JavaScript - Authentification

Description

L'utilisation abusive de scripts d'authentification en JavaScript consiste à utiliser des scripts malveillants ou non autorisés pour contourner les contrôles d'authentification et accéder à des ressources protégées ou sensibles. Les scripts d'authentification en JavaScript peuvent être utilisés pour automatiser l'authentification, falsifier des informations d'identification ou déjouer les contrôles de sécurité mis en place pour protéger les ressources.

Pré-requis d'exploitation

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

Connaissances nécessaires

- Connaissances de base du langage JavaScript ;
- Connaissances en matière de sécurité liée à l'authentification.

Outils nécessaires

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

Flux d'exécution

Explorer

Naviguer sur l'application afin d'identifier le/les script(s) d'authentification JavaScript et essayer de comprendre comment il(s) fonctionne(nt).

Expérimenter

Tester les champs de formulaire d'authentification en y injectant du code tout en observant comment l'application réagit.

Exploiter

Conséquences potentielles

Une exploitation réussie de ce type de vulnérabilité peut permettre :

- L'accès à des données sensibles ;
- L'accès à un espace restreint de l'application.

Contre-mesures

Les contre-mesures suivantes peuvent être mises en œuvre :

- Utiliser une base de données ou un service d'authentification tiers pour stocker les noms d'utilisateur et les mots de passe autorisés au lieu de les stocker en clair dans un script ;
- Chiffrer les mots de passe avant de les stocker dans la base de données ou le service d'authentification ;
- Utiliser l'authentification par jeton ou l'authentification à deux facteurs pour renforcer la sécurité de l'application.

Comment cela fonctionne

Les scénarios suivants peuvent être joués via l'exploitation de cette vulnérabilité : -* Retrouver les identifiants de connexion en clair à partir du script d'authentification. -* Obtenir un accès sur l'espace restreint /flag.

Exemple 1

Voici un exemple d'un script vérifiant un identifiant et un mot de passe en clair :

```
function login() {
  if (username == 'admin' && password == 'Str0ngP4ss') {
    window.location = '/admin/';
  } else {
    window.location = '/login/';
  }
}
```

Exemple 2

Voici un exemple d'un script d'authentification avec les noms d'utilisateurs et les mots de passe en clair :

```
// Tableau contenant les noms d'utilisateur et les mots de passe autorisés
const users = [
  {
    username: 'utilisateur1',
    password: 'motdepasse1'
  },
  {
    username: 'utilisateur2',
    password: 'motdepasse2'
  }
];

// Récupération des champs de formulaire de connexion
const usernameField = document.getElementById('username');
const passwordField = document.getElementById('password');
const loginButton = document.getElementById('login-button');

// Fonction qui vérifie si les champs de formulaire sont remplis et active le bouton de connexion si
c'est le cas
function checkForm() {
  if (usernameField.value && passwordField.value) {
    loginButton.disabled = false;
  } else {
    loginButton.disabled = true;
  }
}

// Écouteurs d'événements pour vérifier le formulaire chaque fois que les champs de formulaire sont
modifiés
usernameField.addEventListener('input', checkForm);
passwordField.addEventListener('input', checkForm);

// Écouteur d'événement pour soumettre le formulaire lorsque le bouton de connexion est cliqué
loginButton.addEventListener('click', function(event) {
  event.preventDefault();

  // Vérification de l'existence de l'utilisateur dans le tableau "users"
  const user = users.find(u => u.username === usernameField.value && u.password ===
passwordField.value);

  if (user) {
    // Connexion réussie, redirection vers la page flag
    window.location.replace('/flag');
```

```
  } else {
    // Afficher un message d'erreur à l'utilisateur
    alert('Nom d\'utilisateur ou mot de passe incorrect');
  }
});
```

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