Häufige Probleme

Die häufigste Ursache für Probleme mit Contenido scheint zu sein, daß die Dateien und Verzeichnisse falsche Berechtigungen und Besitzer haben.

Es ist **wichtig**, daß derselbe Benutzer konsistent das gesamte Contenido-Verzeichnis besitzt. Beispiel:

Es kommt oftmals vor, daß Contenido den Besitzer "A" hat, die Dateien, die jedoch durch das Contenido bearbeitet werden, den Besitzer "B" haben. Sofern der SAFE_MODE in PHP ausgeschaltet ist, ist das kein Problem, sofern die Dateiberechtigungen stimmen, aber da in den meisten Fällen der SAFE_MODE an ist, darf Contenido als "A" nicht auf Dateien von "B" zugreifen, obwohl die Berechtigungen stimmen. Wichtig ist daher, daß sämtliche Verzeichnisse und Dateien denselben Besitzer haben!

Nachfolgendes kleines Script prüft, ob irgendwelche Verzeichnisse und/oder Dateien einen unterschiedlichen Besitzer haben. Einfach in eine Textdatei (z.b. permissioncheck.php) kopieren, hochladen und auf dem Webserver über einen Browser aufrufen.

```
<?php
echo "<pre>";
checkOwnerDir(getcwd()."/");
echo "checkPerms beendet.";
echo "</pre>";
function checkOwnerDir ($from_path)
{
        $old_path = $from_path;
    if (is_dir($from_path))
    {
        chdir($from_path);
        $myhandle=opendir('.');
        while (($myfile = readdir($myhandle))!==false)
        {
            if (($myfile != ".") && ($myfile != ".."))
            {
                if (fileowner($myfile) != getmyuid())
                {
                        $ownerInfo = posix_getpwuid(fileowner(
$myfile));
                        echo
"<font color=\"red
\">Fehler:</font> Der Besitzer \"".$ownerInfo["name"]."\" der Datei
".$from_path.$myfile." ist unterschiedlich
                        zu
 dem Besitzer (\"".get_current_user().
```

```
"\") des aktuellen Scriptes. Stellen Sie sicher, daß
die Datei den gleichen Besitzer (owner) erhält wie das Script.\n";
                }
                if (is_dir($myfile))
                {
                    checkOwnerDir ($from_path.$myfile."/");
                    chdir($from_path);
                }
            }
        }
        closedir($myhandle);
    }
    chdir($old_path);
    return;
}
?>
```

In der Unix-Welt gibt es zu jeder Datei einen Besitzer und die Rechte für diese Datei. Mit chmod hat das nichts zu tun, sondern mit chown. Ein Beispiel für ein übliches Directory Listing unter Linux:
Code:

```
drwxr-xr-x   10 alek202  users         600 Jul  2 11:37 .
drwxr-xr-x    5 wwwrun   wwwrun        120 Jun 23 12:59 ..
drwxr-xr-x    2 alek202  users         112 Jun 26 15:53 css
drwxr-xr-x    5 alek202  users         120 Jun 26 15:53 upload
```

Obiges Beispiel zeigt folgendes:

Der Besitzer für das aktuelle Verzeichnis (.) und die Verzeichnisse css sowie upload ist der Benutzer alek202 in der Gruppe users. Das "drwxr-xr-x" gibt die Rechte für diese Datei an. Mit chmod beeinflusst man die Rechte, mit chown den Besitzer sowie die Gruppe.

Folgende Annahme: Ein Script mit dem Benutzer "alek202" versucht auf eine Datei, die dem Benutzer "klaus303" gehört, zuzugreifen. Dies wird jedoch durch den SAFE_MODE verhindert, sodaß "alek202" trotz voller Zugriffsrechte auf dem Dateisystem nicht auf die Datei von "klaus303" zugreifen darf. Genau das passiert bei einem Upload: Die Datei wird hochgeladen, und technisch sieht es so aus, daß PHP die Datei in ein temporäres Verzeichnis legt. Genau DAS passiert aber mit den Rechten des Webservers: Die Datei, die hochgeladen wurde und jetzt in einem temporären Verzeichnis liegt, gehört z.b. "wwwrun" (Der Benutzer, unter dem der Webserver läuft). Das Script, welches die Datei aber bearbeiten muß, gehört "alek202" - und das geht durch den SAFE_MODE in die Hose. Laut PHP-Manual sollte die Funktion "move_uploaded_file" zwar die UID-Checks übergehen, in der Praxis

geht das aber leider doch nicht.

Was ist also jetzt zu tun?
Die einfachste Möglichkeit ist sicherlich, den SAFE_MODE abzuschalten. Wer dies nicht möchte oder kann, muß entweder Contenido komplett als Benutzer "wwwrun" (so heißt der Benutzer des Webservers meistens) laufen lassen (z.b. chown -R wwwrun contenido-4.3.1b) oder in der Datei php.ini die Direktive "safe_mode_gid" auf "on" zu setzen. "safe_mode_gid" ist dann nicht mehr ganz so streng, und auf obiges Beispiel angewandt würde das bedeuten: Wenn "alek202" und "klaus303" zu derselben Gruppe (z.b. "users") gehören würden, dürfte "alek202" auf die Dateien von "klaus303" zugreifen.

Eindeutige ID: #1014
Verfasser: Peter Beauvain
Letzte Änderung: 2007-06-27 00:46