# Javascript - Code source

## Description

**Les scripts JavaScript peuvent parfois contenir des informations sensibles qui permettent à un attaquant d'obtenir des informations utiles pour accéder à des ressources protégées de l'application. Par exemple, si une authentification est implémentée en JavaScript (ce qui est évidemment à proscrire), il sera possible pour un attaquant de “reverse” le code, c'est à dire l'analyser pour en comprendre le fonctionnement, et ainsi s'authentifier de manière illégitime. Les scripts JavaScript peuvent parfois contenir des mots de passe obfusqués ou des clefs d'API, qu'un utilisateur malveillant pourrait utiliser.**

## Pré-requis d'exploitation

Pour exploiter cette vulnérabilité, il est nécessaire d’avoir accès à une application qui utilise du JavaScript contenant des informations sensibles.

### Connaissances nécessaires

- Connaissances de base du langage JavaScript.

### Outils nécessaires

- Avoir accès à une console de navigateur (Firefox, Chrome).

## Flux d'exécution

### Explorer

Naviguer sur l’application pour énumérer les endpoints et/ou pages de l’application, puis identifier les scripts JavaScript embarqués dans chaque page.

### Expérimenter

Analyser les scripts identifiés et essayer de les désobusquer si nécessaire afin de découvrir d'éventuelles informations sensibles.

### Exploiter

## Conséquences potentielles

Une exploitation réussie de ce type de vulnérabilité peut permettre :

- L'accès à des informations sensibles tel que des identifiants de connexion ;
- L'aide dans la recherche de vulnérabilité en énumérant le fonctionnement ou la/les version(s) des composants de l'application.

## Contre-mesures

Les contre-mesures suivantes peuvent être mises en œuvre :

- Ne pas inclure d'informations sensibles, même obfusquées, dans les scripts JavaScript ;
- Utiliser un pare-feu applicatif pour bloquer les requêtes HTTP non autorisées ou suspectes.

# Comment cela fonctionne

Les scénarios suivants peuvent être joués via l'exploitation de cette vulnérabilité :

- L'utilisation d'informations confidentielles pour se connecter sur un espace protégé par un mot de passe.
- L'énumération d'informations sensibles sur le serveur ou l'application afin de mener une attaque.

## Exemple 1

```
<script type="text/javascript">
function login(){
  pass=prompt("Enter password");
  if ( pass == "pass123456789" ) {
    window.location="/admin";
  else {
    alert("Wrong password !");
  }
}
</script>
```

Dans cet exemple, le code source de l'application expose un mot de passe d'authentification en clair dans le JavaScript. Un attaquant peut facilement récupérer ces informations sensibles et les utiliser pour accéder au système.

# Références

URL :

- https://www.reveillere.fr/M2WEB/cours/JavaScript.pdf
- https://profdoc.iddocs.fr/IMG/pdf/billiejoe_javascript_fiches.pdf
- https://www.tutorialspoint.com/javascript/javascript_tutorial.pdf

# Retour fiches vulnérabilités

- Cyber fiches vulnérabilités