Securing the System's Parameter using SESSION

So, why use SESSION instead? Don't you already have the user login facilities?

User login facility is just the guard upon the entrance. A thief doesn't really need permission from the guard to enter. They simply hope over the fence, cut a hole in the fence, or dig a passage under the fence to sneak in to a premise.

So, for the sake of the premise, we should position a guard for each of the building in the premise compound. It would act as a double layer protection against the unauthorized person. Although a thief could possibly bypass the security guard at the entrance, or sneak in through a hole in the fence, they need to bypass another security personel to enter a building.

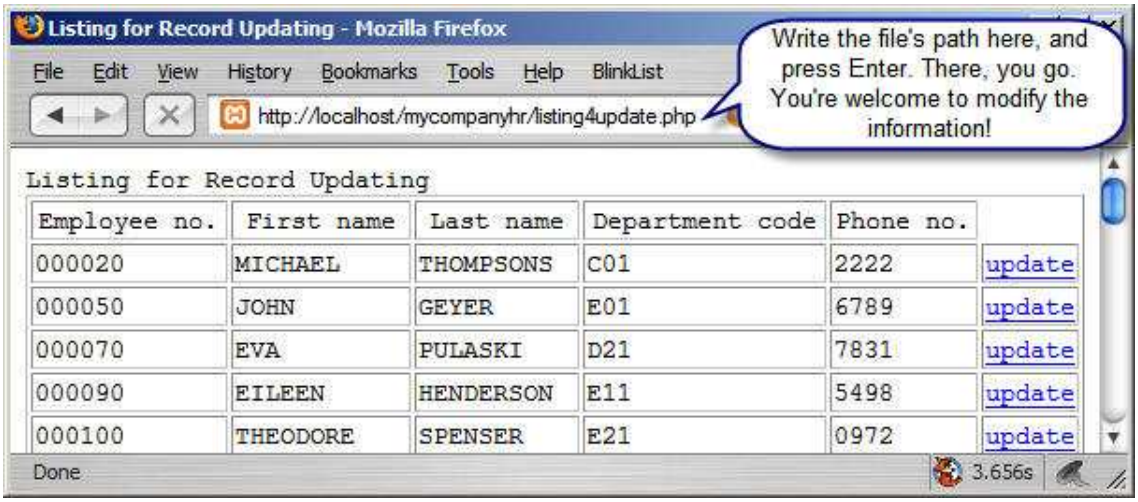
As for the web application concern, we have implemented the first layer protection by applying the user login process. This login will determine only the authorized personel is allowed to access the menu page. This menu page consist of all the links to another pages that provides administration facilities.

However, a hacker doesn't really need the menu page. Anybody can access the administration application if he/she knows the path and the file name of the exact page. For example, the page to update an employee details is *listing4update.php* (we are still using the *mycompanyhr* project for this discussion). We can simply view the address of the page from the address bar (that is why it is important not to disclose the file's path while viewing the file's content in the browser-this issue will be discussed in the later chapter).

So, try typing this path;

http://localhost/mycompanyhr/listing4update.php

in the address bar with the project sample installed in your localhost. Although you did not login (by providing username and passwrod), still you could possibly view the content and are permitted to even update the information.



Applying check session in the system.

Before that lets modify the login.php file. Change the form's action to *verify-session.php*.

```
<html>
<head>
<title>MyCOMPANYHR-Login Form</title>
</head>
<body>

<strong>Login to MYCOMPANYHR system</strong>
<form name="formlogin" method="post" action="verify-session.php">
<table width="400"  border="0">
  <tr>
    <td>Employee No</td>
    <td><input name="EMPNO" type="text" maxlength="6">
      Eg: 999999 </td>
  </tr>
  <tr>
    <td>Password</td>
    <td><input name="PASSWORD" type="password"></td>
  </tr>
  <tr>
    <td> </td>
    <td><input name="submit" type="submit" value="Login"></td>
  </tr>
</table>
</form>

</body>
</html>
```

In order to implement the session control to enhance you system's security, register the connection details at the point where the system has verify the username exists and the password match. So we need to get back on the verification script as in *verify.php* page in Chapter 14.

Rename the file with *verify-session.php*.

```
<?php
//session begins
session_start(); //set the users full name
$EMPNO=$_POST['EMPNO'];
$PASSWORD=$_POST['PASSWORD'];
include "connection.php";
$sql="select * from adminusers where EMPNO='$EMPNO'";
$rs=mysqli_query($db, $sql);
?>
<html>
<head>
<title>MyCOMPANYHR-verify</title>
</head>

<body>
<strong>Verify employee number and password</strong><br>
```

```
<?php
if(mysqli_num_rows($rs)==1){ //found one user
      $record=mysqli_fetch_array($rs);
      $DBPASSWORD=$record['PASSWORD'];//password from database
      $USERPASSWORD=md5($PASSWORD);//MD5 password key-in by user
      if($DBPASSWORD==$USERPASSWORD){
            //compare password from database against password entered by user
            echo "Username and password match,<br>";
            echo "WELCOME $EMPNO !!!<br>";
            //register the session for the user
            $_SESSION["sessionid"]=session_id();
            $_SESSION["empno"]=$EMPNO;
            echo "<a href='menu-session.php'>ADMIN Menu </a><br>";
      }
      else{
            echo "Username found, but password NOT match,<br>";
            echo "<a href='javascript:history.back()'>re-enter
                  password</a><br>";
      }
}
else{
      echo "Username NOT found,<br>";
      echo "<a href='javascript:history.back()'>re-login</a><br>";
}
?>
</body>
</html>
```

After this point of verification, we need to implement session checking for all the administration pages.

Try this script for session checking mechanism. And then include this *checksession.php* file in all the pages involved.

```
<?php
//this script is to check session to verify user login
session_start();
if(!isset($_SESSION["empno"])){ //if session NOT set
      echo "You are not logged in,
            <a href='login-session.php'>click here to login.</a>";
            exit(0);
}
?>
```

`if(!isset($_SESSION["empno"]))` – this line checks whether the session for the user with the particular employee number is registered in the server session. If the session is not set, then user will be asked to go to the login page.

`exit(0);` – this statement terminate the execution the current page. This is to make sure the page is not sent to the unauthorized user.

Reconfigure the *menu.php*, and save as *menu-session.php*.

```
<?php
include "checksession.php";
include "connection.php";
?>
```

```
<html>
<head>
<title>MyCOMPANYHR-menu</title>
</head>

<body>
<strong>Menu for MyCOMPANYHR administration</strong><br>
<?php
//script to display employees information
$empno=$_SESSION['empno'];
/*this SQL commad will fetch the employee's administration level,
firstname, lastname, workdept, deptname in their respective tables*/
$sql="SELECT
           adminusers.EMPNO,
           adminusers.LEVEL,
           employee.FIRSTNAME,
           employee.LASTNAME,
           employee.WORKDEPT,
           department.DEPTNAME
     FROM adminusers
           INNER JOIN employee
                ON adminusers.EMPNO = employee.EMPNO
           INNER JOIN department
                ON employee.WORKDEPT = department.DEPTNO
     WHERE adminusers.EMPNO='$empno'";
$rs=mysqli_query($db, $sql);
$record=mysqli_fetch_array($rs);
$_SESSION['level']=$record['LEVEL'];
$_SESSION['fullname']=$record['FIRSTNAME']." ".$record['LASTNAME'];
$_SESSION['workdept']=$record['WORKDEPT'];
$_SESSION['deptname']=$record['DEPTNAME'];

?>
Welcome, <? echo "$empno ".$_SESSION['fullname']?> <br>
From department:
     <? echo $_SESSION['workdept']." ".$_SESSION['deptname']?> <br>

<?php
//this menu displays depending on the users level
//if level is 1, full access
//if level is 2, limited access
if($_SESSION['level']==1){
?>
     Menu : full access administration<br>
     1. <a href="searchform.php">Search employee</a><br>
     2. <a href="forminsert.php">Insert a new employee</a><br>
     3. <a href="listing4update.php">Update information of existing
          employee</a><br>
     4. <a href="listing4delete.php">Delete existing
          employee</a><br>
     5. <a href="logout.php">Logout</a><br>
<?php
}
else if($_SESSION['level']==2){
?>
```

```
      Menu : limited access user <br>
      1. <a href="searchform.php">Search employee</a><br>
      2. <a href="formupdate-personal.php">Update personel
           information</a><br>
      3. <a href="logout.php">Logout</a><br>
<?php
}

//display footer
include "footer.php";
?>
</body>
</html>
```

So what is in *footer.php* file? The file contains the script to display the current user's information, a link to *menu-session.php*, and also a link to the *logout.php* file. The purpose of this template is to provide easier navigation for the user.

Filename; *footer.php*.

```
<style type="text/css">
<!--
.Copyright {
      color: #FF6600;
      font-weight: bold;
      font-family: Arial, Helvetica, sans-serif;
      font-size: 12px;
}
.style2 {font-size: 14px}
.style4 {font-size: 12px}
-->
</style>
<table width="100%"  border="0">
  <tr>
    <td><div align="center" class="Copyright">
      <div align="left">Copyright   <? echo date('Y');?> &copy; All
Rights Reserved to Kerul </div>
    </div></td>
  </tr>
</table>
<table width="100%"  border="0" bgcolor="#FF6600">
  <tr>
    <td         bordercolor="#FF9933"         bgcolor="#FF9933"><div
align="left"><span class="style2">
      <a href="menu-session.php"> menu</a> -
      <a href="logout.php">logout</a> </span>
      <strong>MyCOMPANYHR</strong>-
      <span class="style4">user (<? echo $_SESSION['fullname']; ?>)
      </span></div></td>
  </tr>
</table>
```

Logout facility. – this is to destroy the session information stored in the server. It is a very important to delete all the session information should the user intend to leave the web application.

Filename; logout.php

```
<? //to destroy session
     session_start();
     session_destroy();
?>
<html>
<head>
<title>MyCOMPANYHR - Logout </title>
</head>

<body>
MyCOMPANYHR<br>
Logged out, TQ...<br>
<a href="login-session.php">Re-ENTER?</a><br>
</body>
</html>
```

The next step is to include *checksession.php* and *logout.php* in all the pages involved in insert, update and delete facilities. If you think the data is highly confidential, you might also include the files in the listing or search facilities. Normally searching or record listing is open to public.

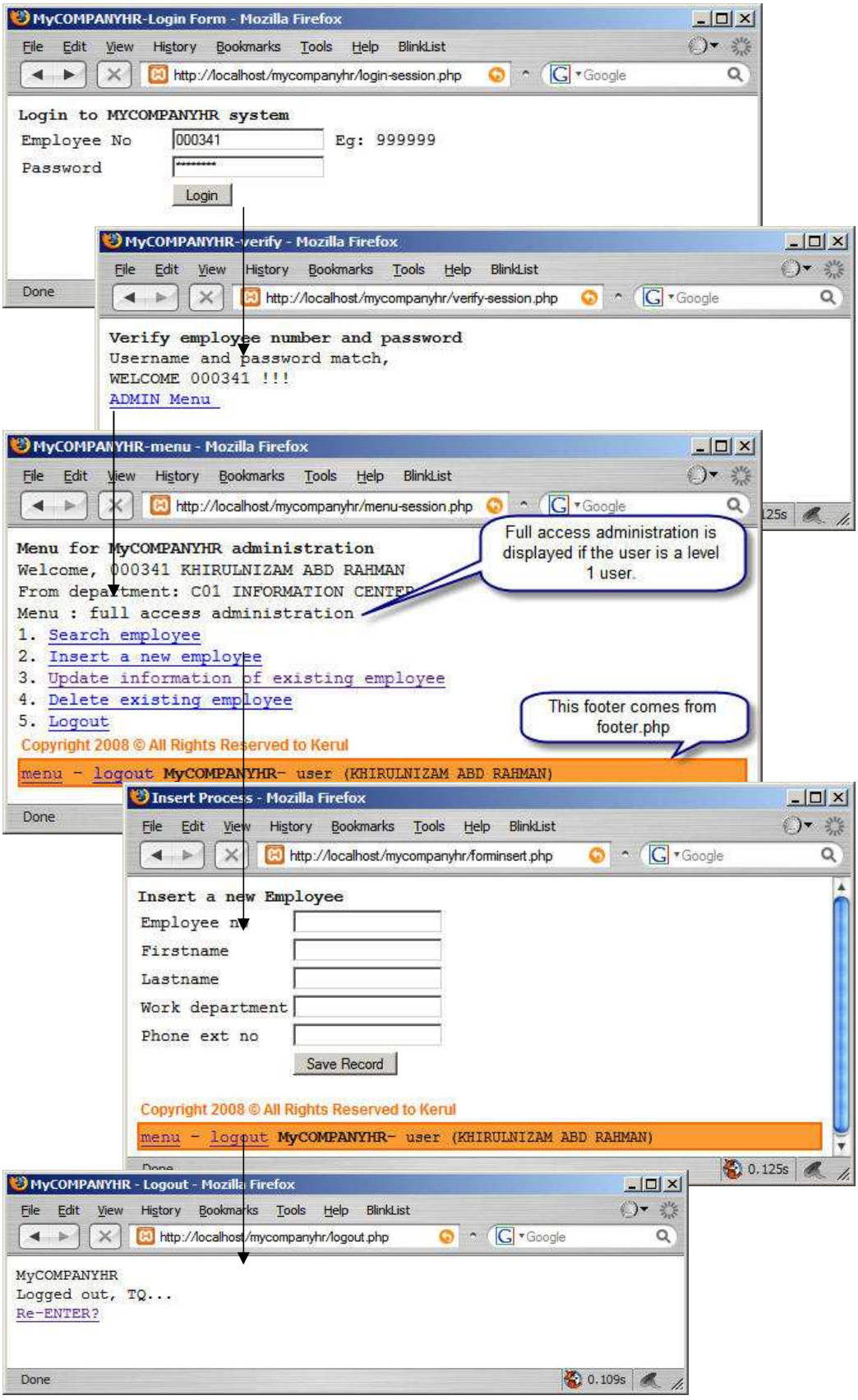
Let insert both files to the *forminsert.php*.

```
<?php include "checksession.php"; ?>
<html>
<head>
<title>Insert Process</title>
</head>
<body>
<strong>Insert a new Employee </strong><br>
<form name="insert" action="insertrecord.php" method="get">

<table border="0">
  <tr>
    <td width="40%">Employee no</td>
    <td width="60%"><input name="EMPNO" type="text"></td>
  </tr>
  <tr>
    <td>Firstname</td>
    <td><input name="FIRSTNAME" type="text"></td>
  </tr>
  <tr>
    <td>Lastname</td>
    <td><input name="LASTNAME" type="text"></td>
  </tr>
  <tr>
    <td>Work department</td>
    <td><input name="WORKDEPT" type="text" maxlength="3"></td>
  </tr>
  <tr>
```

```
    <td>Phone ext no</td>
    <td><input name="PHONENO" type="text" maxlength="4"></td>
  </tr>
  <tr>
    <td> </td>
    <td><input name="save" type="submit" value="Save Record"></td>
  </tr>
</table>
</form>
<?php include "footer.php"; ?>
</body>
</html>
```

Screen shots.


MyCOMPANYHR-Login Form - Mozilla Firefox
http://localhost/mycompanyhr/login-session.php
Login to MYCOMPANYHR system
Employee No
000341
Eg: 999999
Password
Login
MyCOMPANYHR-verify - Mozilla Firefox
http://localhost/mycompanyhr/verify-session.php
Verify employee number and password
Username and password match,
WELCOME 000341 !!!
ADMIN Menu
MyCOMPANYHR-menu - Mozilla Firefox
http://localhost/mycompanyhr/menu-session.php
Full access administration is displayed if the user is a level 1 user.
Menu for MyCOMPANYHR administration
Welcome, 000341 KHIRULNIZAM ABD RAHMAN
From department: C01 INFORMATION CENTER
Menu : full access administration
1. Search employee
2. Insert a new employee
3. Update information of existing employee
4. Delete existing employee
5. Logout
This footer comes from footer.php
Copyright 2008 © All Rights Reserved to Kerul
menu - logout MyCOMPANYHR- user (KHIRULNIZAM ABD RAHMAN)
Insert Process - Mozilla Firefox
http://localhost/mycompanyhr/forminsert.php
Insert a new Employee
Employee no
Firstname
Lastname
Work department
Phone ext no
Save Record
Copyright 2008 © All Rights Reserved to Kerul
menu - logout MyCOMPANYHR- user (KHIRULNIZAM ABD RAHMAN)
0.125s
MyCOMPANYHR - Logout - Mozilla Firefox
http://localhost/mycompanyhr/logout.php
MyCOMPANYHR
Logged out, TQ...
Re-ENTER?
Done
0.109s

Right after the logout process, try to type this path in the address bar;
http://localhost/mycompanyhr/forminsert.php

Unfortunately you cannot see the form to insert new employee. Why this is happening? The checksession.php restrict the user from viewing the page, since the user has logged out. The user need to login again in order to get the session registered.