De :

Antoine LABBE **antoine@intuity.fr**
Maher ABOU ABBAS **maher@intuity.fr**

A :

Issam Chebha **issam.chebha@go1.com**

Rapport De test D'intrusion

Mai 2024

Downloaded by mar.abdou@...

Executive Summary

La société **Coorpacademy**, a mandaté l'équipe sécurité d'Intuity afin de réaliser un test d'intrusion sur leur plateforme de formation en ligne.

Pendant la durée du test nous avons pu mettre en évidence deux vulnérabilités importantes :

- Une personne ayant accès au CMS pourrait uploader un fichier contenant du code malicieux sur la plateforme et s'en servir pour réaliser des attaques plus sophistiquées telles que du phishing.
- Certains composants utilisés sont obsolètes et vulnérables à des exploits connus.

De plus plusieurs mauvaises pratiques telles qu'une politique de mot de passe faible ou encore la divulgation de données techniques abaissent le niveau de sécurité de la plateforme.

Cependant, nous avons observé l'utilisation d'une protection de type WAF (pare-feu d'application Web) qui s'est avérée efficace, en particulier pour sécuriser le site contre les injections.

Dans la suite de ce document, vous trouverez une analyse des vulnérabilités détectées accompagnée de nos recommandations.

Downloaded at Thu, 27 Nov 2025 10:54:21 GMT
Downloaded by mohammad.aboou@skillupmena.com

Table des matières

Downloaded at Thu, 27 Nov 2025 10:54:21 GMT
Downloaded by muhammad.abdou@skillupmena.com

I. Périmètre

Le test d'intrusion a été réalisé pour une durée de 5 jours sur les applications mises à notre disposition aux adresses suivantes :

- https://pentest-2024-coorp-manager-staging.coorpacademy.com/
- https://coorpmanager-staging.coorpacademy.com/brands/pentest-2024-coorp-manager/
- https://pentest-2024-open-signup-staging.coorpacademy.com/
- https://pentest-2024-validated-signup-staging.coorpacademy.com/
- https://pentest-2024-first-connection-form-staging.coorpacademy.com/

II. Méthodologie

Ce test de pénétration a suivi une méthodologie dite boîte grise. Cette approche a pour avantage de déterminer les points sensibles et d'évaluer la sécurité de la cible face à un attaquant n'ayant pas de connaissances préalables de l'application, puis en utilisant certaines informations sur l'architecture et le fonctionnement de l'application.
Notre méthodologie de test est la suivante :

1. Récupération d'informations : obtention du maximum d'informations sur l'application et les technologies utilisées.

2. Exécution des outils de sécurité pour scanner les vulnérabilités communes.

3. Vérifications de sécurité manuelles sur les vulnérabilités les plus connues (les utilitaires automatisés n'ont pas la même logique que le testeur manuel).

4. Lecture et analyse du code de chaque composant afin d'obtenir des informations supplémentaires (HTML, JavaScript).

5. À partir de ces informations, identification de nouvelles surfaces d'attaque.

6. Une fois toutes ces étapes terminées, modification manuelle des paramètres identifiés

7. Exploitation des failles trouvées par modification des paramètres.

8. Enfin, à chaque nouvelle fonctionnalité trouvée, retour à l'étape 3.

Downloaded by muhammad.abdul@skillupmena.com
Downloaded at Thu, 27 Nov 2025 10:54:21 GMT

III. Tests réalisés

Cette liste mentionne les tests que nous avons réalisés :

Audit web

- Recherche de chemins « masqués » dans le fichier robots.txt
- Recherche des fichiers et dossiers courants accessibles
- Analyse de la sécurité des cookies
- Analyse des méthodes HTTP activées
- Analyse des entêtes de sécurité
- Analyse des erreurs de configuration
- Recherche de versions logicielles non mises à jour
- Analyse de l'implémentation TLS
- Analyse du filtrage et de l'encodage des saisies utilisateur
- Analyse des fonctionnalités d'envoi de fichiers
- Analyse des mécanismes de contrôle d'accès aux données
- Analyse des mécanismes de redirection contrôlée par l'utilisateur
- Tests sur la désérialisation
- Analyse du mécanisme d'authentification des API
- Analyse du mécanisme de gestion des sessions
- Scan de vulnérabilités automatisé

Downloaded by muhammad.abdou@skillupmena.com on Thu, 27 Nov 2025 10:54:21 GMT
Downloaded by muhammad.abdou@skillupmena.com

IV. Synthèse des vulnérabilités

Réf	1 – Fonction d'upload sans restriction	Sévérité	Important
Exposition	Exploitable avec un compte « cms editor »		
Risque(s)	Un attaquant pourrait utiliser cette fonction pour importer des fichiers malicieux sur le serveur et s'en servir pour réaliser des attaques plus sophistiquées telles que la compromission de session, la diffusion de malware ou encore du phishing.		
Type	Upload de fichiers sans restriction https://owasp.org/www-community/vulnerabilities/Unrestricted_File_Upload		

Réf	2 – Utilisation de composants vulnérables	Sévérité	Important
Exposition	Exploitable avec un compte sur l'application		
Risque(s)	De vulnérabilités existantes exposent le site à des compromissions.		
Type	Composants vulnérables et obsolètes https://owasp.org/Top10/A06_2021-Vulnerable_and_Outdated_Components/		

Réf	3 – Divulgation d'informations techniques	Sévérité	Faible
Exposition	Exploitable avec un compte sur l'application		
Risque(s)	Un attaquant peut avoir accès à des informations sur la conception du site ainsi que de l'API. Il pourrait se servir de ces informations afin de concevoir des attaques plus sophistiquées contre l'application.		
Type	Mauvaise configuration de la sécurité https://owasp.org/Top10/fr/A05_2021-Security_Misconfiguration/		

Réf	4 – Politique de mot de passe faible	Sévérité	Faible
Exposition	N/A		
Risque(s)	Des mots de passes faibles pourront être définis, et stockés pendant des durées arbitraires sur des supports non sécurisés, ce qui faciliterait la démarche d'un attaquant cherchant à deviner les mots de passe.		
Type	Identification et authentification de mauvaise qualité https://owasp.org/Top10/fr/A07_2021-Identification_and_Authentication_Failures/		

Downloaded by mohamed.abdou@skilupmena.com on Thu, 23 Nov 2023 10:54:21 GMT

V. Liste des vulnérabilités

Cette liste énumère les vulnérabilités trouvées. Chaque vulnérabilité est présentée de la façon suivante :

- **Réf. :** Nom et n° de référence utilisés pour le suivi de la vulnérabilité.
- **Niveau de sévérité :** niveau de criticité lié à la faille défini comme suit :
  - **Faible** : ne met aucunement en danger le système mais donne de l'information sans attrait apparent. En revanche, l'utilisation combinée de ces failles, avec d'autres plus importantes, peut faciliter la compromission d'un système.
  - **Important** : la faille en elle-même n'est pas dangereuse, mais combinée avec une ou d'autres attaques, peut permettre la compromission du système.
  - **Critique** : on considère qu'une partie du site ou une fonctionnalité importante est compromise.
  - **Très critique** : on considère que la totalité du site et/ou de la base de données et/ou du réseau sont sous le contrôle de l'attaquant.
- **Exposition :** précise si l'attaquant a besoin d'être authentifié sur l'application pour exploiter cette vulnérabilité, ou de disposer d'un accès réseau particulier.
- **Risques :** conséquences si l'attaque est exécutée.
- **Commun à :** précise quelle partie du système est impactée.
- **Type de faille :** terminologie attribuée à la faille.
- **Description :** brève description avec généralement un exemple de la faille.
- **Recommandations :** une ou des recommandations pour corriger la faille.

Downloaded by mahamad.abdou@skillupmena.com [22 Nov 2025 10:54:21 GMT]
Downloaded by mahamad.abdou@skillupmena.com [22 Nov 2025 10:54:21 GMT]

Réf	1 – Fonction d'upload sans restriction	Sévérité	Important
Exposition	Exploitable avec un compte sur l'application		
Risque(s)	Un attaquant pourrait utiliser cette fonction pour importer des fichiers malicieux sur le serveur et s'en servir pour réaliser des attaques plus sophistiquées telles que la compromission de session, la diffusion de malware ou encore du phishing.		
Type	Upload de fichiers sans restriction https://owasp.org/www-community/vulnerabilities/Unrestricted_File_Upload		
Commun à	CMS		

Description

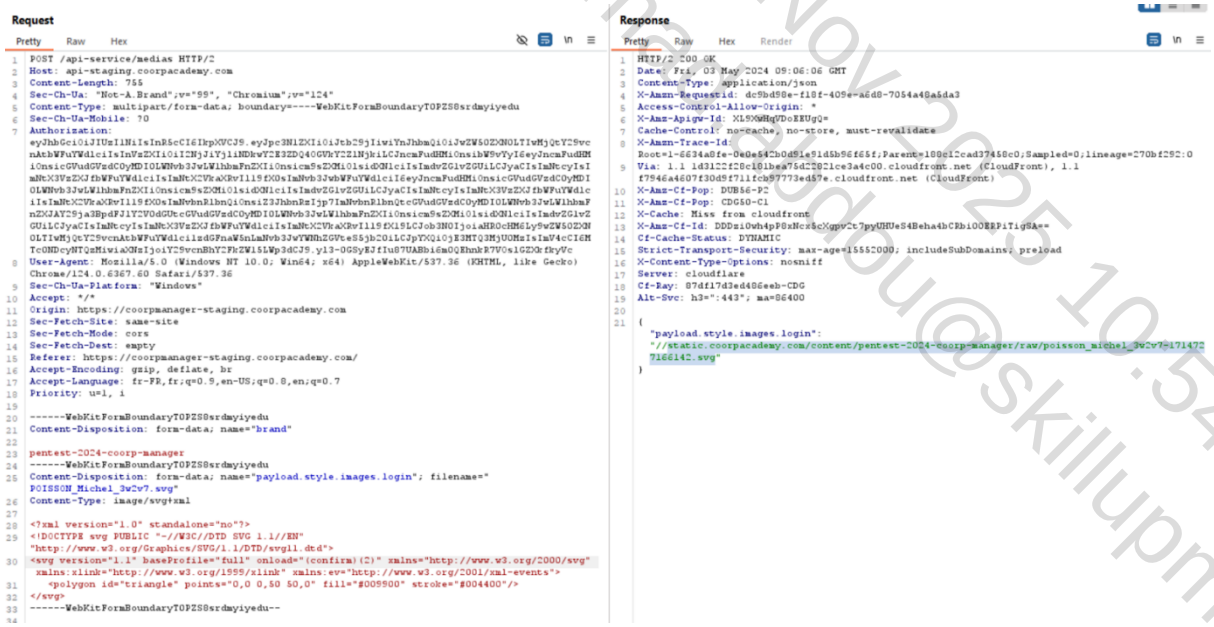
Un utilisateur connecté en tant qu'éditeur du CMS a la possibilité d'uploader des fichiers à plusieurs endroits comme ici par exemple : https://coorpmanager-staging.coorpacademy.com/brands/pentest-2024-coorp-manager/editorialization/look-and-feel

Or, les contrôles sur l'upload ne sont effectués qu'au niveau du navigateur et il est possible de les contourner en interceptant la requête et en modifiant l'extension du fichier ainsi que le content-type.

Il est alors possible d'injecter tous types de fichiers contenant du code malicieux tels que des fichiers svg.

Exemple :

- Upload d'un fichier svg contenant du code JavaScript permettant d'afficher une fenêtre popup :



- Affichage du fichier dans le navigateur et exécution du code :

Downloaded by muhammad.abdu@skillupmena.com on Thu, 27 Nov 2025 10:54:21 GMT

Recommandation

Toujours contrôler que l'extension du fichier est dans une liste blanche d'extensions autorisées.

Idéalement on vérifiera que l'extension annoncée concorde avec les données contenues dans le fichier.

Note : Attention à bien valider l'extension du fichier et pas seulement son Content-Type qui peut être facilement modifié.

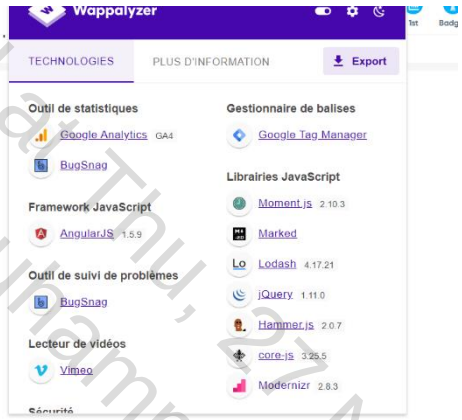
Ex : Imposer une extension lors de l'enregistrement du fichier sur le serveur.

Downloaded at Thu, 27 Nov 2025 10:54:21 GMT
Downloaded by muhammad.abdou@skillupmena.com

Réf	2 – Utilisation de composants vulnérables	Sévérité	Important
Exposition	Exploitable avec un compte sur l'application		
Risque(s)	De vulnérabilités existantes exposent le site à des compromissions.		
Type	Composants vulnérables et obsolètes https://owasp.org/Top10/A06_2021-Vulnerable_and_Outdated_Components/		
Commun à	Versions de composants		

Description

Les versions de plusieurs composants comme Angular, jquery ou npm sont obsolètes et vulnérables à des exploits connus :



https://security.snyk.io/package/npm/moment/2.10.3 directory transversal et redos

https://security.snyk.io/package/npm/jquery/1.11.0 XSS & prototype pollution

https://security.snyk.io/package/npm/angular/1.5.9 XSS, dos, redos, prototype pollution, JSONP Callback Attack (jsonp)

Nous n'avons toutefois pas pu exploiter ces vulnérabilités car les conditions nécessaires n'étaient pas réunies. Cependant, cette situation souligne un manque de support et de mises à jour régulières de l'application.

Recommandation

Assurez-vous de garder les versions des technologies installées à jour et configurez un processus pour les correctifs et les mises à jour.

Les versions vulnérables doivent être mises à jour dans les 30 jours suivant la publication du correctif par l'éditeur.

Downloaded at Thu, 27 Nov 2025 10:54:21 GMT
Downloaded by muhammad.abdou@skillupmena.com

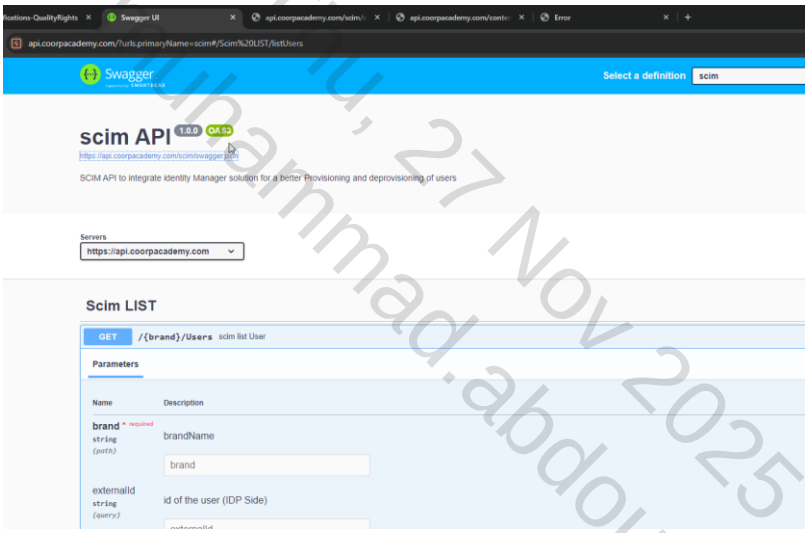
Réf	3 – Divulgation d'informations techniques	Sévérité	Faible
Exposition	Exploitable avec un compte sur l'application		
Risque(s)	Un attaquant peut avoir accès à des informations sur la conception du site ainsi que de l'API. Il pourrait se servir de ces informations afin de concevoir des attaques plus sophistiquées contre l'application.		
Type	Mauvaise configuration de la sécurité https://owasp.org/Top10/fr/A05_2021-Security_Misconfiguration/		
Commun à	Swagger / message d'erreur		

Description

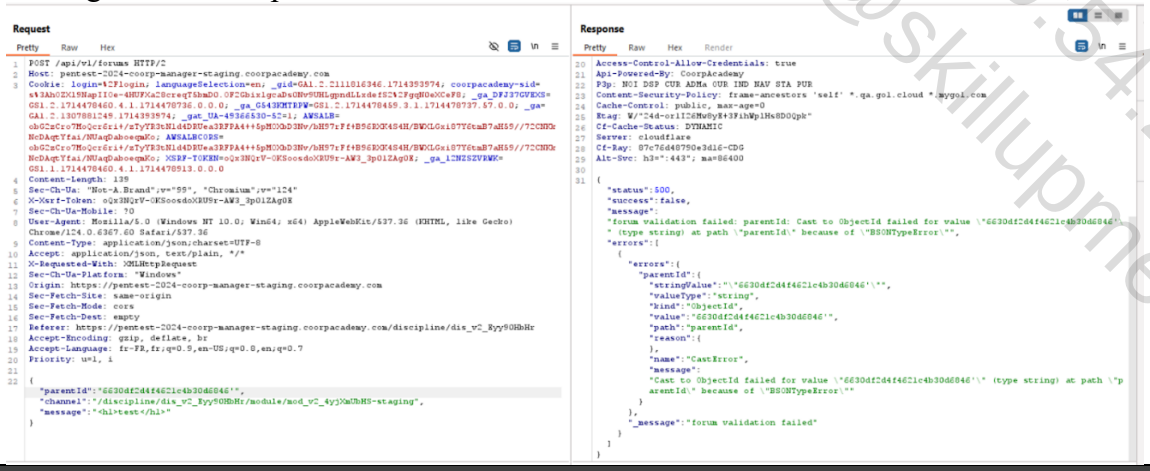
Des détails techniques sont exposés à plusieurs emplacements de la plateforme, tels que les messages d'erreur ou via le Swagger de l'API.

Exemples :

- Swagger exposé :



- Message d'erreur trop verbeux :



Recommandation

Désactivez l'interface Swagger UI ou y restreindre l'accès.

Rendre les messages d'erreur moins verbeux

Downloaded at Thu, 27 Nov 2025 10:54:21 GMT
Downloaded by muhammad.abdou@skillupmena.com

Réf	4 – Politique de mot de passe faible	Sévérité	Faible
Exposition	N/A		
Risque(s)	Des mots de passes faibles pourront être définis, et stockés pendant des durées arbitraires sur des supports non sécurisés, ce qui faciliterait la démarche d'un attaquant cherchant à deviner les mots de passe.		
Type	Identification et authentification de mauvaise qualité https://owasp.org/Top10/fr/A07_2021-Identification_and_Authentication_Failures/		
Commun à	Politique de mot de passe		

Description

Lors de la définition d'un mot de passe à la première connexion, la politique de mot de passe mise en place n'est pas assez robuste. La seule contrainte est le nombre de caractères (8).
Cela n'est pas suffisant pour protéger les mots de passe contre les attaques automatisées.

Recommandation

Définir une politique de mots de passe comme suit :

- Utilisation de mots de passe complexes : 8 caractères minimum, avec au moins 3 classes de caractères (majuscules, minuscules, caractères spéciaux et chiffres)
- Interdiction d'utiliser des mots de passe semblables au nom d'utilisateur
- Mettre en place une procédure de renouvellement de mots de passe à intervalles réguliers
- Privilégier l'authentification multi-facteur

Downloaded at TN - 27 Nov 2025 10:54:21 GMT
Downloaded by mohammad.abdou@skilluptmena.com

VI. Conclusion

Afin de corriger la vulnérabilité énoncée dans ce rapport, nous recommandons d'implémenter des messages d'erreur générique afin d'éviter l'énumération d'utilisateur présent dans l'application.

- Mettre en place un contrôle côté serveur sur la fonction d'upload de fichiers
- Maintenir les composantes technologiques à jour
- Restreindre l'exposition des informations techniques
- Mettre en place un politique de passe forte

L'équipe Sécurité d'Intuity se tient à votre disposition pour revenir sur les recommandations proposées ou apporter un complément d'information sur ce test.

Downloaded at Thu, 27 Nov 2025 10:54:21 GMT
Downloaded by muhammad.abdou@skillupmena.com