

Protezione di pagine con PHP tramite variabili di sessione

Proteggere aree del proprio sito con il php non è troppo difficile. Si possono usare a tale scopo le variabili di sessione. L'esempio qui riportato utilizza quattro file:

- [login.php](#)
- [destroy.php](#)
- [check.php](#)
- [pasx.txt](#)

più tutte le pagine protette (nel nostro esempio solamente [inizia.php](#)). Il principio è piuttosto semplice: verificato il login, viene impostata una variabile di sessione, in particolare `autorizzato`, che verrà poi testata in ogni pagina da proteggere. Il controllo avviene tramite il minuscolo script `check.php`, che deve essere incluso in tutte le pagine riservate, prima del tag `<html>`. Tutti i nomi utente e password sono contenuti in un file di testo. Tale file (`pasx.txt`) in questo esempio, per semplificare, è leggibile in chiaro, ma si può aumentare notevolmente il livello di sicurezza criptando sia il nome utente che la password tramite, ad esempio, la codifica MD5: <http://www.php.net/manual/en/function.md5.php>

Un ulteriore grado di protezione può essere introdotto posizionando il file `pasx.txt` all'interno di una cartella dedicata, e rendendolo inaccessibile alla lettura via web tramite il pannello di controllo sui piani windows, e tramite il file `.htaccess` su linux. La soluzione ottimale è naturalmente quella di conservare i dati di accesso in un database.

La struttura di `pasx.txt` è la seguente:

pasx.txt

```
prova~::~prova
test~::~test
tentativo~::~tentativo
```

ogni riga del file contiene una coppia (userid,password) separata dai caratteri speciali ~::~

login.php

```
<?php
session_start();
if (isset($_POST["invio"])) {
    $puntatore = fopen("testi/pasx.txt", "r");
    $trovato = 0;
    while ((!feof($puntatore)) && (!$trovato)) {
        $linea = fgets($puntatore);
        $trovato = strstr($linea, $_POST["userid"]);
        $puntatore++;
    }
    fclose($puntatore);
    list($nomeutente, $password) = split("~::~", $linea);
    if (($trovato) && ($_POST["passwd"] == trim($password))) {
        session_register('autorizzato');
        $_SESSION["autorizzato"] = 1;
        $destinazione = "inizia.php";
    } else {
        $destinazione = "destroy.php";
    }
    echo '<script language=javascript>document.location.href="'. $destinazione. "'</script>';
} else {
    // HTML ?>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html>
<head>
<title>Prova Login php</title>
<meta name="Generator" content="EditPlus">

<meta name="Author" content="">
<meta name="Keywords" content="">
<meta name="Description" content="">

</head>
```



```
session_unset();
session_destroy();
?>
<script language="JavaScript">

document.location.href = "login.php"
</script>
```

Gli script mostrati in questo breve esempio non sono completi, e devono essere integrati nelle pagine desiderate. Qui di seguito un esempio di pagina protetta dalla variabile di sessione `autorizzato`:

```
inizia.php
<? include("check.php"); ?>
<!doctype html public "-//W3C//DTD HTML 4.0 Transitional//EN">
<html>
<head>
<title> Pagina di test - PHP login </title>

<meta name="Generator" content="EditPlus">
<meta name="Author" content="">
<meta name="Keywords" content="">
<meta name="Description" content="">

</head>
<body>
Esempio: login effettuato con successo
</body>
</html>
```