

Classe V^a Abacus – Analisi sistema di autenticazione basato su files

Problema

Ci poniamo, con questa esercitazione, il problema di realizzare un semplice sistema di autenticazione¹, capace di leggere la presenza (o meno) di una coppia login e password opportuna all'interno di un file di testo, opportunamente formattato².

Analisi di una soluzione

Spesso i sistemi di autenticazione dei siti si appoggiano ad un opportuno DB, che quasi sempre affianca i sistemi Web con pagine attive. In questa esercitazione vogliamo però utilizzare ancora i file per mostrare, che, se necessario, gli stessi possono essere utili a risolvere problemi di questo tipo almeno nei casi più semplici.

Il nostro sistema è pensato in modo che le pagine ASP possano controllare l'accesso di un utente autorizzato in base ad una lista, opportunamente formattata, presente in un file di testo (.txt).

Si prevederà quindi una maschera HTML opportuna per l'inserimento dei classici dati di login e password, successivamente questa pagina chiamerà una pagina che ha il compito di verificare che i dati inseriti corrispondano effettivamente a qualche utente autorizzato all'accesso, ed infine la pagina di controllo setterà una variabile di sessione con l'identificatore dell'utente che è entrato e rimanderà ad una pagina di login effettuato con successo o ad una che indica il fallimento delle operazioni di login.

Lo schema quindi nel complesso è piuttosto semplice e semplice è anche il sistema di ricerca di questo login e password nel file di testo. In realtà in tale file si inseriranno coppie login abbinate alle rispettive password e la pagina di controllo andrà a verificare che dato un certo input i due dati corrispondano proprio ad una delle coppie indicate. Se ciò accade l'utente ha effettuato l'accesso con successo (ed usualmente viene settata una apposita variabile di sessione che indica questo fatto, in modo opportuno).

Nei punti sottostanti stiamo ipotizzando che il file di testo sia formattato in questo modo:

- Il file non ha righe vuote
- In ogni riga è presente solo la login e, separata da uno spazio, la password
- Login e password, ovviamente, non hanno spazi interni

Schematizzando in una scaletta di punti:

- 0) Viene realizzato un file di testo opportunamente formattato di coppie login e password
- 1) L'utente carica la pagina per il login
- 2) L'utente inserisce login e password per l'accesso nella apposita pagina di login
- 3) I dati vengono spediti ad una pagina ASP per il controllo
- 4) La pagina apre il file di testo delle coppie login e password in lettura

1 Per autenticazione intendiamo il riconoscimento di un utente autorizzato all'accesso ad una parte protetta di un sistema, ad esempio una parte o tutta una applicazione Web.

2 Vale a dire formattato con regole certe ed opportune

- 5) Viene letta una riga e scomposta in login e password, individuando i due dati³
- 6) i dati letti dal file vengono paragonati a quelli inseriti dall'utente, ossia si confrontano le login e le password.
- 7) Se non c'è corrispondenza si passa a leggere un'altra riga del file
- 8) Se il file non ha più righe da leggere l'utente **non ha l'autenticazione**
- 9) Se c'è corrispondenza si termina il ciclo di lettura del file e si permette l'accesso dell'utente che quindi **è autenticato**

Questa per sommi capi la strategia delle operazioni su file per capire se esso abbia presentato credenziali valide o no. Il file di testo, in questo esempio, si presume sia compilato a parte e a mano. Ovviamente in un sistema "chiavi in mano" questo non sarebbe proponibile, ma ci dovrebbe essere la presenza di apposite pagine che permettano l'editazione di quanto contenuto nel file di testo delle login / password, di modo da poter modificare opportunamente il suo contenuto.

Il codice soluzione

La pagina per il login è banale e quindi non commentiamo il semplice codice HTML riportato sotto. La pagina può essere una pagina HTML pura:

```

...
<form action="autentica.asp" method="post">
<table width="326" height="204" border="0" align="center">
  <tr>
    <td width="139" height="57"><span class="Stile6">Login:</span></td>
    <td width="177"><input type="text" name="login" id="login" /></td>
  </tr>
  <tr>
    <td height="58"><span class="Stile6">Password:</span></td>
    <td><input type="password" name="passwd" id="passwd" /></td>
  </tr>
  <tr>
    <td height="81" colspan="2"><div align="center">
      <input type="submit" name="button" id="button" value="Entra" />
    </div></td>
  </tr>
</table>
</form>
...

```

Tale form invia dati, ad una pagina ASP, in questo esempio chiamata autentica.asp.

Il codice di tale pagina sarà il seguente⁴:

```

' legge i dati di autenticazione da verificare inseriti dall'utente
Login_Utente = Request.Form("login")
Passwd_Utente = Request.Form("passwd")

' istanzia oggetto FileSystemObject per operazioni sul file system
Set Fso = Server.CreateObject("Scripting.FileSystemObject")

' Istanza oggetto TextStream per operazioni su file di testo
Set Fto = Fso.OpenTextFile(Server.MapPath(".") & "\passwd.txt")

' finchè ci sono dati (righe) da leggere...
Do While Not Fto.AtEndOfStream
  'Legge login e password assieme (entrambe sulla stessa riga)

```

³ Questo è possibile solo se la formattazione dei contenuti del file è precisa e univoca

⁴ Questo codice è strettamente legato al tipo di formattazione del file vista precedentemente.

```

        LoginEPasswd = Fto.ReadLine()
'Operazioni per separare login e password
        PosSpazio = InStr(LoginEPasswd, " ")
        Login = Left(LoginEPasswd, PosSpazio - 1)
        Passwd = Right(LoginEPasswd, Len(LoginEPasswd) - PosSpazio)

        Response.Write Login & " "
        Response.Write Passwd & "<br>"

        ' se si verifica vuol dire che e' autenticato
        If (Login = Login_Utente) And (Passwd = Passwd_Utente) Then
            Session("NomeUtente") = Login
            Exit Do
        End If
    Loop

    'redirige a seconda dei casi (login effettuato oppure nessun login)
    If Not IsEmpty(Session("NomeUtente")) Then
        Response.Redirect "entrato.asp"
    Else
        Response.Redirect "nologin.html"
    End If

```

Come si vede nel codice si effettuano i seguenti passi:

- Si crea l'oggetto FileSystemObject per permettere ad ASP di operare sui file
- Si apre il file di testo passwd.txt, con l'apposito metodo dell'oggetto FileSystemObject, OpenTextFile(...), in lettura (per far ciò basta non specificare nessun ulteriore parametro oltre al path completo ed assoluto del file delle login e password.
- Si ripete la lettura di una riga alla volta dal file, quante volte necessario grazie ad un ciclo Do While
- Letta la singola riga come stringa, viene individuata la posizione dello spazio bianco che nella stessa separa login e password tramite la funzione InStr applicata in modo opportuno
- Appena individuato lo spazio è possibile rilevare la parte relativa alla login prendendo la parte di sinistra della stringa, ossia quella prima dello spazio e rilevare la password, prendendo la parte di destra della stringa considerando i caratteri a destra dello spazio.
- In realtà, dato lo spazio in una certa posizione pos, avremo che la lunghezza di ciò che precede lo spazio sarà $pos - 1$ e ciò che segue lo spazio è pari a $lunghezza\ della\ stringa - pos$. La lunghezza totale della stringa è calcolabile con la ben nota funzione BASIC Len(...), mentre la le parti di stringa volute possono essere rilevate a destra con la funzione Right (estrazione degli ultimi n caratteri della stringa) e della funzione Left (estrazione dei primi m caratteri della stringa)
- Una volta isolate login e password possiamo effettuare il solito confronto tra quelle lette dal file e quelle passate dall'utente. Se c'è corrispondenza il programma setta una variabile di sessione ed esce immediatamente dal ciclo Do While(...).
- Il ciclo si fermerà quando il programma ha raggiunto la fine del file di testo, grazie alla apposita proprietà applicata all'oggetto TextStream, AtEndOfStream, che diviene vera se la lettura del file ha raggiunto la fine dello stesso.

- Al seguito del ciclo si indirizza a seconda del risultato (sessione settata o no, ossia utente riconosciuto o no) la navigazione su una pagina che conferma l'autenticazione o su una che indica la non possibilità di autenticarsi con i dati inseriti.

Con leggere varianti è possibile modificare il file delle login e password in modo che risulti più semplice leggere e separare i due dati login e password, ma ciò comporta anche alcune modifiche al file di controllo, autentica.asp.

Specificamente se le login e le password sono una sotto l'altra a coppie, prevedendo due letture per ciclo, è possibile separare i dati già al momento della lettura degli stessi. In questo modo si può evitare del tutto di introdurre l'uso di funzioni sulle stringhe (quali InStr, Left, Right e Len), semplificando di molto il codice. In questo caso il codice della pagina di controllo della corrispondenza login e password sarà:

```
' legge i dati di autenticazione da verificare inseriti dall'utente
Login_Utente = Request.Form("login")
Passwd_Utente = Request.Form("passwd")

' istanzia oggetto FileSystemObject per operazioni sul file system
Set Fso = Server.CreateObject("Scripting.FileSystemObject")

' Istanzia oggetto TextStream per operazioni su file di testo
Set Fto = Fso.OpenTextFile(Server.MapPath(".") & "\passwd.txt")

' finchè ci sono dati (righe) da leggere...
Do While Not Fto.AtEndOfStream
    Login = Fto.ReadLine()
    Passwd = Fto.ReadLine()

    ' se si verifica vuol dire che e' autenticato
    If (Login = Login_Utente) And (Passwd = Passwd_Utente) Then
        Session("NomeUtente") = Login
        Exit Do
    End If
Loop

'redirige a seconda dei casi (login effettuato oppure nessun login)
If Not IsEmpty(Session("NomeUtente")) Then
    Response.Redirect "entrato.asp"
Else
    Response.Redirect "nologin.html"
End If
```

Anche la soluzione in cui login e password risultano fuse potrebbe sembrare percorribile; in essa si sceglierebbe di affiancare nel file login e password unendole assieme senza spazi. Successivamente concatenando gli analoghi dati di input. In realtà tale soluzione soffre però di un difetto almeno teorico. Infatti data la stringa unione data da login e password, potrebbe essere possibile accedere inserendo una login più breve e una password più lunga, o viceversa, che ovviamente rispetti il complessivo risultante da login+password. Per concretizzare il discorso ammettiamo di avere un utente che abbia come login *pippo* e come password *baudo*. Nel file la riga di testo relativa all'utente risulterà *pippobaudo* tutto attaccato, ma si può notare che in questo caso divenendo non distinguibile login e password, se noi inserissimo pip e pobaudo, avremmo comunque l'accesso pur non essendo corrette le credenziali.