Passwortzugang mit Symbolbefehl für CMBasic

(Version 2 vom 30.03.2020)

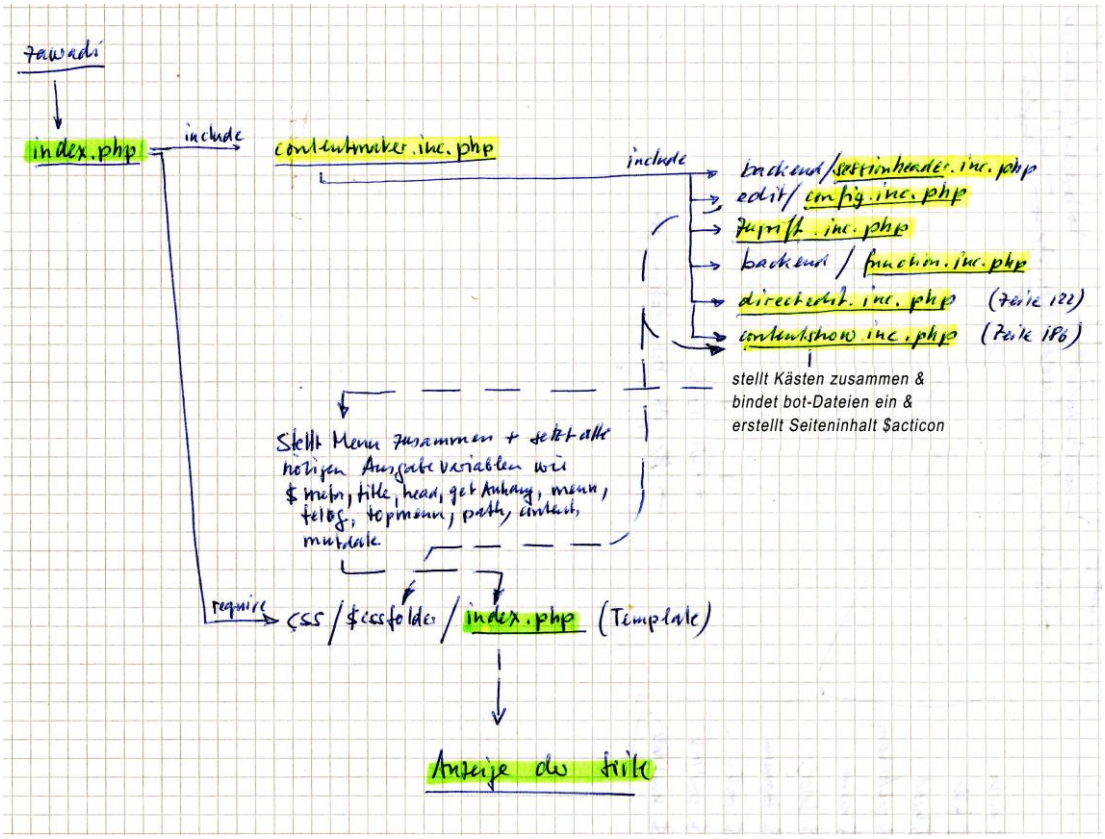
Vorstudien

Idee des Passwortzuganges:

- Mittels Symbolbefehl auf beliebiger Seite einsetzbar
- Symbolbefehl enthält Pseudo-Passwort und Ziel URLAnhang.
- Symbolbefehl blendet Eingabeformular für Passwort ein
- Das Formular sendet das eingegebene Passwort zusammen mit dem Pseudo-Passwort und dem Ziel-URLAnhang zur Auswertung
- Die Auswertung vergleicht das aus dem Pseudo-Passwort abgeleitete Referenz-Passwort mit dem eigegebenen Passwort und öffnet bei Übereinstimmung die Seite mit dem Ziel-URLAnhang (z.B. URL = famlist)
- Bei fehlerhafter Eingabe wird eine vorher auf der Website versteckt eingerichtete Abfangseite mit Rücksprungmöglichkeit geöffnet.
- Durch das Verwenden von Sessionen und Tarnkappenskripts kann eine vollständige Absicherung der geschützen Seiten erreicht werden.

Progammabläufe im CMBasic

Das index.php bindet die Datei **contentmaker.inc.php** ein, welche ihrerseits die Dateien backend/**sessionheader.inc.php**, edit/**config.inc.php**, **zugriff.inc.php**, backend/**function.inc.php**, **directedit.inc.php** (Zeile 122), **contentshow.inc.php** (Zeile 186) einbindet. Die Datei contentshow.inc.php bindet die config.inc.php freigegebenen bot_Dateien ein und erstellt den Seiteninhalt **$acticon** der gewünschten Seite, welcher via CSS/$cssfolder/index.php angezeigt wird.



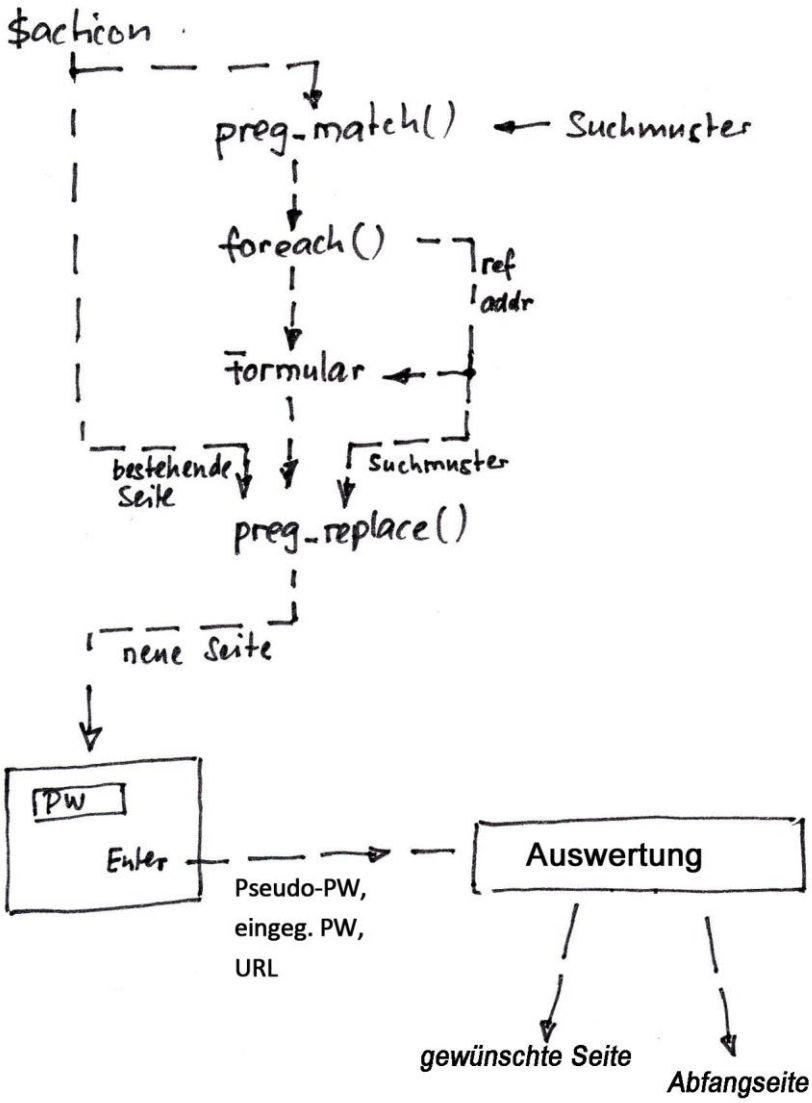
Aufbau und Name des Symbolbefehls:

Aufbau ähnlich wie andere Symbolbefehle (Beispiele: {sitemap} / {blog,nnn} / {video,dok.flv [breite][höhe]} / {audio,dok.mp3} / {galerie,galerieX} etc.):

{PWZUGANG,Pseudo-Passwort,URL-Anhang}

Einbindung in den Programmablauf:

- Aufbau und Benennung des verwendeten Symbolbefehls „PWZUGANG“
- Neue Zeile mit Freigabemöglichkeit des Symbolbefehls in der Datei **config.inc.php**
- Neue Zeile zur Einbindung des Symbolbefehls in der Datei **contentshow.inc.php**
- Einfügung einer neuen Datei **bot_pwzugang.inc.php** im Verzeichnis minibot, in welcher das Eingabeformular und die Auswertung integriert ist.



Erweiterung der Datei config.inc.php:

Neue Zeile mit Freigabemöglichkeit des Symbolbefehls einfügen:

```
define('PWZUGANG', TRUE); // Passwortzugang freigeben
```

Erweiterung der Datei contentshow.inc.php (Zeile 584 ff)

Neue Zeilen zur Einbindung des Symbolbefehls einfügen:

```
if (defined('PWZUGANG') && PWZUGANG == TRUE)
  include_once 'minibot/bot_pwzugang.inc.php';
}
```

Einfügen der Tarnkappendatei spezlist.inc.:

Der Tarnkappendatei tarnkappe.inc.php den Namen der zu schützenden Seite geben (z.B. ~~tarnkappe~~.inc.php -> famlist.inc.php) und die Datei unter skripts/content ablegen.

Wichtig: Alle hinter der Passwortseite liegenden Dateien müssen diese Tarnkappe besitzen!

Entwicklung des Moduls „bot_pwzugang.inc.php“:

Variablen:

- $acticon (Seitentext)
- $pw (eingegebenes Passwort)
- $ref (Pseudo- bzw. Referenz-Passwort)
- $adr (ULR-Anhang der Zielseite)

Programmablauf:

- An Stelle des Symbolbefehls erscheint das Eintrittsformular
- Das eingegebene Passwort wird zusammen mit dem ULR-Anhang und dem Pseudo-Passwort mittels POST zur Auswertung weitergeleitet.
- Das Pseudo-Passwort wird in das entsprechende Referenz-Passwort umgesetzt.
- Die Auswertung erkennt, filtert und vergleicht das eingegebene Passwort ($pw) mit dem Referenz-Passwort ($ref).
- Bei Übereinstimmung wird die Zielseite mit dem URL-Anhang ($adr) geöffnet und eine Session ZUGANG gestartet.
- Bei falschem Passwort wird eine vorgängig unter den versteckten Einträgen eingerichtete Abfangseite mit Kontakt- und Retourlink angeschaltet.

Script der Symbolbefehlsdatei inkl. Eingabeformular (bot_pwzugang.inc.php):

Kurzbeschreibung: Die Datei bot_pwzugang.inc.php erkennt den Symbolbefehl {PWZUGANG}, liest seine Attribute aus, erzeugt ein Eingabeformular und sendet es per POST an das in derselben Datei liegende Auswertescript, welches je nach Resultat die gewünschte Seite öffnet oder eine Abfangseite ansteuert.

```
<?php
/*
name:       bot_pwzugang.inc.php für CMBasic 1.6.7 und höher
author:     Walo Zach  zawadi@bluewin.ch   http://www.zawadi.ch
date:       edited 30.03.2020 by wz
description: erkennt Symbolbefehl, erzeugt Formular und sendet Eingabe
an eigene Datei zurück, wechselt Pseudoreferen gegen Referenzpasswort aus,
wertet eingegebenes Passwort aus und schaltet entsprechende Seite oder
Abfangseite ein. Eröffnet die SESSION[Zugang]*/
/* vorsorglicher Sessions-Start für evtl. installierte Tarnkappe */
ini_set("session.use_cookies", 1);
ini_set("session.use_only_cookies", 1);
ini_set("session.use_trans_sid", 0);
session_start();
defined('CMBASIC') or die();
/* Entscheidung ob Auswertung oder Eingabe */
if((isset($_POST['ref'])) && (isset($_POST['adr'])) && (isset($_POST['pw']))) {
/* Zweig für Validation der Eingaben */
/* empfangen und umsetzen der POST-Dateien */
$ref_m = $_POST['ref'];
$ref_t = htmlspecialchars($ref_m); // Verhindert Einschleusen von html-Tags
$ref_x = trim($ref_t);          // Eingabefilter
/*Pseudoreferenzen-Schluessel */
$ref_a = 'aaaaa';
$ref_b = 'bbbbb';
$ref_c = 'ccccc';
$ref_d = 'ddddd';
/* Umwandlung der Pseudoreferenz $ref_x in richtige Referenz */
 switch ($ref_x) {
 case ($ref_a):
 $ref_v = 'Alpha';
 break;
 case ($ref_b):
 $ref_v = 'Beta';
 break;
 case ($ref_c):
 $ref_v = 'Gamma';
 break;
 case ($ref_d):
```

```
 $ref_v = 'Delta';
 break;
 default:
 $ref_v = 'Beta';
 break;
 }
/* Anpassung der restlichen POST-Parameter */
$adr_m = $_POST['adr'];
$adr_t = htmlspecialchars($adr_m);  // Verhindert Einschleusen von html-Tags
$adr_v = trim($adr_t);            // Eingabefilter
$pw_m = $_POST['pw'];
$pw_t = htmlspecialchars($pw_m);    // Verhindert Einschleusen von html-Tags
$pw_v = trim($pw_t);              // Eingabefilter
/* Auswertung */
$dokAdresse =
array("./index.php?home,abfangmeldung","./index.php?home,$adr_v","./index.php?home,$adr_spe
z");
$URL=$dokAdresse[0];  // setzt vorsorglich Abfangseite
 switch ($pw_v) {
 case ($ref_v):
 $URL=$dokAdresse[1];  // setzt Zielseite
 $_SESSION['ZUGANG'] = 'erlaubt'; // setzt Session ZUGANG auf 'erlaubt' und schaltet Tarnkappe frei
 break;
 default:
 $URL=$dokAdresse[0];  // setzt Abfangseite
 break;
 }
 $URL = "Location: ". $URL;
 Header($URL);  // schaltet gewählte Seite ein
}
else{
/* Zweig für Formularerzeugung */
$muster_s = "|{pwzugang,referenz,adresse}|";  // Syntax-Muster
$muster0 = "|{((PWZUGANG),(.*?))}|si";       // Suchmuster

/* Musterdefinitionen: i = Gleichschaltung Klein- & Grossschreibung
/ . = Jockerpunkt für Zeichen / s = Erweiterung Jockerpunkt auf alle Zeichentypen / * = alle Zeichen / ?
= nur letzter Durchgang */
/* Erkennen Systembefehl und Zerlegen in Parameter */
$preg = preg_match_all($muster0,$acticon,$pwfund);      // acticon = Zeichenstring / pwfund = Array
foreach($pwfund[1] as $pwpara_komp) {   // Auslesen der Parameterzeile
$pwpara = explode("," , $pwpara_komp);  // Parameterzeile wird durch "," in Parameter aufgeteilt
 if (!empty($pwpara[1])) {        // erster Parameter (Pseudoreferenz)
   $ref_0 = $pwpara[1];
 }
 if (!empty($pwpara[2])) {        // zweiter Parameter (ULR der Zielseite)
```

```
   $adr_0 = $pwpara[2];
  }
  }
/* Zusammenstellen des Formulars bzw. des Symbolbefehl-Ersatzes */
$pwcode_0 = <<<PWCODE
<p>Dies ist ein geschuetzter Bereich. Bitte zuerst das notwendige Passwort eingeben!</p><br/>
<form action='' method='post'>
<input type='hidden' name='ref' value='{$ref_0}' maxlength='20'>
<input type='hidden' name='adr' value='{$adr_0}' maxlength='20'>
<table width='287px' border='0'>
<tbody>
<tr align='left' valign='middle'>
<td width='20px' align='left'></td>
<td width='150px'></td>
<td><input name='pw' type='password' /></td>
</tr>
<tr height='10px'>
<td></td>
</tr>
<tr>
<td width='20px'></td>
<td width='150px'></td>
<td><input type='submit' value='Eintreten' /></td>
</tr>
</tbody>
</table>
</form>
<p><br/>Das Passwort kann beim Webmaster bezogen werden!</p>
PWCODE;
$pwcode = "$pwcode_0";      // mit Stringzeichen versehen
$muster2 = "|{PWZUGANG,$ref_0,$adr_0}|";
/* Ersetzen des Symbolbefehls */
$acticon = preg_replace($muster2, $pwcode, $acticon);
}
?>
```

Tarnkappen-Script (tarnkappe.inc.php) zur Erhöhung der Sicherheit:

Kurzbeschreibung: prüft mittels SESSION ob die zur Veröffentlichung nötige Bedingung (erfolgreicher PW-Zugang -> Session ‘ZUGANG‘ aktiv) erfüllt ist und blendet die geschützte Seite bei Nichterfüllen der Bedingung aus.

```
<?php
session_start();   // schaltet Session ein
if (isset($_SESSION['ZUGANG']) || isset($_SESSION [‘CMBASICLOG‘]))  {    // prüft ob PW-Zugang aktiv ist
   $sessinhalt = $_SESSION['ZUGANG'];              // Dummy-Befehl: liest Inhalt der Session Zugang erneut aus
   } else {
```

```
        die ('Zugang nicht freigegeben!');          // schaltet bei unerlaubtem Zugang Seitenansicht aus
      }
?>
```

Dieses Datei wird unter skripts/content abgelegt. Sie muss zum Aktivieren nur noch in den Dateinamen der zu schützenden Seite (z.B. ~~tarnkappe~~.inc.php -> spezlist.inc.php) umbenannt werden. Skripts werden im Programmablauf immer zu Beginn einer Seite eingefügt.

Beschreibung fürs Handbuch

Einbinden eines einfachen Passwortzuganges

Wenn erwünscht, lässt sich pro Seite ein ***einfacher Passwortzugang*** mit einem für alle Benutzer gemeinsamen Passwort einbinden, der es erlaubt mit einem bis zu 20 Zeichen grossen Passwort eine versteckte Seite (negativer Ebene2 - Wert) zu öffnen. Der dazu verwendete Symbolbefehl lautet:

{PWZUGANG,Pseudo-Passwort,URLAnhang}

Das erste fixe Attribut (PWZUGANG) zeigt dem System an, dass am Platz des Symbolbefehls ein Formular zur Passworteingabe erstellt werden soll. Mit dem zweiten Attribut (Pseudo-Passwort) wird ein Pseudo-Passwort und mit dem dritten Attribut (URLAnhang) wird der URLAnhang der Zielseite eingegeben. Das Pseudo-Passwort dient zur Auswahl eines der vier voreingestellten Passwörter, die vom Webdesigner vorher eingebracht sein müssen.

Zur Passwortvergabe sind die Zeichen 1 – 0 A – Z a – z ä, ö, ü ; / : -_ ! zugelassen und es wird Gross- und Kleinschreibung unterschieden. Es dürfen keine Zeichen ,? ( ) { } [ ] + * \ <> verwendet werden. Das Passwort darf zudem nicht länger als 20 Zeichen sein und nicht mit einem Leerzeichen beginnen oder enden; einzelne Leerzeichen innerhalb des Passwortes sind jedoch zugelassen.

Der einzusetzende URLAnhang der Zielseite findet man in der Content-Tabelle im Backend.

Sind Pseudo-Passwort und Ziel-URLAnhang normgerecht eingefügt, erscheint am Platz des Symbolbefehls das erwünschte Eingabeformular. Bei falschem Passwort wird eine vorgängig unter den versteckten Einträgen erstellte Seite mit dem Namen „abfangmeldung“ angesteuert.

Dieser Zugang ***bietet keinen perfekten Schutz***, da er ohne Benutzerkennung arbeitet und für alle Benutzer das gleiche Passwort verwendet. Er ist einzig zum Schutz vor Suchmaschinen und Neugierigen geeignet.

Damit dieser Passwortzugang funktioniert, muss er in der Datei config.inc.php (im Verzeichnis «edit») freigeschaltet werden! Zudem muss die Datei contentshow.inc.php im Root entsprechend angepasst sein und müssen die versteckte Zielseite (URLAnhang) und die Seite mit der Abfangmeldung (abfang-meldung) in die Website eingefügt sein.

Um die beim gegenwärtigen CMBasic den nicht ganz hundertprozentigen Zugriffschutz zu erhöhen ist es empfehlenswert jeden versteckten Eintrag mit dem vorgeschlagenen optionalen Tarnkappen-Script zu versehen. Nur so kann verhindert werden, dass die versteckten Einträge nicht direkt aus dem Internet eingesehen bzw. heruntergeladen werden können.

Test der entwickelten Funktion

Funktionstest in Arbeitsumgebung

Ein Funktionstest zeigte, dass der Symbolbefehl ‚PWZUGANG' funktioniert. Beim Test wurden folgende Eigenheiten festgestellt:

- Pro Seite ist nur ein einziger PW-Zugang möglich
- Die Positionierung des Systembefehls ist irgendwo auf jeder Seite möglich
- Gross- und Kleinschreibung wird unterschieden
- Umlaute werden akzeptiert
- Leerzeichen werden akzeptiert, solange nicht mehrere nacheinander folgen bzw. nicht zu Beginn und am Ende des Strings
- Es dürfen keine Sonder- und Satzzeichen wie (),:;._-?!"+&* im Passwort vorhanden sein
- Wenn die Begrenzung von 20 Zeichen überschritten wird, werden überzählige Zeichen nicht mehr angeschaut. Dies gilt bei PW-Eingabe und beim Pseudo-Passwort!

Überprüfung der Sicherheit

Einrichten des Symbolbefehls:

- Ein nicht normgerecht eingerichtetes Passwort wird durch Nichtfunktionieren des Symbolbefehls quittiert (d.h. Formular wird nicht eingeblendet).
  Akzeptierte Zeichen ;/:,-_! Nicht akzeptierte Zeichen ,? ( ) { } [ ] + * \<>
  Bei Verwenden des Zeichens < innerhalb des PW-Strings wird der Symbolbefehl bei erneutem Öffnen nicht mehr vollständig angezeigt! (Grund: html-Entitäten werden in DB nicht abgelegt)
- Verwenden einer nicht normgerechten Ziel-Adresse (z.B. **?**URLAnhang) wird ebenfalls durch Nichtfunktionieren des Symbolbefehls quittiert.

Eingeben des Passwortes:

- Falsches bzw. leeres bzw. mit Leerzeichen gefülltes Passwort führt zu einer Fehlermeldung
- folgende Zeichen führen zu einer Fehlermeldung: ,? ( ) { } [ ] + * \ <>
- folgende Zeichen werden akzeptiert: ;/:-_!
- Überlange Passwörter werden nach 20 Zeichen abgeschnitten

Offene Probleme:

Die Ansteuerung von passwortgeschützten, versteckten Einträgen aus dem Netz ist mit voran-gestelltem Zusatz „?home" möglich. Dies ist ein Problem von CMBasic, lässt sich aber mit dem Tarnkappenschutz eliminieren. Dieser Schutz (unter skript/content) verhindert, dass versteckte Einträge aus dem Netz gelesen werden können. Es wird „Zugang nicht freigegeben" angezeigt.