Rapport de Red Team

Matthieu SEGUY et Line CUVELIER

Groupe Numa :

À la suite de différents tests réalisés sur le site de Numa, aucune faille exploitable n'a été identifiée lors des vérifications suivantes :

- Test de Broken Access Control
- Test injection XSS
- Test d'injection SQL
- Test des défaillances d'authentification (cookies)
- Test de présence d'un mot de passe administrateur par défaut
-

Remarque : L'utilisation de Cloudflare Zero Trust constitue une bonne pratique en matière de sécurité.

Groupe Tarek et Ewan :

Pour le groupe de Tarek et Ewan, nous avons identifié une mauvaise pratique. Lorsqu'un administrateur accède à la page rst.php afin de réinitialiser un mot de passe, le lien généré contient le hash du login de l'utilisateur concerné.
Si un attaquant parvient à réaliser une élévation de privilèges, il peut alors récupérer le hash des utilisateurs et réinitialiser leurs mots de passe.

En revanche, aucun problème n'a été identifié lors des tests suivants :

- Test de Broken Access Control
- Test d'injection XSS
- Test d'injection SQL
- Test des défaillances d'authentification (cookies)
- Test de présence d'un mot de passe administrateur par défaut

Groupe Ange et Armelle :

Dans le fichier Auth.php, une session est démarrée et le script vérifie si le navigateur possède un cookie nommé authbypass. Si ce cookie est présent, sa valeur est directement injectée dans la session comme userid.

Autrement dit, la session considère la valeur du cookie comme étant l'identifiant de l'utilisateur courant.

Le code accepte donc une valeur de cookie non vérifiée et l'enregistre dans la session comme identifiant utilisateur. Un attaquant peut ainsi ajouter ce cookie manuellement et se faire passer pour n'importe quel utilisateur, y compris un utilisateur inexistant.

Un attaquant peut également se connecter en tant qu'administrateur et effectuer une élévation de privilèges s'il connaît l'identifiant de l'admin.

```
const COOKIENAME = 'authbypass';

protected function __construct(){
    session_start();
    if(isset($_COOKIE[self::COOKIENAME])) {
        $this->log($_COOKIE[self::COOKIENAME]);
    }
}
```

De plus, lors de la tentative de connexion au site, la requête envoyée à la base de données est vulnérable à l'injection SQL. Sachant qu'un compte nommé admin existe, l'utilisation de la charge utile suivante :
admin" OR "1"="1
permet de se connecter au site en tant qu'administrateur.

```
public function subscribe($login, $pwd) {
    global $db;
    $hash = password_hash($pwd, PASSWORD_DEFAULT);
    $q = 'insert into users values(null, "'.addslashes($login).'", "'.$hash.'", 0)';
    $db->query($q);
}
```

```php
public function tryLog($login, $pwd): bool {
    global $db;
    $q = 'select * from users where login="'.$login.'"';
    $stmt = $db->query($q, PDO::FETCH_ASSOC);
    $user = $stmt ? $stmt->fetch() : null;

    if($user && password_verify($pwd, $user['pwd'])) {
        $this->log($user['id']);
        return true;
    }

    return false;
}
```

```php
public function resetPwd($id, $code, $newPwd) {
    global $db;
    $hash = password_hash($newPwd, PASSWORD_DEFAULT);
    $q = 'update users set pwd="'.$hash.'" where id="'.$id.'" and pwd="'.$code.'"';
    $db->query($q);
}
```

En revanche, aucun problème n'a été identifié lors des tests suivants :

- Test de Broken Access Control
- Test d'injection XSS
- Test de présence d'un mot de passe administrateur par défaut