TP-B3 : Attaque par porte dérobée (Backdoor)

1/ Pour identifier les portes dérobées ou vulnérables dans le code de l'annexe proposée :

```
<?php
$conn = new mysqli(« localhost », « root », « », « application_db ») ;

if ($conn → connect_error){
        die(« erreur de co » . $conn→connect_error) ;
}
$username = $_POST[‘username’] ;
$password = $_POST[‘password’] ;

$query = « select * from users where username = ‘$username’  and password = ‘$password’ » ;
$result = $conn→query($query) ;

if ($result → num_rows > 0){
        echo « bienv » . $username. « ! » ;
}
else{
        echo « Nom ou mdp non »
}
$conn→close() ;
?>
```

L'insertion des deux variables $username et $password rend la requêtes vulnérable a des attaques notamment des attaques par injection SQL.

2/ Pour sécuriser le code, il faudra utiliser une requête préparé afin d'éviter une attaques par injection SQL.

6. Simulation d'une attaque de porte dérobée.

Scénario : Une application web utilise un formulaire d'inscription demandant au client d'entrer ses informations à caractère personnel. Malgré les dispositif de sécurité utilisé, la base de donnée est attaquer.

Code vulnérable : $query = « select * from Utilisateurs where Adresse = ‘$adresse’ and Numero = ‘$telNumero» ;

Démonstration de l'attaque : L'attaquant interagit avec le formulaire d'inscription et peut injecter des données malveillantes dans les champs Adresse et Numero.

Correction et sécurisation :
On utilisera une requête préparée afin d'éviter une attaque dans les jours a venir.

```
$conn = new mysqli (« localhost », « root »,  « », « application_db ») ;

$query = "SELECT *
FROM Utilisateurs
WHERE Adresse = ? AND Numero = ?";

$secur = $conn->prepare($query);
$secur->bind_param($adresse, $telNumero);
$secur->execute();
```