ne couvre pas encore toutes les innovations ou attaques émergentes.

Le **MITRE ATT&CK** s'impose aujourd'hui comme un **cadre incontournable** en cybersécurité offensive et défensive. Sa granularité permet aux entreprises d'anticiper et de détecter plus efficacement les comportements adverses. En complément de la **Cyber Kill Chain**, il fournit une approche moderne, réaliste et adaptée aux attaques avancées actuelles, renforçant ainsi la capacité des SOC et des équipes de sécurité à protéger les systèmes d'information.

3.4 OWASP Top10

L'**OWASP** (Open Web Application Security Project) est une organisation mondiale à but non lucratif qui œuvre pour améliorer la sécurité des logiciels et applications web. Elle met à disposition de la communauté des développeurs, administrateurs systèmes et experts en cybersécurité un ensemble de ressources gratuites : guides, outils, frameworks et recommandations.

Parmi ses initiatives les plus connues figure l'OWASP Top 10, une classification régulièrement mise à jour qui recense les **dix principales menaces de sécurité** pesant sur les applications web. Publié pour la première fois en 2003 et actualisé environ tous les quatre ans, le Top 10 est devenu un **référentiel de sécurité incontournable**, adopté par les entreprises, auditeurs et pentesteurs dans le monde entier.

Figure 6 – Les vulnérabilités de OWASP TOP 10

Dans le cadre de ce stage, j'ai retenu trois vulnérabilités issues du Top 10 (ou s'y rapportant directement) pour les exploiter et les analyser en laboratoire : **SQL Injection (SQLi)**, **Command Injection**, et **Broken Access Control**. Ces choix ont été motivés par leur fréquence dans les applications web et par l'intérêt pédagogique qu'ils présentent (exploitation directe, impacts concrets et contre-mesures claires).

SQL Injection (Injection — correspond à A03 dans le Top 10)

Description :
L'injection SQL survient lorsque des données non fiables fournies par l'utilisateur sont intégrées dans des requêtes SQL sans être correctement filtrées ou paramétrées. Un attaquant peut alors modifier la requête pour contourner l'authentification, lire, modifier ou supprimer des données.

Exemples d'impact : contournement d'authentification, vol de données sensibles (comptes, mots de passe), altération de la base.

Prévention : utiliser des requêtes préparées / paramètres liés, valider et normaliser les entrées, limiter les privilèges de la base de données, journaliser les erreurs SQL sans exposer de détail aux utilisateurs.

Command Injection (sous-catégorie d'Injection — A03)

Description :

La command injection se produit lorsqu'une application transmet des données utilisateur directement à une interface système (shell, commandes), permettant l'exécution de commandes arbitraires sur le serveur. Elle est souvent présente sur des fonctionnalités telles que ping, traceroute, ou traitement de fichiers lorsque les entrées ne sont pas correctement filtrées.

Exemples d'impact : exécution de commandes système, obtention d'un shell distant, lecture/écriture de fichiers sensibles, pivot vers d'autres machines.

Prévention : ne jamais concaténer des entrées utilisateur dans des commandes shell ; utiliser des API sécurisées, valider strictement les entrées (liste blanche), exécuter les processus avec des privilèges limités, appliquer le principe du moindre privilège.

Broken Access Control (A01)

Description :

Un contrôle d'accès défaillant permet à un attaquant d'accéder à des ressources ou fonctionnalités qui devraient lui être interdites (accès à des comptes d'autres utilisateurs, opérations réservées aux administrateurs, modification de paramètres via des URL manipulées).

Exemples d'impact : prise de contrôle de comptes, modification non autorisée de données, escalation de privilèges logiques.

Prévention : implémenter et vérifier systématiquement les contrôles d'accès côté serveur, appliquer le principe du moindre privilège, vérifier les autorisations pour chaque action et ressource, utiliser des tests automatisés d'IDOR et de privilèges.

Le **Top 10 OWASP** constitue une **référence incontournable** pour tout pentesteur et développeur. Il permet d'identifier les menaces critiques, de comprendre leur exploitation et de mettre en place des contre-mesures adaptées.

Dans le cadre de mon stage, il a servi de **fil directeur** à ma mission pratique : les trois vulnérabilités exploitées (SQL Injection, XSS, Composants vulnérables) figurent toutes parmi ce classement, confirmant leur **pertinence et criticité dans la sécurité web**.

4. Application pratique : Simulation du Pentesting

Après un premier mois consacré à la **documentation, aux fondamentaux de la cybersécurité et aux cadres méthodologiques** (Cyber Kill Chain, MITRE ATT&CK, SOC, pentesting, etc.), la seconde phase de mon stage a été orientée vers la **mise en pratique**. Afin d'acquérir des compétences techniques concrètes et de m'exercer dans des environnements sécurisés, j'ai eu recours à deux plateformes

pédagogiques majeures : **TryHackMe** et **PortSwigger Web Security Academy**.
Ces environnements m'ont permis de tester, expérimenter et assimiler les concepts vus lors de la phase théorique, en reproduisant des scénarios réalistes d'attaques et de défense.

4.1 Apprentissage et mise en pratique sur TryHackMe

TryHackMe est une plateforme en ligne qui propose des **labs interactifs et guidés** de cybersécurité. J'ai suivi plusieurs parcours (paths) et rooms thématiques qui m'ont permis de consolider mes connaissances.

a) Révision des fondamentaux

- **Introduction to Networking** : révision des concepts réseaux (adressage IP, routage, protocoles TCP/UDP).
- **Cybersecurity Fundamentals** : sensibilisation aux menaces, aux concepts de confidentialité/intégrité/disponibilité (CIA), et aux différents domaines de la cybersécurité.
- **SOC Fundamentals** et **SOC Engagement** : découverte des missions d'un analyste SOC, des outils utilisés (SIEM, IDS, logs), et des bonnes pratiques pour gérer un incident.

b) Pentesting et sécurité offensive

- **Pentesting Fundamentals** : apprentissage de la méthodologie d'un pentest (reconnaissance, exploitation, escalade de privilèges, post-exploitation).
- **Vulnerability Management** : identification et gestion des failles via des outils de scanning (Nessus, OpenVAS).
- **OWASP Top 10** : étude des principales vulnérabilités web, en particulier celles que j'ai choisies pour mon projet : **SQL Injection, XSS, Composants vulnérables/obsolètes**.
- **Juice Shop** : application web volontairement vulnérable, servant de laboratoire pratique pour tester l'exploitation des failles.

c) Rôles en cybersécurité

- **Red Teaming** : initiation aux techniques offensives utilisées par les attaquants (phishing, exploitation de vulnérabilités, mouvements latéraux).

Ces rooms m'ont permis non seulement de consolider mes acquis théoriques, mais aussi de me familiariser avec des environnements réalistes où chaque notion est appliquée de manière pratique.

4.2 Exercices d'exploitation sur PortSwigger Web Security Academy

En parallèle de **TryHackMe**, j'ai utilisé la plateforme **PortSwigger Web Security Academy**, un environnement gratuit spécialisé dans la **sécurité des applications web**.

a) OWASP Top 10 – Vulnérabilités étudiées

J'ai concentré mes efforts sur les failles que j'avais choisies comme cibles dans ma mission :

- **SQL Injection (SQLi)** : exploitation des requêtes non sécurisées pour accéder à des données sensibles.
- **Broken Access Control** : tests et exploitation de contrôles d'accès défaillants (IDOR, manipulation d'URL/paramètres) permettant d'accéder ou de modifier des ressources d'autres utilisateurs.
- **Command Injection** : injection de commandes système via des champs non filtrés , permettant l'exécution de commandes arbitraires et la compromission du serveur.

b) Autres vulnérabilités explorées sur PortSwigger

Pour enrichir ma pratique, j'ai également travaillé sur d'autres labs importants :

- **Cross-Site Scripting**
- **Server-side request forgery (SSRF)**
- **File Upload vulnerabilities**.

Chaque lab proposait une approche guidée, une mise en situation réaliste, et un correctif attendu, ce qui m'a permis de développer une compréhension complète : **exploitation + remédiation**.

Mise en place d'un environnement virtuel interne :

En parallèle des plateformes en ligne, j'ai mis en place un **environnement virtuel local** pour tester mes compétences de pentesting dans un cadre contrôlé. Cet environnement m'a permis de reproduire les vulnérabilités étudiées et de simuler des attaques réalistes.

- **Machines virtuelles sous Linux (Kali Linux)** pour disposer à la fois de systèmes cibles et d'outils d'attaque.
- **Serveurs vulnérables (Mutillidae)** installés comme cibles d'entraînement.
- **Outils de pentesting** utilisés :
    - Nmap (scan réseau),
    - Burp Suite (proxy, injection, interception de requêtes),

- SQLmap (exploitation SQLi),
- OWASP ZAP (analyse web),
- Hydra (brute force)..

4.3 Mise en place de l'environnement virtuel de test

Afin de mettre en pratique les connaissances acquises durant la phase théorique et d'expérimenter les scénarios d'attaques et d'exploitation de vulnérabilités, j'ai déployé un **environnement virtuel contrôlé**. Celui-ci avait pour objectif de fournir une infrastructure sécurisée, isolée et réaliste où les tests pouvaient être effectués sans risques pour des systèmes de production.

L'environnement a été constitué de deux machines principales :

- **Kali Linux**, utilisé comme poste d'attaque.
- **Mutillidae**, utilisé comme machine cible volontairement vulnérable.

Ces deux systèmes ont été virtualisés et configurés de manière à interagir dans un réseau local simulé.

4.3.1 Kali Linux – Machine d'attaque

Kali Linux est une distribution basée sur Debian, spécialement conçue pour le **pentesting** et l'**audit de sécurité**. Elle intègre par défaut une vaste collection d'outils utilisés par les professionnels de la cybersécurité.

Rôle de Kali Linux dans l'environnement :

Cette machine a servi de **poste d'attaque principal**, me permettant de :

- Scanner Mutllidae pour identifier ses services vulnérables.
- Lancer des exploits via Metasploit.
- Réaliser des attaques ciblées sur les vulnérabilités OWASP (SQLi, XSS, etc.).
- Documenter les résultats obtenus.

4.3.2 Mutillidae – Machine cible vulnérable

Mutillidae est une application web volontairement vulnérable, fournie sous forme d'image/container (Docker) ou déployable sur une VM. Elle reproduit de nombreuses failles web typiques (OWASP) et sert d'environnement pédagogique pour l'apprentissage du pentesting applicatif..

Vulnérabilités présentes dans Mutllidae :

- Injection SQL (SQLi) sur différents paramètres et formulaires.
- Cross-Site Scripting (XSS) réfléchi et stocké.
- Broken Access Control / IDOR (références d'objets prévisibles).
- Command injection via formulaires qui exécutent des commandes système (ex. ping).
- Mauvaise gestion des sessions et mots de passe faibles.
- Autres failles web (vulnérabilités dans les uploads, etc.).

Étapes de déploiement

Étant donné que Kali et Mutillidae partagent la même machine physique, la configuration réseau doit permettre des scans et captures réseau fiables tout en conservant une isolation logique suffisante.

Topologie et choix réseau :

- Kali (hôte) exécute Docker et contient la machine d'attaque (outils pentest).
- Mutillidae fonctionne dans un conteneur Docker exposé sur localhost:8080 (mode bridge par défaut).
- Le réseau Docker par défaut (bridge) est recommandé : il préserve l'isolation tout en permettant l'accès depuis l'hôte.

a) Cloner le dépôt Mutillidae :

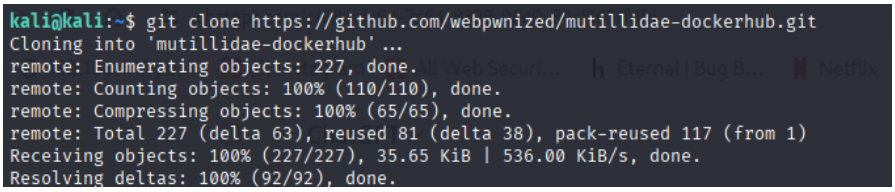

Figure 7 – Clonage du dépôt de Mutillidae

b) Installer docker-compose :

Figure 8 – Installation de Docker-compose

c) Lancer l'application avec docker-compose :

```
kali@kali:~$ cd mutillidae-dockerhub
kali@kali:~/mutillidae-dockerhub$ ls
docker-compose.yml  README.md  res  version
kali@kali:~/mutillidae-dockerhub$ sudo docker-compose up
[+] Running 81/33
 ✔ database_admin Pulled
 ✔ directory Pulled
 ✔ www Pulled
 ✔ directory_admin Pulled
 ✔ database Pulled
```

Figure 9 – Lancement de la machine vulnérable

d) La machine vulnérable est lancée :

Figure 10 – Page d'acceuil Mutillidae

4.3.3 Vulnérabilités à exploiter :

a) Injection SQL (SQLi)

La vulnérabilité SQLi permet à un attaquant d'injecter des commandes SQL dans les entrées de l'application, ce qui peut entraîner la divulgation, la modification ou la suppression de données sensibles.

Labs SQLi dans Mutillidae

Pour cette vulnérabilité on va faire ces 2 labs :

- SQLi - Extract Data
- SQLi - Bypass Authentification

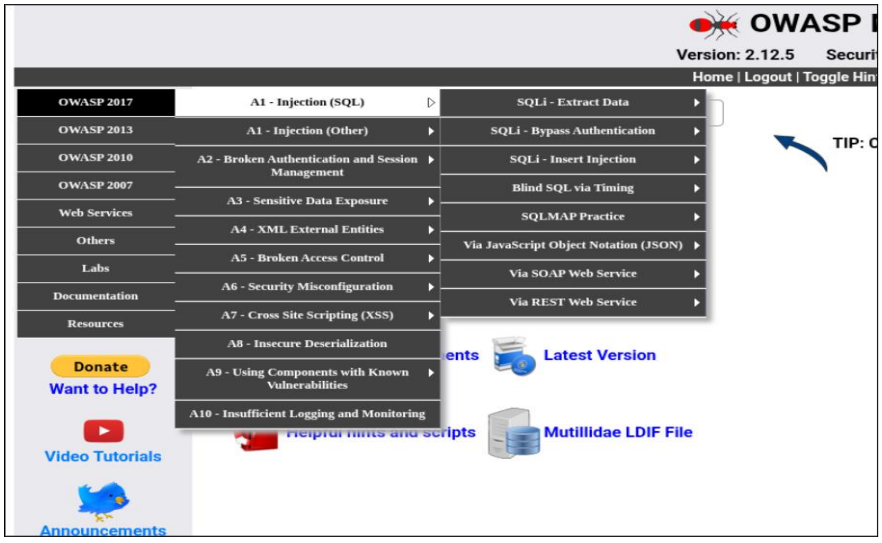

Figure 11 – Les vulnérabilités de Injection SQL

SQLi - Extract Data

1) Premierement ,on a accédé à la page de login.

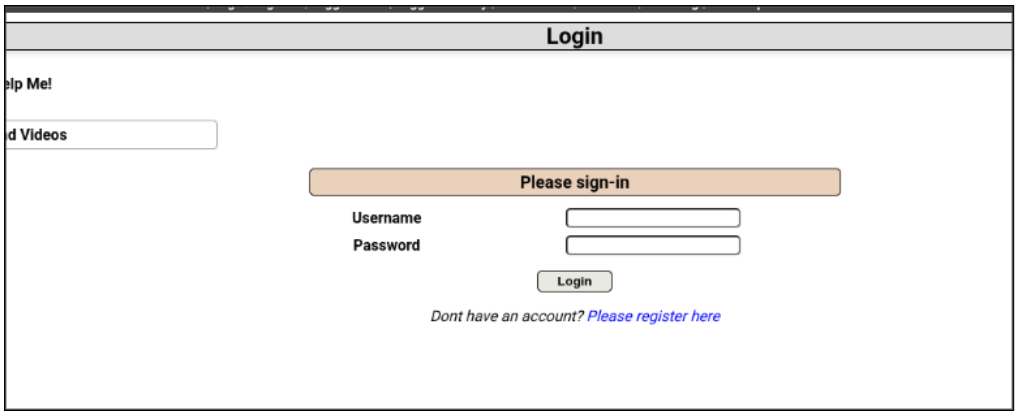

Figure 12 – Page de login du 1er lab

- On a utilisé les informations suivantes :

Username	vide
Password	‘ or 1=’1

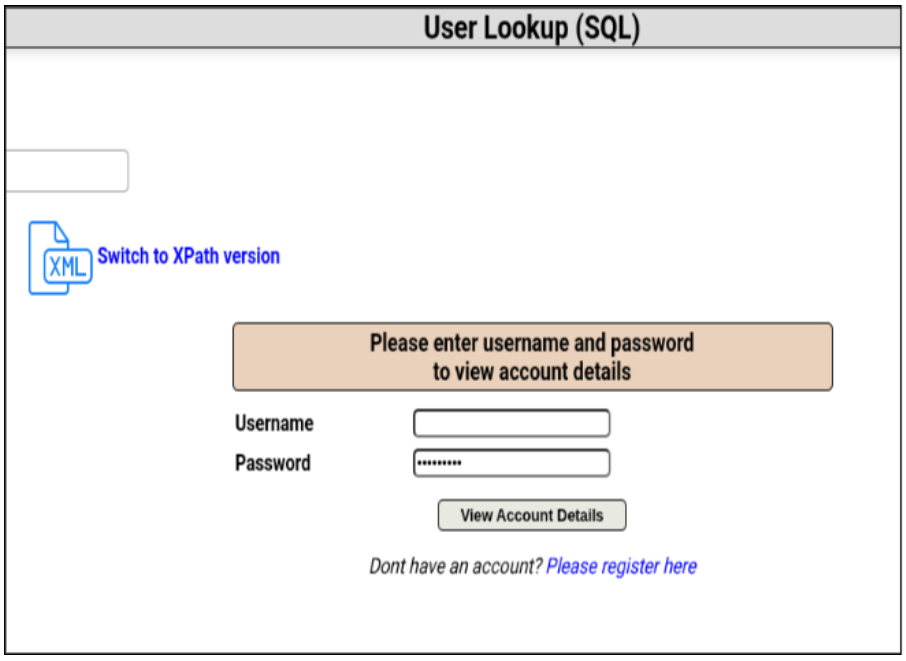

Figure 13 – Remplissage des champs (username vide, password = ' or 1='1)

2) Finalement, la liste des utilisateurs s'est affichée.

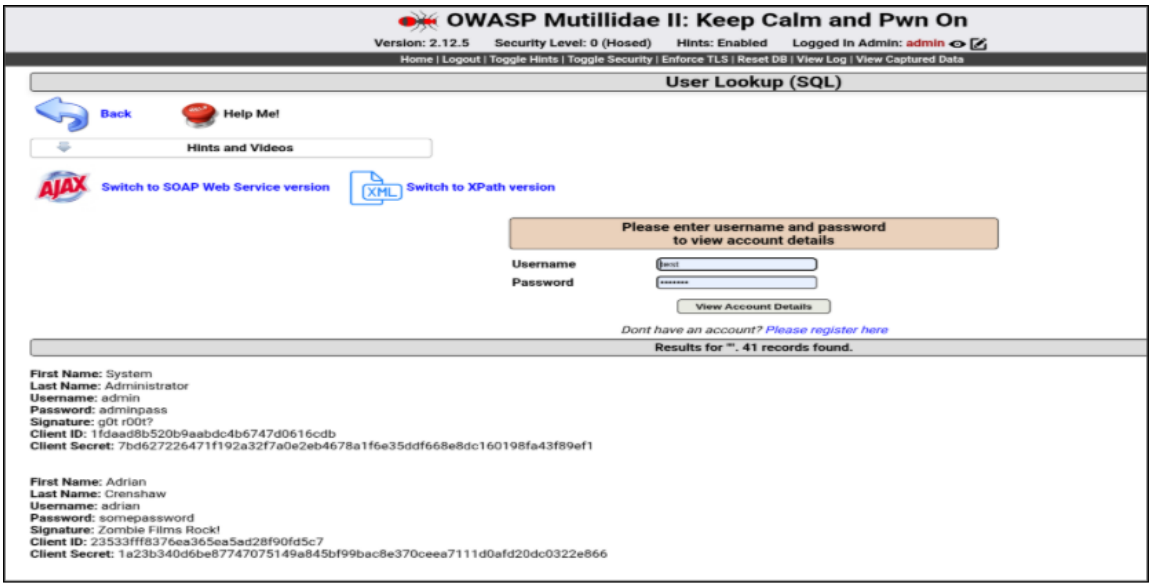

Figure 14 – Réponse de l'application (liste des utilisateurs affichée)

SQLi – Bypass Authentification

Dans cette partie, on va essayer de se connecter en tant qu'admin sans connaitre le mot de passe. On a essayé de se connecter avec les informations admin:admin. Le résultat était erreur d'authentification.

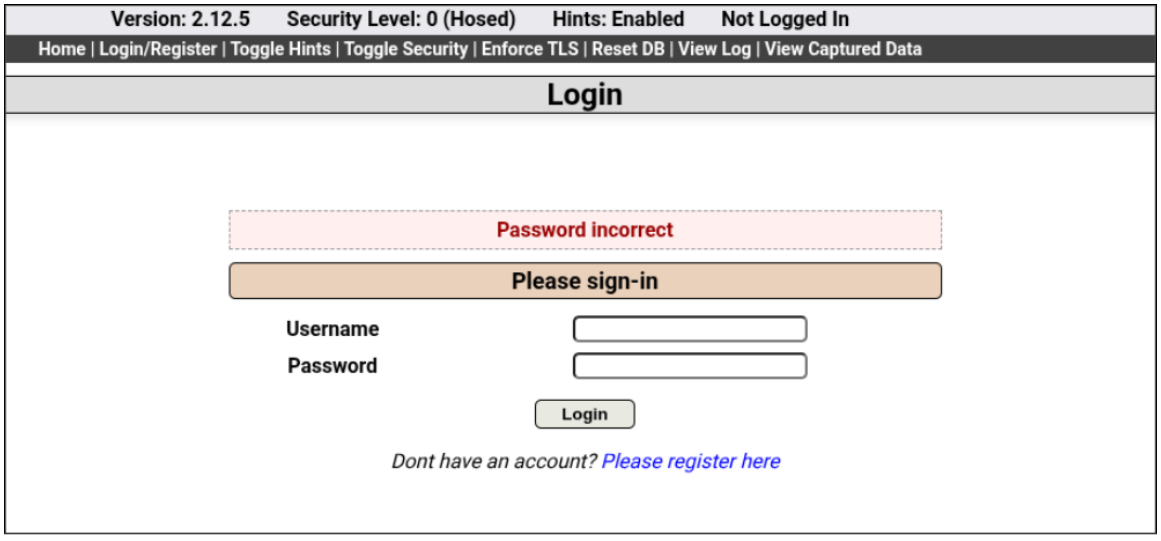

Figure 15 – Page de login de 2eme lab

- On a utilisé les informations suivantes :

Username	admin' or 1='1
Password :	admin

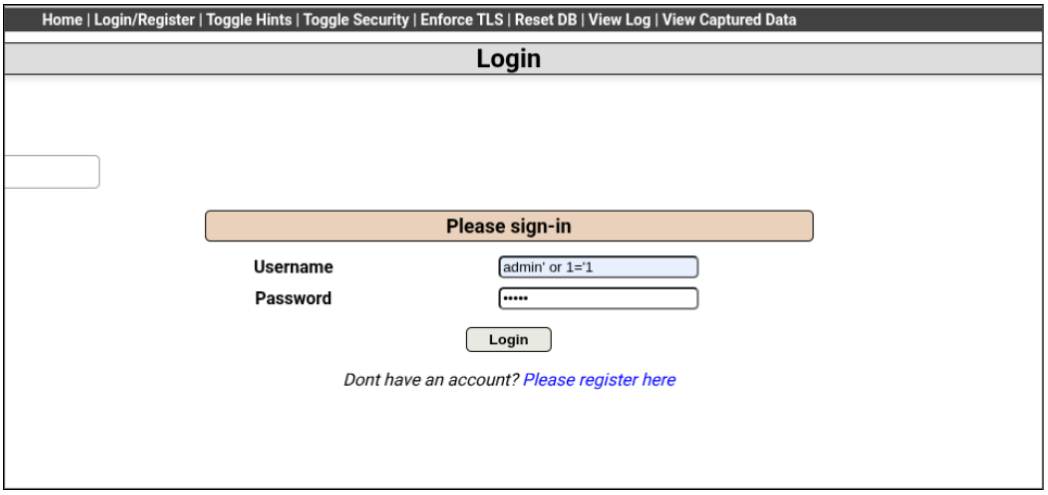

Figure 16 – Saisie du payload dans le formulaire

→ Le résultat était l'accès au compte administrateur.

Figure 17 – Soumission et réponse : accès admin

b) Broken Access Control

L'application ne restreint pas correctement l'accès selon les rôles ou permissions.

Labs de Broken Access Control dans Mutillidae

Pour cette vulnérabilité on va faire ces 2 labs :

- Text File Viewer
- Source Viewer

Figure 18 – Les vulnérabilités de Broken Access Control

Text File Viewer

Dans cette partie, on s'est connecté en tant qu'administrateur**.**

1) On a la page d'édition du profil utilisateur dans Mutillidae. On remarque **l'id** de l'utilisateur est affiché dans **l'url, donc** on peut le changer.

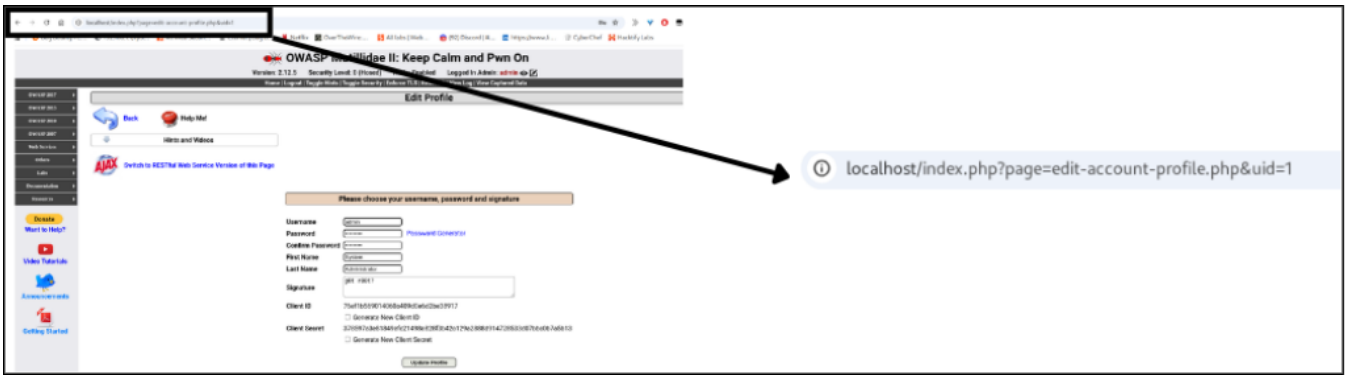

Figure 19 – Page d'édition du profil (uid=1

Son URL **:** http://localhost/index.php?page=edit-account-profile.php&uid=1

2) On a changé l'**id** de **1** à **2**. On a pu accéder à **autre profil** avec la capacité de **changer ses informations**.

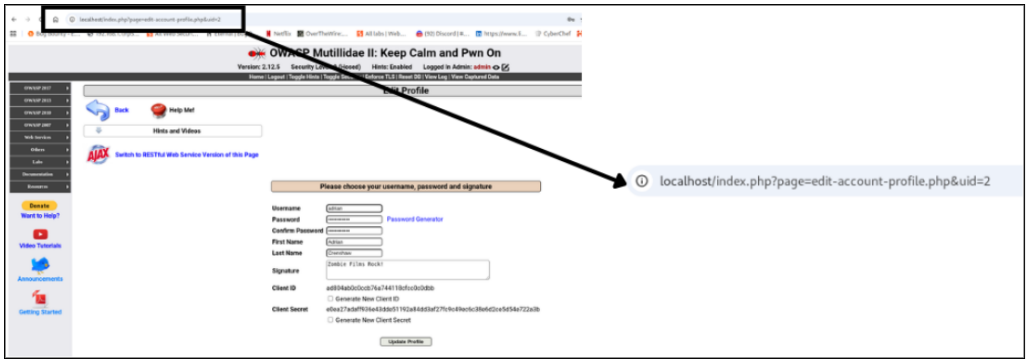

Figure 20 – Accès au profil d'un autre utilisateur (uid=2)

- De même pour **l'id 5.**

Figure 21 – Accès au profil d'un autre utilisateur (uid=5)

Source Viewer

1) On a une page qui permet de voir le code source PHP des fichiers du site.

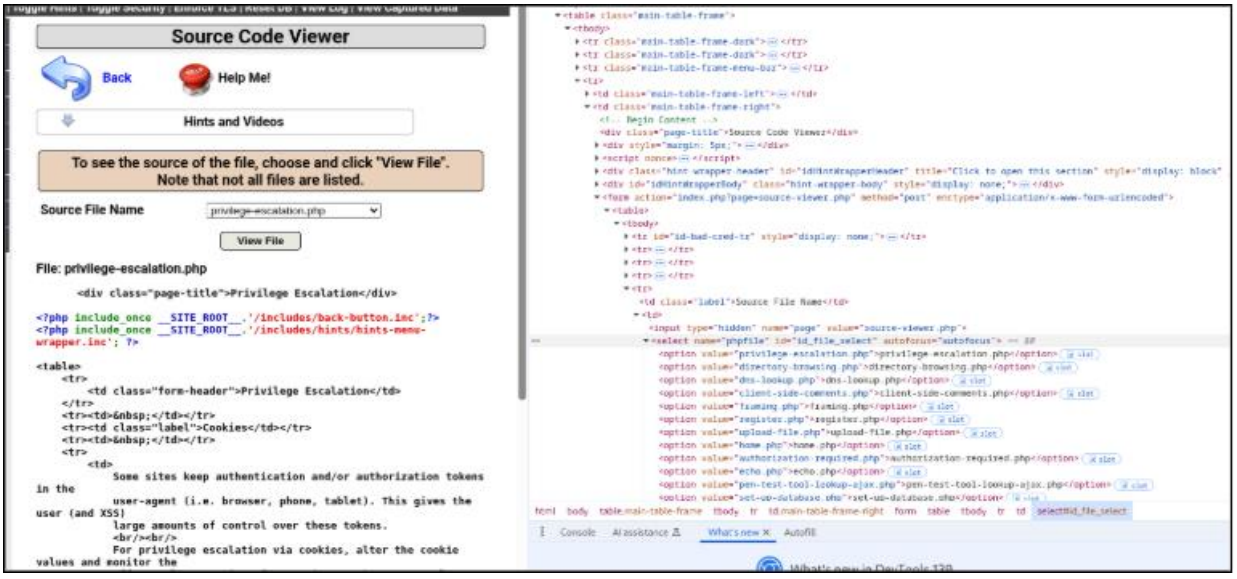

Figure 22 – Page Source Viewer initiale

- Si on voit le code source de la page, on trouve que les lien des fichiers sont apparues .

  On peut changer un lien d'un fichier par un autre qui est malveillant.

2) On fait ce changement "privilege-escalation.php" to "/etc/passwd" pour obtenir la liste des mots de passe.

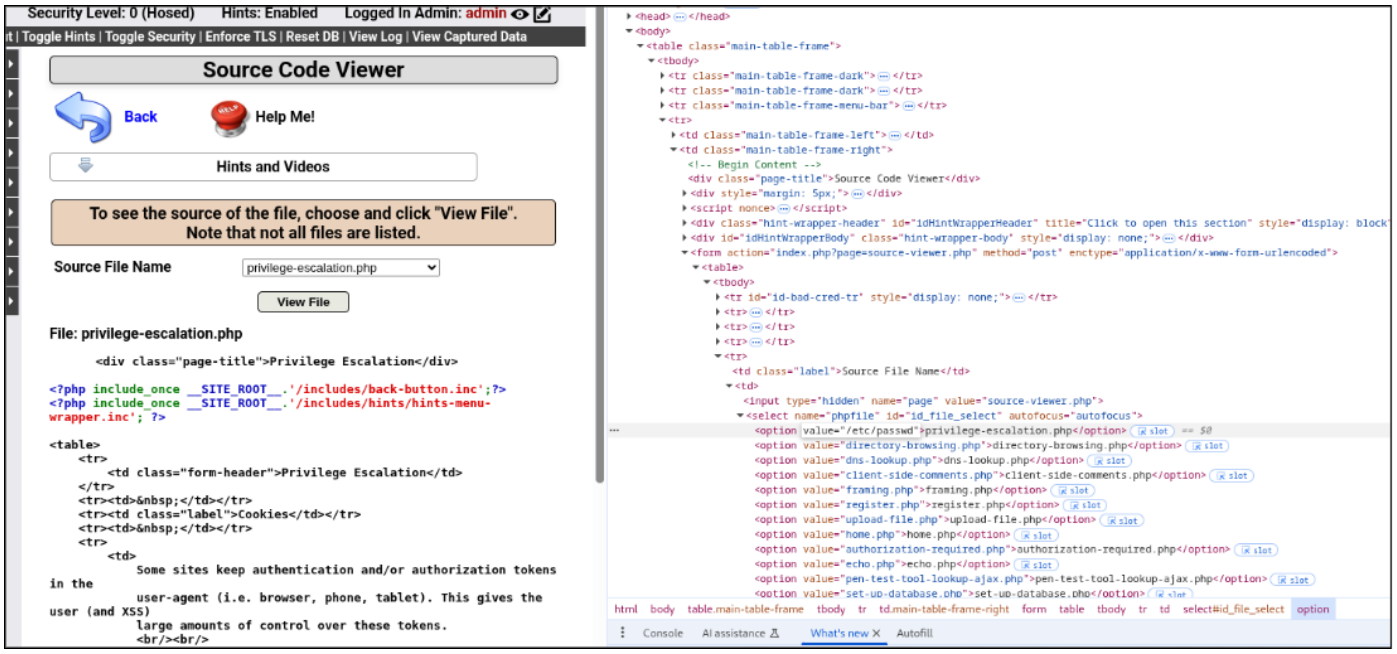

Figure 23 – Modification du chemin du fichier

3) on clique sur le boutton View File , on reçoit la liste des mots de passe.

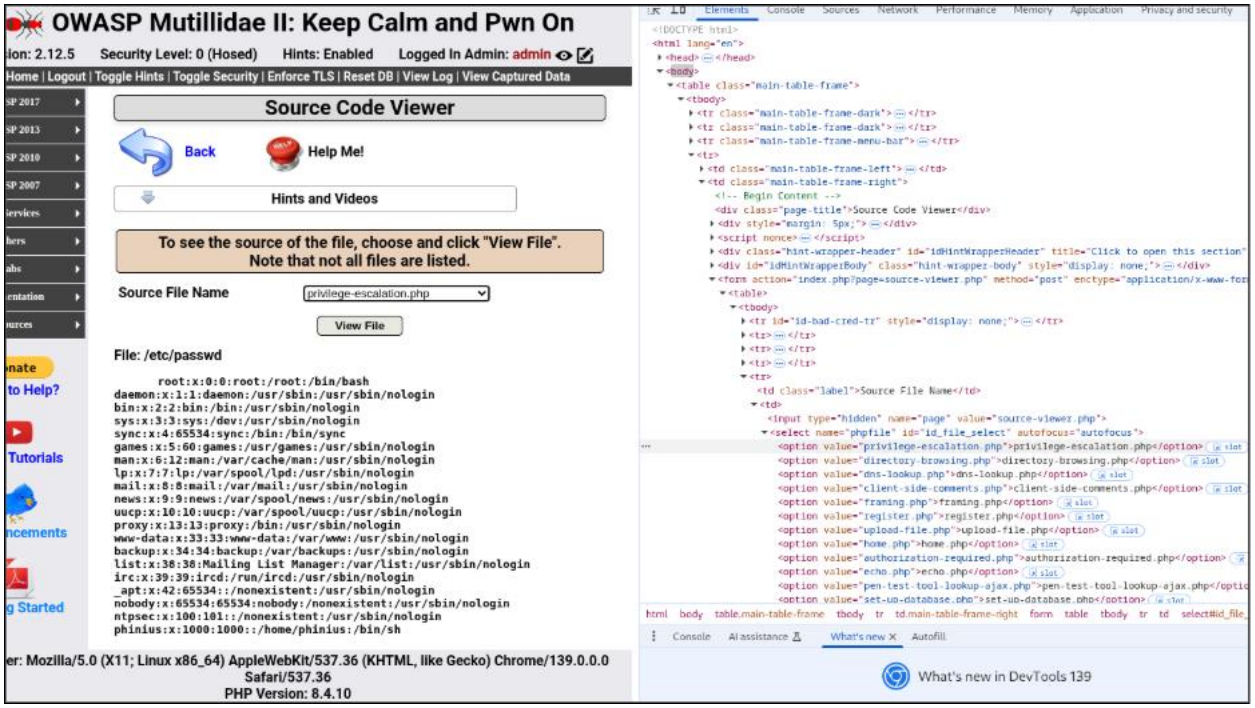

Figure 24 – Affichage du contenu de /etc/passwd

c) Injection de commande :

Possibilité d'exécuter des commandes systèmes via les champs de saisie

Labs d'Injection de commande dans Mutillidae

Pour cette vulnérabilité on va faire ces 2 labs :

- DNS lookup
- Echo Message

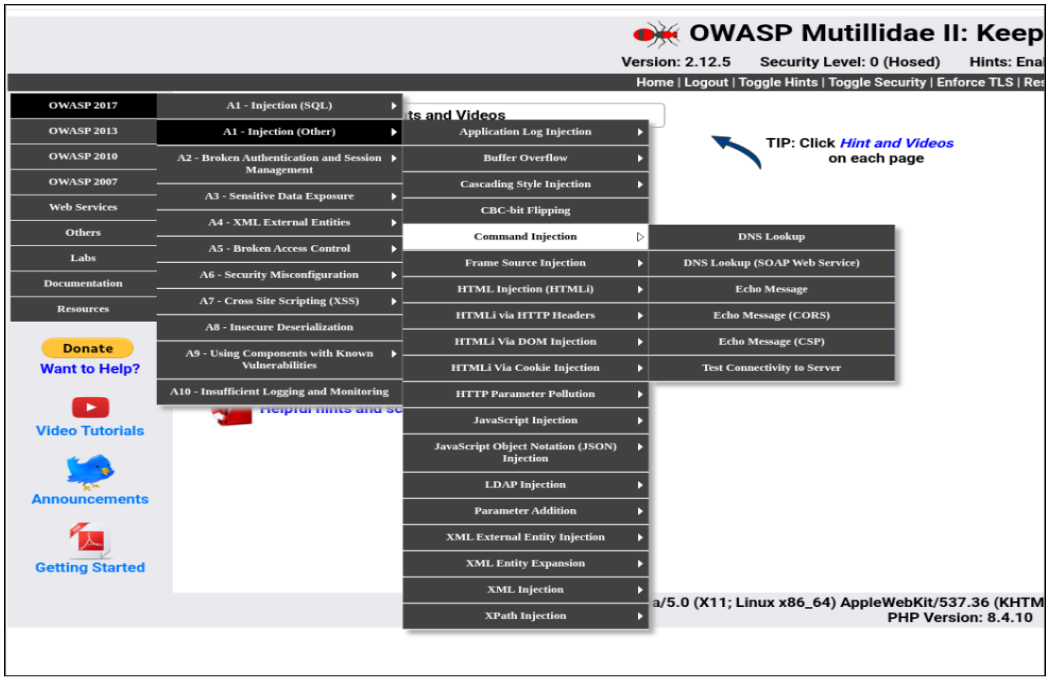

Figure 25 – Les vulnérabilités d'Injection de commande

DNS Lookup

Cette page permet de tester la résolution de noms ou la connectivité réseau en saisissant un domaine/IP et en affichant le retour de la commande système associée.

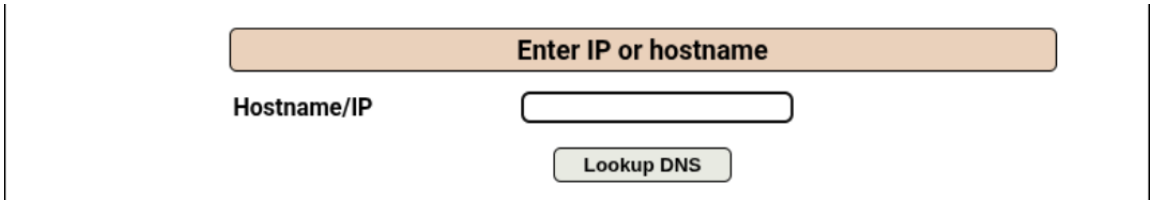

Figure 26 – Page de test de connectivité / résolution DNS

1) On a testé avec l'adresse ip 8.8.8.8.

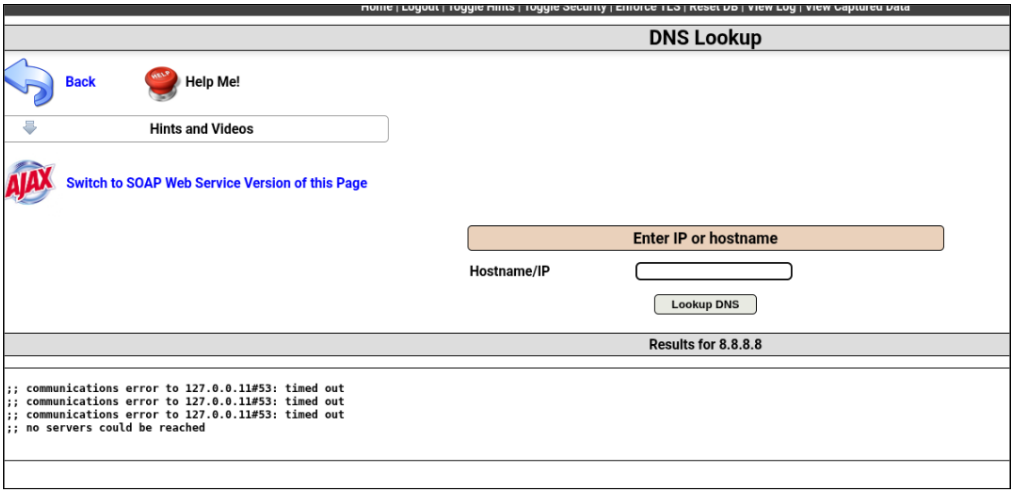

Figure 27 – Test avec IP normale (8.8.8.8)

2) On a essayé d'injecter ce 8.8.8.8|whoami. Il nous a retourné le nom de la machine de système.

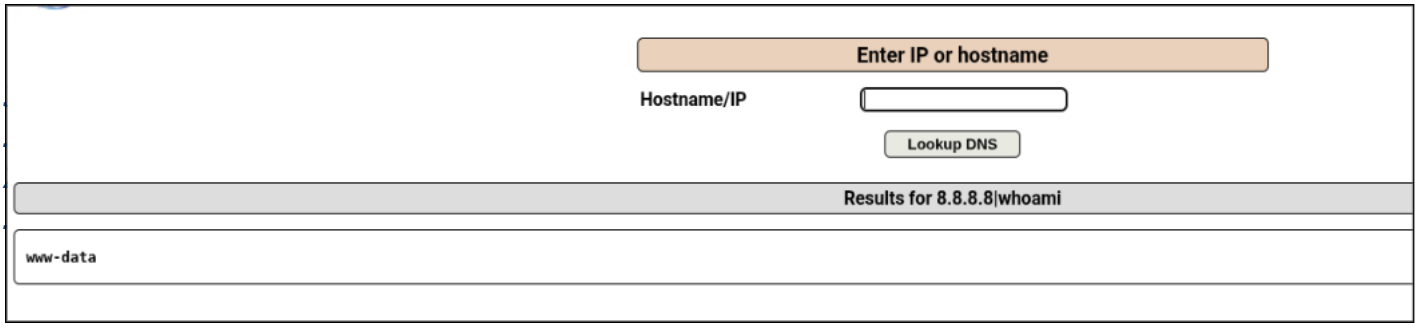

Figure 28 – Injection de commande (8.8.8.8|whoami)

Echo Message

Cette page prend ce que tu saisis dans le champ **Message** et l'affiche directement dans la réponse, **sans filtrage.**

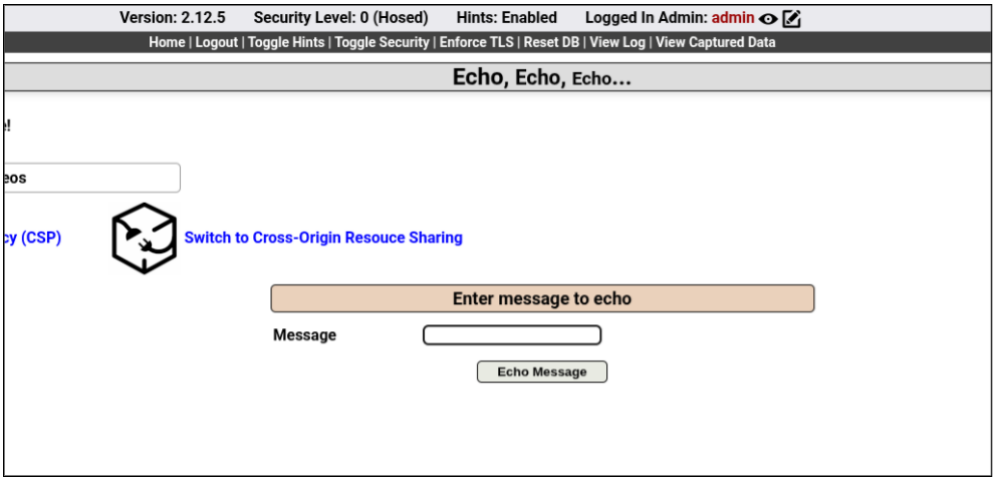

Figure 29 – Page Echo Message

2) On a injecté un message avec une commande hi&&whoami .

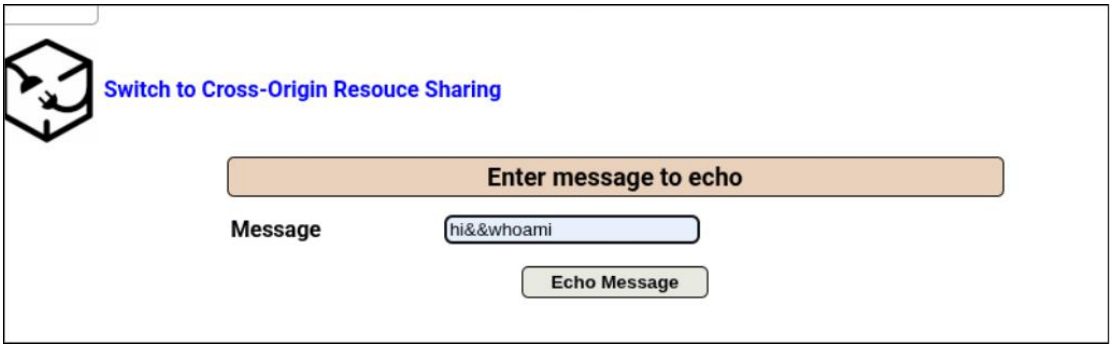

Figure 30 – Saisie du message avec commande (hi&&whoami)

3) Il nous a affiché le message "hi" avec le résultat de la commande "**whoami**".

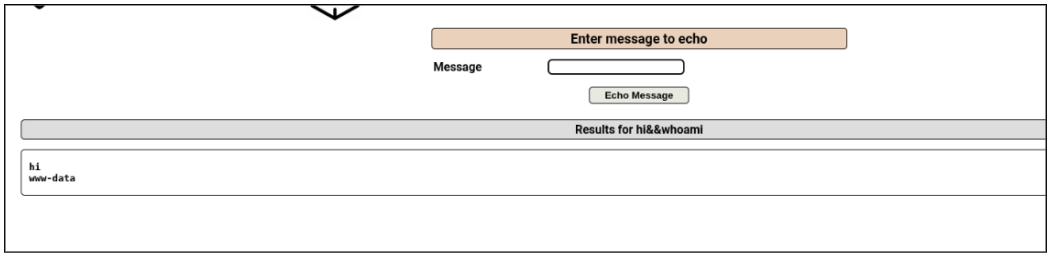

Figure 31 – Résultat de l'injection

Conclusion

Ce stage a constitué une expérience riche et formatrice, permettant de couvrir à la fois la **théorie** et la **pratique** de la cybersécurité. L'étude des modèles de référence (Cyber Kill Chain et MITRE ATT&CK) a permis de comprendre le cycle de vie d'une attaque et les différentes tactiques utilisées par les adversaires. Les présentations sur les bases réseaux et les équipes défensives/offensives ont apporté un cadre conceptuel solide.

La réalisation de **labs OWASP** a permis d'explorer concrètement les vulnérabilités web les plus courantes, tandis que la conception d'une **machine vulnérable** a offert un environnement réaliste pour mettre en œuvre des scénarios d'attaque et de défense. Ce travail pratique, structuré selon la Cyber Kill Chain et MITRE ATT&CK, a permis de mieux relier la **vision offensive** (Red Team) et la **vision défensive** (Blue Team).

En définitive, ce projet a renforcé nos compétences en analyse et en exploitation de vulnérabilités, mais aussi en détection et en réponse aux incidents

Bibliographie

- Https://tryhackeme.com
- Https://portswigger.net
- https://www.lockheedmartin.com/
- https://attack.mitre.org
- https://www.microsoft.com/en-us/security/business/security-101/what-is-cyber-kill-chain
- https://www.ibm.com/fr-fr/think/topics/mitre-attack
- https://mitre-attack.github.io/attack-navigator/
- https://iritt.medium.com/understanding-cyber-kill-chain-9b04eccd217b