

BEN'S HOBBYHOEKJE

Gebruikershandleiding voor:

Beperkte Password protectie met JavaScript

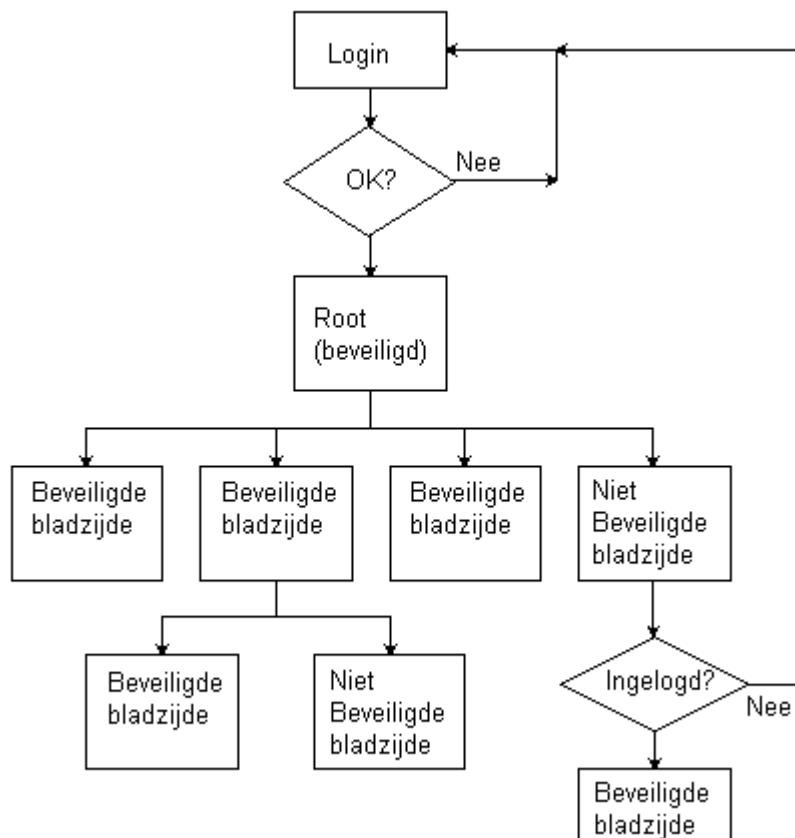
URL: <http://home.hccnet.nl/s.f.boukes/html-2/html-202.htm>

INHOUD :

1. Structuur van een beveiligde site
2. Login-formulier
3. Login geweigerd
4. Geldigheidsduur van een login-code
5. Een bladzijde beveiligen
6. Login-codes maken
7. Beperkingen van de beveiliging en het gebruik
8. Tenslotte...

1. Structuur van een beveiligde site.

Zie onderstaand schema.



Toegang wordt verkregen door in te loggen met een gebruikersnaam en een password. Je komt binnen op een vastgestelde pagina, de *root*. Van daaruit kun je naar andere, beveiligde en onbeveiligde bladzijden navigeren.

BEN'S HOBBYHOEKJE

Als het inloggen fout gaat, wordt een pagina getoond waarmee je het opnieuw kunt proberen. Er is geen beperking aan het aantal pogingen om in te loggen.

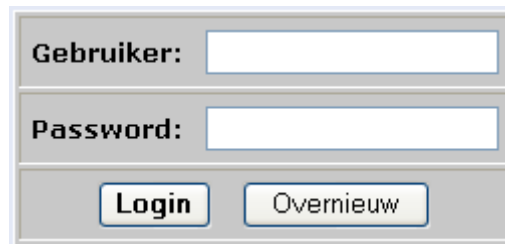
Niet-beveiligde bladzijden binnen het beveiligde deel kunnen direct (via deeplinks) worden benaderd. Als je echter vanuit een niet-beveiligde bladzijde naar een beveiligde bladzijde wilt moet je alsnog inloggen.

Er is geen beperking aan het aantal beveiligde bladzijden binnen het beveiligde deel van de website. Er kan echter binnen een website echter maar één beveiligd deel zijn (één beveiligde root).

Dit werkt alleen als de browser cookies accepteert. Als de browser cookies weigert verschijnt er een alert met een melding dat de toegang is geweigerd.

2. Login-formulier

Een user-name en een password zijn vereist. Je moet de bezoekers gelegenheid geven in te loggen op één duidelijke plaats in je website. Een typisch formulier om deze gegevens in te vullen staat hier onder.



The image shows a login form with a light gray background and a thin border. It contains two input fields: one for the username labeled 'Gebruiker:' and one for the password labeled 'Password:'. Below the fields are two buttons: 'Login' and 'Overnieuw'.

De HTML-code voor dit formulier luidt:

```
<a name="LOGINFORM"></a><form action="JavaScript:ValidateLogin()">
<table border="1" cellspacing="2" cellpadding="5" align="center"
bgcolor="#C8C8C8" frame="box" rules="rows">
<tr>
<td align="right"><b>Gebruiker:</b> </td><td><input type="text"
id="Bezoeker" value=""; size="20"></td>
</tr><tr>
<td align="right"><b>Password:</b> </td><td><input type="password"
id="WachtWoord" size="20"></td>
</tr><tr>
<td align="center" colspan="2"><input type="submit" value="Login"
style="font-weight:bold">&nbsp;&nbsp;&nbsp;<input type="reset"
value="Overnieuw"></td>
</tr>
</table>
</form>
```

De tag `` wordt in hoofdstuk 5 besproken. Je bent verder vrij om de opmaak van dit formulier aan te passen aan je eigen smaak en wensen. De rood geprinte zaken moet je onveranderd laten.

De JavaScript code van de function `ValidateLogin()` staat in de file `.protectpages`. In de bladzijde waar je dit formulier gebruikt neem je de volgende code op tussen de `<HEAD>.....</HEAD>` tags:

```
<script language="JavaScript" src=".protectpages"></script>
```

BEN'S HOBBYHOEKJE

De functie `ValidateLogin()` brengt je bezoekers naar de root van het beveiligde deel van je website. De naam van die file geef je op in de variabele `accessGrantedPage` in de file **.protectpages**. In de bijgevoegde voorbeelden is [protect_1.htm](#) gebruikt.

3. Login geweigerd

Als het inloggen fout gaat, wordt er een pagina getoond waar dat wordt gemeld. Dit is een HTML-file die, naast een foutmelding, hetzelfde formulier toont als hiervoor. De naam van die file geef je op in de variabele `incorrectLoginPage`, in de files **.protectpages** en **.protectedpage**. Je moet dat dus twee keer doen.

In de bijgevoegde voorbeelden is [noaccess.htm](#) gebruikt. Deze file wordt ook gebruikt als de bezoeker een beveiligde bladzijde oproept zonder eerst ingelogd te zijn.

De code van [noaccess.htm](#) staat hieronder. De opmaak kun je naar eigen inzicht aanpassen. De rood gedrukte zaken moet je onveranderd laten.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">

<html>
<head>
<title>Ben's HobbyHoekje -- Toegang geweigerd</title>
<style type="text/css"><!--
BODY,TD { font-family : verdana, arial, helvetica, sans-serif;
font-size : 10pt; font-weight: normal; color: #000000; text-decoration:
none
}
.Kop1 { font-family: Verdana, Arial, Helvetica, sans-serif;
font-size: 14pt; font-weight: bold; color: #990000; text-decoration: none
}
--></style>
<script language="JavaScript" src=".protectpages"></script>
<script language="JavaScript" type="text/javascript"><!--
function ChooseText() {
    if (location.search.length == 0)
        document.getElementById('LoginError').style.display = "block";
    else
        document.getElementById('NoAccess').style.display = "block";
}
--></script>
</head>

<body bgcolor="#E4E9F6" link="#0000CC" vlink="#CC0000" alink="#009900"
onload="ChooseText()">
<div align="center"><p class="Kop1">Sorry...</p>
<p id="LoginError" style="display:none">De door u opgegeven Gebruikersnaam
en/of het Wachtwoord<br> zijn niet juist.<br>
&nbsp;<br>Probeer het opnieuw.</p>
<p id="NoAccess" style="display:none">U moet eerst inloggen om toegang te
krijgen<br>tot dit deel van de website.</p>
<p> formulier, zelfde code als in hoofdstuk 2 </p>

<p><a href="javascript:goRoot()"></a></p>
</div></body>
</html>
```

BEN'S HOBBYHOEKJE

4. Geldigheidsduur van een login-code.

Met de variabele `accessGrantedTime` bepaal je hoe lang een correcte login geldig blijft. Deze variabele staat in de file **.protectpages** en is ingesteld op 1 uur. Je kunt hem echter net zo lang maken als je wilt.

Als je bezoeker langer dan 1 uur blijft zal hij/zij na 1 uur dus opnieuw moeten inloggen. Als dit te kort is kun je hem natuurlijk altijd langer maken.

5. Een bladzijde beveiligen.

Je beveiligt een bladzijde door de volgende regel code op te nemen tussen de `<HEAD>.....</HEAD>` tags:

```
<script language="JavaScript" src=".protectedpage"></script>
```

Geef je bezoekers op elke beveiligde bladzijde de gelegenheid om uit te loggen. Dat doe je door de JavaScript function `LogOff()` aan te roepen:

```
<a href="JavaScript:LogOff()">Log Uit</a>
```

De bezoeker wordt nu teruggestuurd naar het eerste login-formulier (zie hoofdstuk2). De plaats daarvan geef je op in de variabele `protectionRootPage` in de files **.protectpages** en **.protectedpage**. Je moet dat dus twee keer doen. Laat de roodgedrukte zaken onveranderd!

In de bijgevoegde voorbeelden is [html-202.htm#LOGINFORM](#) gebruikt. Hiermee wordt binnen [html-202.htm](#) direct naar de bookmark [LOGINFORM](#) (het formulier) gesprongen. De code van [html-202.htm](#) is gebruikt op de website.

6. Login-codes maken.

Voor het maken van login-codes is het programma **createlogincodes.htm** beschikbaar, zie de figuur op de volgende bladzijde.

Vul de bovenste twee velden in. De versleutelde versies worden in de onderste twee velden getoond.

Het programma accepteert geen lege invoer. Gebruikersnamen zijn uniek. Een naam kan dus maar éénmaal worden ingevoerd. Passwords kunnen wel meer dan eenmaal voorkomen.

De versleutelde gebruikersnaam moet je opnemen in de array `protu`; het versleutelde password moet je opnemen in de array `protw`. Beide arrays staan in de file **.protectpages**.

Aangezien je zelf de toegangscode maakt, bepaal je zelf hoe lang de user-name en het password zijn. Om een redelijke beveiliging te bereiken moet de user-name minimaal 6 en het password minimaal 9 karakters lang zijn. Het programma controleert daar niet op. Het aantal karakters dat je gebruikt in de user-name en het password hoeft je niet apart op te geven. Dat bepaalt het programma zelf.

Je kunt net zoveel gebruikers toevoegen als je wilt. Zorg er voor dat de nummering bij nul begint en dat er geen gaten in de nummering voorkomen.

BEN'S HOBBYHOEKJE

De versie van **.protectpages** die in de download file staat bevat als voorbeeld één gebruiker: **Ali-Baba** met als password **Sesam_Open_U**. Aanbevolen wordt om deze te verwijderen als je dit op je eigen website toepast!

Login-code maken

Geef de naam van de gebruiker en het password op, klik daarna op **Versleutelen**
Neem het resultaat op in *.protectpages*.

Geldige karakters zijn: [A..Z], [a..z], [0..9], [@\$%&+ -_ =.*]

Invoer voor **Gebruiker**:

Invoer voor **Password**:

Resultaat voor **Gebruiker**:

Resultaat voor **Password**:

Merk op dat niet alle karakters van het toetsenbord kunnen worden gebruikt. De geldige karakters staan op dit formulier.

Als je meer of minder karakters wilt toestaan om de login-codes mee te maken, dan kan dat door de variabele `TheChars` in de function `Scramble()` te wijzigen. Deze function staat in de file **.protectpages**. De code van het formulier, waarmee de beschikbare karakters worden getoond, wordt niet automatisch aangepast. Dat moet je dus zelf doen! Als je hier aanpassingen doet, moet je daarna **alle** gebruikersnamen en passwords opnieuw aanmaken!

Het is **noodzakelijk** dat het aantal karakters in `TheChars` een **priemgetal** is. Anders is de versleuteling niet éénduidig (een versleutelde string kan gebaseerd zijn op verschillende invoer-strings).

7. Beperkingen van de beveiliging en het gebruik.

De login-codes moeten –door de webmaster- zelf worden ingebouwd in de programma-code.

Dit betekent dat een nieuwe gebruiker eerst toegang moet aanvragen bij de webmaster. Deze maakt de login-code, plaatst de versleutelde versie in de code en maakt deze beschikbaar. Het is dus voor bezoekers van de site niet mogelijk om zelf een account te maken en te activeren. Daar is deze aanpak overigens ook niet voor bedoeld.

De function die de gebruikersnaam en het password versleutelt staat in code die in de internet-caches van de browser terechtkomt, en is daardoor voor iedereen zichtbaar. Dat maakt het mogelijk om de versleutelde gebruikersnamen en wachtwoorden "terug te rekenen". In combinatie met wat creativiteit kunnen dan login-codes worden gevonden.

Bij het uploaden van de files **.protectpages** en **.protectedpage** naar de host kan het een en ander fout gaan, waardoor de beveiliging niet werkt. Dit komt doordat de filenamen beginnen met een punt. Daardoor hebben ze geen echte naam, maar alleen

BEN'S HOBBYHOEKJE

een extensie. Veel upload-programma's kunnen daar niet mee overweg, en maken een willekeurige naam aan (wel met de goede extensie), zoals abcdef.protectedpage.

Er zijn twee manieren om dit te ondervangen:

- Verander de namen van **.protectpages** en **.protectedpage** in **protectpages.txt** respectievelijk **protectedpage.txt**. Upload deze files naar de host en verander daar de file namen weer terug.
- Je kunt er ook voor kiezen de namen van **.protectpages** en **.protectedpage** te permanent te wijzigen, bijvoorbeeld in **x.protectpages** en **x.protectedpage**. Je moet dan wel alle verwijzingen naar deze files in je HTML-code aanpassen!

Als de beveiliging van de browser op een te hoog niveau staat ingesteld, kan het gebeuren dat JavaScript is uitgeschakeld. In dat geval kan een login-code niet uit het formulier worden gelezen en wordt er dus geen toegang verleend. Het beveiligde deel van je website is dan niet toegankelijk, tenzij de gebruiker de namen van de beveiligde HTML-bestanden weet. Hij/zij kan zichzelf dan toch toegang verschaffen.

8. Tenslotte...

Nogmaals: Deze beveiliging is niet waterdicht. Een slimme programmeur kraakt de login-code zonder al te veel problemen, omdat het algoritme voor de versleuteling bekend is.

Lees dus ook dit nog even:

Disclaimer

Gebruik van deze techniek is geheel voor eigen risico. De auteur van deze website is niet aansprakelijk voor enige schade die het gevolg is of het gevolg zou kunnen zijn van enige toepassing van deze techniek
Toepassing van deze techniek op je eigen website betekent automatisch dat je instemt met deze disclaimer en dat je de auteur van deze site vrijwaart van elke schadeclaim.