La superglobale Session

# Introduction

La superglobale session est utilisé pour stocker des informations dans la session de l’utilisateur.

Un utilisateur peut avoir une session même s’il n’est pas connecté via une page de login.

Donc un simple visiteur peut aussi disposer d’une session.

Les données enregistrer dans la session seront consultable/modifiable depuis n’importe quelle page du site (tant que l’utilisateur a bien une session).

Les données de la session sont enregistrées sur le serveur, a chaque session on attribut un identifiant, du coté client seul l’id de la session est présent en tant que cookie.

Une faille XSS peut permettre a du JS de communiquer à un tier (un hacker) votre id de session et peut donc se faire passer pour vous tant que la session n’expire pas.

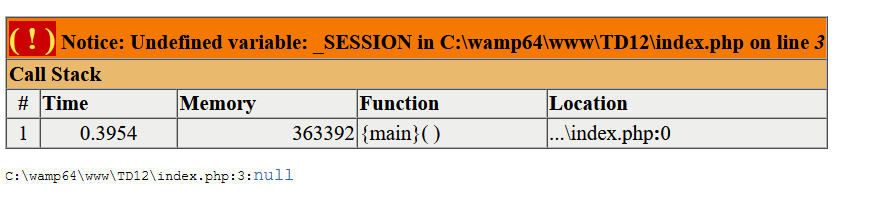
On commence par essayer de faire un var\_dump de la session :

<?php

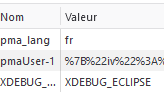
var\_dump($\_SESSION);

?>

On voit alors le message suivant :



En effet tant que la session n’est pas démarrée nous ne pouvons pas utiliser la superglobale session. Avant de démarrer la session consultons nos cookies.



Pour l’instant j’ai seulement 3 cookies.

Maintenant si on démarre la session.

<?php

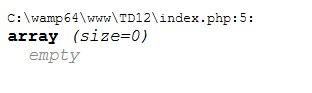
//Permet de generer une session

session\_start();

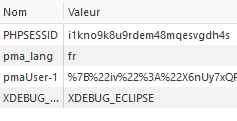
var\_dump($\_SESSION);

?>

On ne voit plus le message d’erreur, on nous indique que pour l’instant la session est un array vide.



Si je consulte mes cookies on voit apparaitre un nouveau cookie :



PHPSESSID est un cookie qui stocke l’id de ma session. Si je me connecte à la page dans le même navigateur je vais avoir le même numéro. En revanche si je me connecte dans un autre navigateur ou en mode anonyme je vais avoir un autre numéro de session. Normalement une session expire au bout de 180 minutes.

# Enregistrer une valeur dans la session

Comme dans n’importe quelle array je peut enregistrer une valeur dans ma session généralement on veut associer la clé manuellement.

<?php

//Permet de generer une session

session\_start();

//Enregistrer des données dans la session

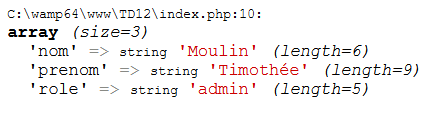
$\_SESSION["nom"]="Moulin";

$\_SESSION["prenom"]="Timothée";

$\_SESSION["role"]="admin";

var\_dump($\_SESSION);

?>



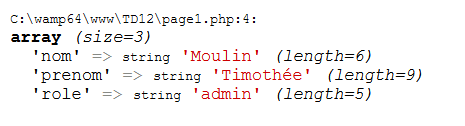
Pour l’instant je suis sur la page index.php mais si je crée une nouvelle page page1.php.

<?php

session\_start();

var\_dump($\_SESSION);

?>



Je remarque que les informations sont toujours enregistrées dans ma session.

Je peux donc écrire le code suivant dans page1.php :

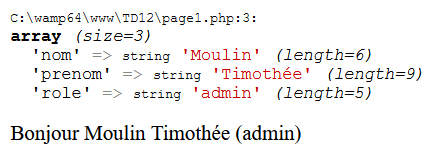
<?php

session\_start();

var\_dump($\_SESSION);

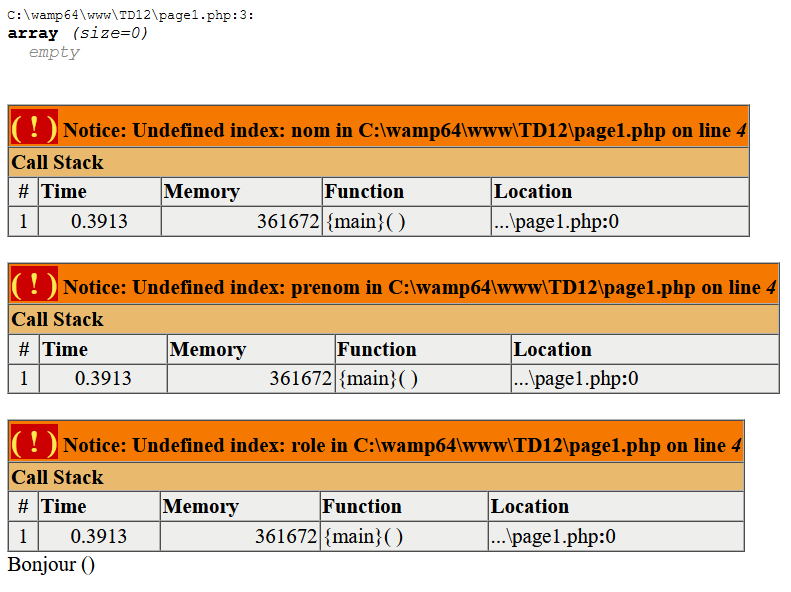
echo "Bonjour ".$\_SESSION["nom"]." ".$\_SESSION["prenom"]." (".$\_SESSION["role"].")<br>";

?>



Cela veut dire que si j’ai stocker des valeurs dans ma session les valeurs seront accessible sur toute les pages a partir du moment ou les pages utilisent bien session\_start().

Si j’ouvre dans un autre navigateur ou en mode anonyme directement la page1.php par contre on va obtenir des erreurs car le nom, prénom, et rôle n’auront pas était enregistrer dans cette session.



Pour éviter ce problème je peux me servir de isset sur la page1.php

<?php

session\_start();

var\_dump($\_SESSION);

if(isset($\_SESSION["nom"],$\_SESSION["prenom"],$\_SESSION["role"]))

{

    echo "Bonjour ".$\_SESSION["nom"]." ".$\_SESSION["prenom"]." (".$\_SESSION["role"].")<br>";

}

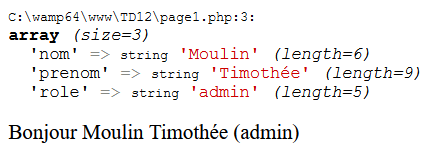
else{

    echo "Bonjour visiteur <br>";

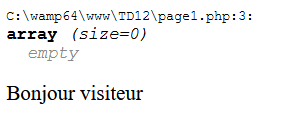
}

?>

Si je suis passée à un moment sur index.php je vais voir :



En revanche si je ne suis jamais passé par index.php je vais voir :



Normalement c’est lors de l’authentification que on enregistre les informations de l’utilisateur.

Nous allons donc modifier la page index.php :

<?php

//Permet de generer une session

session\_start();

if(isset($\_SESSION["email"]))

{

    echo "Bonjour ".$\_SESSION["email"]."<br>";

}

else{

    echo "Bonjour visiteur <br>";

}

var\_dump($\_SESSION);

?>

Nous allons créer une page login :

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Login</title>

</head>

<body>

    <h1>Login</h1>

    <form action="traitementLogin.php" method="post">

        <label for="">Email</label>

        <br>

        <input name="email" type="email">

        <br>

        <label for="">MDP</label>

        <br>

        <input name="mdp" type="password">

        <br>

        <button>Envoyer</button>

    </form>

</body>

</html>

Page traitementLogin.php

<?php

session\_start();

//Etape 1 : verifcation que les données sont bien transmisent

if(isset($\_POST["email"],$\_POST["mdp"])){

    //Etape 2 : netoyage des données

    $email=filter\_var($\_POST["email"],FILTER\_SANITIZE\_EMAIL);

    $mdp=filter\_var($\_POST["mdp"],FILTER\_SANITIZE\_FULL\_SPECIAL\_CHARS);

    //cyrpter le mdp

    $mdp=hash("sha256",$mdp);

    //Etape 3  : Validation des données

    // Dans le cas de du login la validation est faite en comparant les valeurs du form a la BDD

    // Ici nous n'avons pas encore de BDD

    $estValide=true;

    if($estValide)

    {

        //Etape 4 : enregistrement des valeurs dans la session

        // on peut recuperer des informations en plus quand on utilse la bdd

        // comme le nom le prenom, le role ...

        $\_SESSION["email"]=$email;

        $\_SESSION["mdp"]=$mdp;

        //Etape 5 : Redirection (retirer pour debugger)

        header("location:index.php");

    }

    else{

        echo "email ou mdp incorecte";

    }

}

else{

    echo "Pas de données transmisent";

}

?>

# Supprimer une session :

On peut supprimer des données d’une session de plusieurs manières :

Si on cherche juste a supprimer une seul donnée et pas le reste on peut utiliser unset()

session\_start();

unset($\_SESSION["mdp"]);

Ici on supprime seulement le mdp et pas le reste de la session.

Si on cherche à supprimer les données de toute la session :

<?php

session\_start();

//remplace la session par un array vide

$\_SESSION=[];

//destruction de la session

session\_destroy();

?>