



## Pop Quiz 2

Stage PFE – Entreprise **VOID**

**Stagiaire :** Hibat Allah Rguiti

22 février 2026

# 1. Exploration d'une extension PHP inconnue

## 1.1. 1. Lancement du conteneur

Dans un premier temps, j'ai lancé le conteneur Docker fourni à l'aide de la commande suivante :

```
~ % docker run --rm --init --platform linux/amd64 --name void-lab-php-ext hbahalouane/void-labs-php-ext ]
```

Ensuite, j'ai accédé au terminal du conteneur avec :

```
-----  
void@192 ~ % docker exec -it void-lab-php-ext bash
```

Cela m'a permis d'entrer dans l'environnement isolé contenant PHP et l'application `app.php`.

## 1.2. 2. Exécution du script

J'ai exécuté le script PHP présent dans le conteneur :

```
root@615606a79d00:/usr/src/myapp# php app.php  
-----  
Hello You!  
Find the the extension (.so) path and how it was loaded.-----
```

Le script appelle la fonction suivante :

```
\VoidLabs\canYouSeeMe();
```

Cette fonction n'étant pas une fonction native de PHP, j'ai supposé qu'elle provenait d'une extension.

## 1.3. 3. Identification des extensions chargées

Afin d'identifier l'extension responsable, j'ai listé les extensions PHP actives avec :

```

root@615606a79d00:/usr/src/myapp# php -m
[PHP Modules]
Core
ctype
curl
date
dom
fileinfo
[filter
hash
iconv
json
libxml
mbstring
mysqlnd
openssl
pcre
PDO
pdo_sqlite
Phar
posix
random
readline
Reflection
sample
session
SimpleXML
sodium
SPL
sqlite3
standard
tokenizer
xml
xmlreader
xmlwriter
zlib

```

Parmi les extensions affichées, j'ai remarqué la présence d'une extension nommée **sample**, qui ne correspond pas à une extension standard connue de PHP.

## 1.4. 4. Vérification de l'extension

Pour confirmer que l'extension **sample** était bien chargée, j'ai exécuté :

```

[root@615606a79d00:/usr/src/myapp# php -r "var_dump(extension_loaded('sample'));"
bool(true)

```

La commande a retourné :

```
bool(true)
```

Cela confirme que l'extension est bien active.

Enfin, afin de vérifier les fonctions exposées par cette extension, j'ai utilisé :

```
[root@615606a79d00:/usr/src/myapp# php -r "print_r(get_extension_funcs('sample'));"
Array
(
    [0] => VoidLabs\canYouSeeMe
)
```

Cette commande permet d'afficher les fonctions ajoutées par l'extension.

## 1.5. Conclusion

L'extension responsable de l'ajout de la fonction `\VoidLabs\canYouSeeMe()` est l'extension `sample`.

Cette découverte a été réalisée en analysant les extensions chargées par PHP et en identifiant celle qui n'était pas standard dans l'environnement.

## 2. Bonus : Création et intérêt d'une extension PHP

### 2.1. Comment créer une extension PHP

Pour créer une extension PHP similaire, on peut suivre les étapes suivantes :

- Écrire l'extension en C ou un autre langage compatible en définissant les fonctions à exposer à PHP.
- Déclarer les fonctions dans la structure du module afin que le moteur PHP puisse les reconnaître.
- Compiler l'extension avec les outils PHP (`phpize`, `./configure`, `make`, `make install`) pour obtenir un fichier `.so` (Linux).
- Activer l'extension dans le fichier `php.ini` avec `extension=mon_extension.so`.
- Redémarrer PHP pour que les nouvelles fonctions soient disponibles dans les scripts.

### 2.2. Intérêt d'une extension PHP

Une extension PHP permet d'ajouter des fonctionnalités supplémentaires au langage, comme de nouvelles fonctions ou l'accès à des bibliothèques système. Elle peut également améliorer les performances, car le code compilé s'exécute plus rapidement qu'un script PHP pur. Les extensions sont particulièrement utiles pour les traitements intensifs, la manipulation de grandes données.