

Dateiaustausch

Letzte Aktualisierung Montag, 12. Juli 2010

Sie möchten ab und zu eine Datei (Text, Audio, Video, ...) jemandem anderes zur Verfügung stellen, haben aber keinen eigenen FTP- oder Webserver, wo Sie die Datei ablegen könnten? Sie können die Datei nicht in eine EMail packen, weil sie zu groß würde? Sie haben aber auch kein Dokumenten-Management-System zur Hand, mit dem Sie einen solchen Dateitransfer veranstalten können?

Die Lösung

Das hier beschriebene Programm, exchange.pl, füllt diese Lücke. Es wird auf einem Webserver installiert und fortan kann es von jedem Internet-fähigen Arbeitsplatz aus mit einem normalen Webbrowser bedient werden. Durch einen simplen html-Fileupload kann jeder Dateien in das sogenannte Dateilager hochladen. Man bekommt eine URL zurück, unter der das Dokument fortan für eine beschränkte Zeit verfügbar ist. Diese URL kann nun per EMail verschickt werden und erlaubt den direkten Zugriff auf die Datei ohne Passwort oder weitere Kenntnisse. Die auf dem Server gespeicherten Dateien sind exklusiv über diese URL zu erreichen und so vor ungewolltem Zugriff geschützt.

B sagt zu A "Schick mir mal die große Datei" geht nun so:

- A ruft dieses Programm mit seinem Webbrowser auf
- A wählt die hochzuladende Datei aus
- A schreibt die E-Mailadresse von B und eine kurze Notiz in die entsprechenden Felder und klickt auf "Hochladen"
- Fertig: B bekommt die E-Mail vom System mit der URL zum Herunterladen der Datei (mit der Notiz von A).

Dieses Programm löscht die abgelegten Dateien nach einer gewissen Zeit (abhängig von der Konfiguration), der Betrieb ist wartungsfrei.

Der Zugang wird durch IP-Bereiche oder Authentifizierung geregelt. Für Betriebsfremde Nutzer gibt es ein TAN-Verfahren: Interne Nutzer können TAN erzeugen, die von externen für einen oder mehrere Uploads verwendet werden können. TANs können mit einer Emailadresse versehen werden, in diesem Fall sorgt das System für eine Benachrichtigung per E-Mail an den TAN-Erzeuger, wenn die TAN verwendet wurde. Der Vorgang schick mir mal die große Datei ist also auch bei Betriebsfremden sehr einfach:

- A erzeugt eine TAN, die mit seiner eigenen E-Mailadresse verknüpft ist
- A gibt die TAN und die Webadresse dieses Programmes an B
- B lädt die Datei hoch, authentifiziert sich mit der TAN
- Das System erzeugt automatisch eine E-Mail an A mit der URL der hochgeladenen Datei.
- Per Klick in die E-Mail wird die Datei von A heruntergeladen.

Das TAN-Management ist ebenfalls wartungsfrei, abgelaufene oder endgültig verwendete TAN werden automatisch ungültig.

Das System enthält ein Administrationsinterface, auf dem mit einem Blick festgestellt werden kann, ob es dem Speicher noch gut geht,

welche Dateien welcher Größe im Lager liegen. Dateien, die zweifelhaften Inhalt haben, können vom Administrator mit einem Mausklick gelöscht werden. Alle Transaktionsnummern mit Ablaufdatum, Erzeugungsdatum und verknüpfter Emailadresse sind auch eine gewisse Zeit, nachdem sie ungültig geworden sind, für den Admin einsehbar (Supporterleichterung).

Auf der Serverplatte werden die Dateien nicht mit dem vom Client übermittelten Namen gespeichert, um Mißbrauch auf diese Weise auszuschliessen. Stattdessen wird der Name in einer Datenbank gespeichert und die Datei mit einer selbst erzeugten Zahlen-Ziffern-Kombination im Namen abgelegt.

Bei jedem Aufruf des Programmes - egal, ob vom Admin oder einem normalen Client - werden Dateien und TANs auf Alter und Gültigkeit gecheckt. Vor jeder Aktion werden dann die Daten, die ihre maximale Lebenszeit erreicht haben, gelöscht. So ist es möglich, daß der gesamte Dienst mit einem einzigen Perlprogramm plus 1-3 Konfigdateien funktioniert. Kein cron-job, kein Programm, welches in regelmäßigen Abständen aufzurufen wäre. Nur ein einziges Perlprogramm.
Screenshots

Details

Dieses Programm erfüllt die folgenden Anforderungen:

- Einfachst zu bedienen
- Client-Systemunabhängig
- Einfache Authentifizierung beim Upload per Einmal-TAN, Passwort oder Radius-Account
- Automatische Überwachung des Dateilagers auf Größe etc., dadurch absolute Wartungsfreiheit

- In andere Sprachen übersetzbar

Anforderungen

Das Programm wurde in Perl erstellt und sollte insofern auf einem Server lauffähig sein, auf dem ein Perl funktioniert. Entwickelt und getestet wurde erst auf Suse-Linux, dann auf Ubuntu. Desweiteren sind einige Perlmodule notwendig, die sehr einfach per CPAN-Modul (perl -MCPAN -e shell) nachinstalliert werden können, wenn sie fehlen. Es sind dies die Module

- CGI für die Kommunikation über die CGI-Schnittstelle.
- File::Basename um aus den übergebenen Dateinamen die Pfade zu entfernen.
- DB_File zur Speicherung der Daten in Hashes auf der Serverplatte.
- Time::Local für Zeitberechnungen
- Date::Calc ebenso.
- Sys::Syslog um Meldungen in das Systemlog schreiben zu können.
- Email::Valid testet die eingegebenen EMailadressen bei TAN-Erzeugung auf Gültigkeit.
- Mail::Mailer schickt die Benachrichtigungs-EMails.
- HTML::Template für die ausgelagerten HTM- und E-Mail-Texte

Darüber hinaus wird ein Webserver (möglichst Apache) sowie Speicherplatz auf der Serverplatte benötigt. Wenn Radiusauthentifizierung verwendet wird, muß der FreeRadius-Client installiert werden.

Vorausgesetzt wird ein lauffähiges System mit Perl und Webserver. Das System muß via http und ggf https erreichbar sein und in der Lage sein, E-Mails zu verschicken (E-Mailempfang nicht notwendig!).

- Installation der Perlmodule aus obiger Liste
- Kopieren des Programmes in das cgi-bin Verzeichnis des Webservers
- Optional: Anpassen der Webserverkonfiguration, so daß das Script unter einer einfach zu merkenden URL erreichbar ist. Dazu wird in die Konfigurationsdatei des Apache die Zeile

Redirect /exchange /cgi-bin/exchange.pl

eingefügt und der Apache neu gestartet. Das Programm ist nun unter der URL <http://mein.web.server/exchange> erreichbar
- aber
jetzt noch nicht aufrufen!.

- Erstellen eines Verzeichnisses auf dem Webserver, welches nicht unterhalb des Documentroot des Webserverprozesses liegen darf. Nun wird der Eigentümer und die Gruppe dieses Verzeichnisses auf den User gesetzt, unter dem der Webserverprozess läuft. Unter SuSe wäre das

chown wwwrun:nogroup /pfad/zum/verzeichnis

Dieser User muß hier Schreib- und Leserechte haben.

- Entpacken der Sprachdateien in ein eigenes Verzeichnis, welches nicht unterhalb des Documentroot des Webservers liegt. Es schadet nichts, die Sprachdateien als Unterverzeichnis zum oben genannten /pfad/zum/verzeichnis zu erstellen. Dort liegen dann also /pfad/zum/verzeichnis/lang/de/*.tmpl . Möchte man weitere Sprachen realisieren, erstellt man ein Verzeichnis parallel zu de mit dem entsprechenden Sprachkürzel (bspw /pfad/zum/verzeichnis/lang/en), kopiert alle Dateien aus de dorthinein und übersetzt diese. In den Dateien steht oben jeweils ein Kommentar, der die Bedeutung der Datei erklärt. Dieser Kommentar kann auch entfernt werden. Wichtig ist, dass die Variablen, die zur Laufzeit ersetzt werden, nicht inhaltlich geändert werden. Es sind dies die Blöcke wie z.B. <TMPL_VAR Name=BLAHBLAH>

Die neue Sprache ist dann, wenn das Programm fertig installiert ist, per an die URL angehängtem ?lang=en zu erreichen, also bspw <http://meine.domain.tld/exchange?lang=en>

- Bei der Benutzung der Radius-Authentifizierung: Installation von FreeRadius. Es wird nur der Client benötigt, die reine Installation genügt also, keine Free-Radius-Configdatei ist notwendig, kein Dienst muß gestartet werden.

- Kopieren der Configdatei in dieses Verzeichnis (oder an eine andere Stelle, die der Webserver lesend erreichen kann).

- Anpassen der Configdatei. Achtung: Es ist wichtig, einmal alle Optionen in der Datei durchzugehen und sich zu überlegen, ob man mit diesen Werten glücklich ist. In jedem Fall müssen angepasst werden: \$filelocation, alles, was "meine.domain.tld" enthält, die Radiuseinstellungen oder \$useradius=0 und das Adminpasswort. Wenn man mit den voreingestellten Dateigrößen nicht zufrieden ist oder weniger als die voreingestellten GB Speicherplatz hat, muß das natürlich auch geändert werden.

- Anlegen der Datei, die in \$passwordfile konfiguriert ist. Hier steht das immer gültige Uploadpasswort. Bei der Verwendung von Radiusauthentifizierung und TANs braucht dieses Passwort nicht veröffentlicht zu werden.

- Bei Verwendung von TANs: Dasselbe für \$tancreatepasswordfile.

- Anpassen der Zeile

```
our $configfile = '/hd2/upload/exchange.conf.pl';
```

im Script. Hier muß der absolute Pfad zur Konfigurationsdatei eingetragen werden.

Unterhalb dieser Zeile im Script werden alle weiteren konfigurierbaren Optionen gelistet. Eine Änderung ist hier zwar auch möglich, allerdings fehlt hier die Erklärung der Optionen und die hier gelisteten Defaultwerte werden von denen in einer Konfigurationsdatei überlagert. Es ist also empfehlenswert, nur die Optionen in der Konfigurationsdatei zu bearbeiten und im Script lediglich den Pfad zur Konfigurationsdatei einzustellen. Außerdem ist dann ein zukünftiges Update einfacher.

- Im Webbrowser die URL öffnen, die man in der REDIRECT-Anweisung konfiguriert hat. Im obigen Beispiel ist dies

<http://mein.server.tld/exchange>

Jetzt

sollte die Uploadmaske erscheinen. Wenn nicht oder wenn sonst etwas merkwürdig erscheint, lohnt ein Blick in die Syslogdatei (meistens /var/log/messages) und das Errorlog des Apache (z.B. /var/log/httpd/error_log oder /var/log/apache2/error_log).

- Die Adminschnittstelle ist erreichbar unter

http://mein.server.tld/cgi-bin/exchange.pl?admin=password_welches_in_der_configdatei_gesetzt_wird

oder kürzer bei Verwendung der REDIRECT-Anweisung:

http://mein.server.tld/exchange?admin=password_welches_in_der_configdatei_gesetzt_wird

- Keine Fehlermeldung? Fertig.
- Geht nicht, aber keine Ahnung, warum? Schon lange gesucht, aber keine Ursache gefunden? Mail an mich genügt, bitte dann die eigene Konfigurationsdatei anhängen.

Scriptgesteuertes Hochladen

Das Dateiaustauschprogramm nimmt Dateien über ein http-upload entgegen. Natürlich kann man diese Schnittstelle mit geeigneten Mitteln auch per Script ansteuern. Um dies zu zeigen, verwende ich im Folgenden das Programm curl, für das es u.a. auch eine php-Library gibt. curl kann auf allen mir bekannten Linuxsystemen einfach aus dem repository nachinstalliert werden, es gibt u.a. auch eine Windowsvariante.

Hochladen per Script im Beispiel:

```
#!/bin/bash
```

```
#
```

```
NACHRICHT=$(cat <<EOT
```

```
Hallo,
```

```
dies ist die Datei, von der wir gestern gesprochen haben.
```

```
Gruesse,
```

```
Michael
```

```
EOT)
```

```
curl \
```

```
--form-string notifyemailto="ich@meine.domain.tld" \
```

```
--form-string notifyemailltext="$NACHRICHT" \
```

```
-F putfile=@datei.png \
```

```
http://meine.domain.tld/cgi-bin/exchange.pl
```

Dies lädt die lokal vorhandene datei.png auf den Dateiaustauschdienst hoch und versendet die Download-URL an die angegebene E-Mail-Adresse, inklusive des angegebenen Textes.

Dies kann man etwas allgemeingültiger formulieren:

```
#!/bin/bash

#

URL="http://meine.domain.tld/cgi-bin/exchange.pl"

echo -----

echo Upload a file

echo $0 email notificationfile uploadfile

echo -----

curl \

  --form-string notifyemailto="$1"          \

  --form-string notifyemailtext="`cat $2`"   \

  -F putfile="@$3"                          \

  $URL
```

Diesem Script kann man die Datei, die E-Mail-Adresse und eine Datei mit einer Textnachricht übergeben. Die Datei wird hochgeladen und die E-Mail wird entsprechend verschickt. Da dies nur zeigen soll, wie es funktioniert, sind keinerlei Parameterprüfungen enthalten. Diese Scripte setzen auch voraus, dass sich der sendende PC im freigeschalteten IP-Bereich befindet, also keine Authentifizierung notwendig ist. Eine Übergabe von Nutzernamen und Passwort oder einer passenden TAN ist mit der Angabe der passenden Parameter (siehe Quelltext der Hochlademaske) aber auch einfach möglich.

Übrigens kann curl auch eine Fortschrittsanzeige beim Hochladen anzeigen (Manpage von curl, Stichwort Progress meter). Merkwürdigerweise muss man dazu die Ausgabe von curl in eine Datei umleiten. Die Angabe von `-#` als Parameter zeigt einen "Balken", einfache Umlenkung der Ausgabe ohne diesen Parameter zeigt aktuelle Statistiken zum Upload - inkl. Transferrate und voraussichtlicher Restzeit. Das obige Beispiel müsste also für Statistikanzeige lauten:

```
#!/bin/bash

# Hurra, eine Fortschrittsanzeige!

URL="http://meine.domain.tld/cgi-bin/exchange.pl"
```

```
TEMP=`tempfile`  
echo -----  
echo Upload a file  
echo $0 email notificationfile uploadfile  
echo -----  
  
curl \  
  --form-string notifyemailto="$1"          \  
  --form-string notifyemailtext="`cat $2`"  \  
  -F putfile="@$3"                          \  
  $URL                                       \  
  > $TEMP  
rm $TEMP
```

FAQ

- Dateien werden nach der angegebenen Anzahl maximaler Downloads nicht gelöscht?

Siehe unten bei Historie, "Problembehebung, Tipps (7/10)"

- Der Transfer großer Dateien bricht ab bzw wird vom Server gestoppt?

Apache hat Sitzungstimeouts. Diese sollte man hochsetzen, wenn über langsame Leitungen große Dateien geladen werden sollen. In den Konfigurationsdateien von Apache sucht und erhöht man dazu die Werte Timeout sowie ggf SSLSessionCacheTimeout.

Im Übrigen haben manche Apacheversionen Probleme mit Dateien > 2 GB.

- Es erscheint beim Hochladeversuch die Fehlermeldung, dass das Passwort falsch sei, obwohl sich der Client in einem erlaubten IP-Bereich befindet.

Bei Aufruf der Hochlademaske von einer als "darf-hochladen"-konfigurierten IP wird das Hochladepasswort vom Server beim Ausliefern der Maske in einem Feld eingetragen. Mancher Benutzer kommt auf die Idee, dieses Passwort im

Browser zu speichern, wenn ihn der danach fragt. Sollte nun auf dem Server das Passwort geändert werden, füllt mancher Browser das gespeicherte statt des vom Server eingetragenen ein. Dies führt dann zu der Fehlermeldung. Lösung: Im Browser das gespeicherte Passwort für diese Seite löschen.

Inzwischen ist das Programm geändert, so dass dieses Problem nicht mehr auftritt.

Historie

- Problembhebung, Tipps (7/10)

- Es kann sein, dass nach starkem Betrieb und ggf einem Serverabsturz während eines Schreibvorganges eine der Dateien defekt ist, in den das Programm die Namen der Dateien, Downloadstatistiken etc speichert. Einmal ist dies bei mir vorgekommen, gemerkt habe ich es daran, dass Dateien, die nach x Downloads gelöscht werden sollte, weiterhin erreichbar waren.

Das Problem kann mit etwas Handarbeit dadurch gelöst werden, dass die Hashdateien mit den enthaltenen Daten neu erzeugt werden. Dazu habe ich das kleine Programm hashcopy.pl erstellt, welches vor Benutzung allerdings noch editiert werden muss! Anleitung im Programm.

- Das Programm hashcopy.pl

- Dasselbe, aber im Browser lesbar

- 3.11, 1/10:

- Ein Fehler bei der Authentifizierung der TAN-Erzeugung wurde berichtigt: Die Passwordeingabe - auch bei richtigem Passwort - führte zu keinem Ergebnis. Dies fiel nur auf, wenn von einem Rechner, der nicht per IP freigeschaltet wurde, auf die TAN-Erzeugung zugegriffen wurde und nicht die radius-Authentifizierung verwendet wurde.

- Die IP-Freischaltung war bei der TAN-Erzeugung nicht manipulationssicher. Wahrscheinlich gab es diesen Fehler seit Version 3.10. Nutzer, die Version 3.10 im Einsatz haben und die TAN-Erzeugung in Verbindung mit der IP-Freischaltung haben, sollten updaten. Die Voraussetzung für eine Manipulation ist allerdings recht hoch, mehr auf persönliche Nachfrage.

- 3.10, 12/09:

- Die überflüssige Anzeige des Passwortes (vorbesetzt) beim Hochladen und der TAN-Erzeugung von einem freigeschalteten Rechner aus wurde herausgenommen.

- Beim Hochladen kann man nun eine E-Mailadresse (und eine Notiz) angeben. Sofern das geschieht, wird das Programm nach erfolgtem Hochladen der Datei eine E-Mail mit der URL der Datei an die angegebene Adresse schicken. Die ggf. eingegebene Notiz wird angehängt.

- Das Versenden von E-Mails beim Upload (E-Mail aus einer TAN-Erzeugung oder weil der Hochladende eine Adresse angegeben hat) wird nun in der Logdatei vermerkt.

- Update:

- Eine neue Configvariable einfügen (\$allownotifyemail)

- Die geänderten und neuen Sprachdateien updaten (

main_gotpw_notan.tpl
main_gotpw_tan.tpl
main_gotpwtext_notan.tpl
main_gotpwtext_tan.tpl
tancreate_pwdtext.tpl
tancreate_pwd.tpl
upload_wrongemail.tpl
email_upload.tpl
starthtml.tpl
upsuccess.tpl
upsuccess_emailsent.tpl
main.tpl
main_notifyemail.tpl
)

- Das Programm selbst ersetzen.

- Achtung beim Update: Zwischen Punkt 2. und 3. gibt es beim Aufruf des Programmes nur eine Fehlermeldung. Die Sprachdateien sollten also in einem getrennten Verzeichnis vorbereitet werden!

- 3.02, 9/08: Eine XSS-Sicherheitslücke wurde geschlossen. Alle Versionen vor 3.02 sind verwundbar. Ein Angreifer, der per Passwort, TAN oder IP-Erlaubnis in der Lage ist, Dateien über dieses Programm hochzuladen, könnte in den Besitz des Admin-Passwortes kommen und so alle TANs und alle hochgeladene Dateien einsehen und löschen (nicht aber manipulieren). Der erfolgreiche Angriff erfordert noch das Aufrufen des Adminbereiches durch einen Admin inklusive des dortigen Ansehens der Dateilisten, damit das Passwort in den Besitz des Angreifers kommt.

Bitte installieren Sie die aktuelle Version und ändern Sie anschließend das Adminpasswort.

Danke an Robin Fink für das Herausfinden und Melden des Problemes!

- 3.01, 7/08: Ein Bug verhinderte die Passwordeingabe auf der TAN-Erzeugungsseite von einem PC aus mit einer IP, die nicht als intern konfiguriert war. Gefixt.

- 3.0, 4/08:

- Mehrsprachigkeit möglich (bisher nur deutsch inklusive, Übersetzungen anderer stelle ich hier gerne zur Verfügung). Ausgenommen ist das Admin-Interface, das ist immer deutsch, Übersetzung nur im Sourcecode möglich.

- Bruteforce-Angriffe auf Datei-IDs werden erkannt und gebremst

- TAN können vom Admin deaktiviert und wieder aktiviert werden.

- 2.9, 2/07: Beim Herunterladen einer Datei mit (im Server) unbekanntem Mimetype wurde kein Mimetype mitgeliefert. Das ist jetzt geändert, es wird application/octet-stream zurückgeliefert.

- 2.8, 12/06: Kleinere kosmetische Änderungen und Einbindung von MIME::Types, um dem Browser den korrekten Dateityp zu übermitteln. Mit der alten Version kann man mit Firefox 2.0 sonst die Datei nur speichern, nicht aber direkt mit der zugehörigen Applikation starten.

- 2.7, 2/06: Die Authentifizierung mit dem Radius-Modul arbeitete nicht mit dem neuen Radiusserver der UniDo zusammen. Deswegen habe ich das Modul gestrichen und durch einen Kommandozeilenaufruf des freeradius-Client ersetzt. Natuerlich kann man hier auch jede andere Art der Authentifizierung einbauen.

Ausserdem diverse kleine Änderungen.

- 2.3, 10/05: TAN können nun entweder nach erster Benutzung oder nach Ablauf der Gültigkeit ungültig werden. Man kann also TAN erzeugen, die innerhalb eines Zeitraumes für beliebig viele Hochladevorgänge gültig sind. Außerdem leichte Änderungen an der Textausgabe, englische Fachbegriffe wurden durch deutsche Worte ersetzt. Behebung kleinerer Bugs, so wird z.B. nun eine Fehlermeldung gezeigt, wenn versucht wird, Dateien mit 0 Byte hochzuladen.

- 2.1, 01/05: TAN-Löschung nach Ablauf der Lebenszeit wird jetzt geloggt.

- 2.0, 01/05: TAN-Erzeugung und -Verarbeitung eingeführt. Passwort/TAN jetzt case-INsensitive.

Emailbenachrichtigung eingeführt.

Radiusauth. eingeführt. Diverse kosmetische Änderungen.

- 1.24, 01/05: Bugfix: Im Januar eines Jahres gab's ne Fehlermeldung, weil die Datumsberechnung mit 0 nicht arbeiten kann.

- 1.23, 11/04: Kleiner Bugfix: Falsche form-URL beim Downloadfeld

- 1.22, 10/04: Kleiner Bugfix: Bei zu großen Dateien wurde "Bytes" statt "MB" in der Fehlermeldung angezeigt.

- 1.21, 10/04: uploadurl eingeführt, kann verschieden von ownurl sein

- 1.2, 10/04: iprange eingeführt: In vorkonfigurierten Adressbereichen ist Passwort vorausgefüllt

- 1.11, 10/04: Bug in der Sortierung der Dateien im Store in der Adminübersicht behoben

- 1.1, 10/04: Configdatei auslagerbar und kleinere Änderungen wg. Sicherheit

- 1.0, 10/04: Erste Version

Wunschliste

Ein dringender Wunsch ist die Realisierung einer Progressbar, also einer möglichst grafischen Anzeige des Fortschritts des Hochladens. Hierzu empfehlen sich einige Tools, die allerdings alle irgendeinen Nachteil haben. Das Thema ist aber noch nicht vom Tisch, siehe dazu aber auch den Abschnitt "Scriptgesteuertes Hochladen" weiter oben auf dieser Seite.

Ein weiterer Wunsch wäre die Möglichkeit, mehrere Dateien in einem Rutsch hochladen zu können. Die bekämen dann eine ID, die beim Aufruf auf eine Seite führt, von der man alle diese Dateien herunter laden kann. Voraussetzung dafür ist ein Tool, welches mit Ajax/Javascript o.ä. arbeitet und dann idealerweise auch gleich die oben erwähnte Progressbar realisiert. Gibt es alles - die Integration in dieses Programm ist allerdings alles andere als einfach.

Wenn ich mal Zeit habe...

Schön wäre auch die Wiederaufnahme einmal abgebrochener Uploads. Da allerdings das Programm keine Sessionverwaltung oder gespeicherte Useraccounts hat und jeder misslungener Upload weggeworfen wird, passt dieser Wunsch nicht ins Konzept.

Auf der Liste steht ebenfalls eine ldap-Anbindung (siehe auch hier), die ich aber mangels ldap-Server zur Zeit nicht realisieren kann.

Download

Dieses Programm ist kostenfrei im nicht-kommerziellen Umfeld verwendbar. Das Programm darf ohne meine Einwilligung nicht kommerziell verwendet werden. Die Stellen, an denen mein Name steht, dürfen nicht verändert werden; die Stellen, an denen mein Name in den html-Ausgaben auftaucht, dürfen ebenfalls nicht verändert oder gelöscht werden.

Selbstverständlich übernehme ich keine Haftung jeglicher Art für irgendwelche Schäden, die direkt oder indirekt durch die Nutzung dieses Programmes entstehen könnten.

Fehlerberichte und Anregungen zur Weiterentwicklung sind jederzeit willkommen.

Ich freue mich auch über eine Benachrichtigungsmail, wenn das Programm irgendwo eingesetzt wird.

- Das Programm exchange.pl
- Dasselbe, aber im Browser lesbar
- Die Konfigurationsdatei
- Dasselbe, aber im Browser lesbar
- Die Sprachdateien

Ach, übrigens: Wenn Sie bei diesem Programm auf dem Laufenden bleiben wollen, können Sie den Dienst Changedetection.com verwenden. Einmal anmelden, die URL dieser Seite dort eintragen und schon bekommen Sie sofort (in einstellbaren Zeitabständen) eine E-Mail, wenn sich hier etwas ändert.