

Da die meisten Webserver mit Linux betrieben werden, sollten wir einen Blick auf die Zugriffsrechte werfen. Das Linux-Dateisystem unterscheidet zwischen Eigentümer, Gruppe und Alle (bzw. Welt). Diese drei können die Rechte – **Lesen [r]**, **Schreiben [w]** und **Ausführen [x]** haben. Je nach Systemkonfiguration des Webserver, bekommt PHP einen eigenen Besitzer-Namen bzw. eine bestimmte Gruppenzugehörigkeit. XAMPP vergibt einer neuen, durch PHP erzeugten Datei (bzw. Verzeichnis) den Eigentümer **daemon** mit der Gruppe **daemon**. Damit kann auch nur mehr **daemon** eine Datei löschen – ausgenommen davon ist definitiv **root** oder ein Benutzer der Gruppe **daemon**.



Zugriffsrechte numerisch (über ein Oktal) definieren.

Die numerischen Rechte sind 4 = Lesen, 2 = Schreiben und 1 = Ausführen. Um Rechte zu kombinieren (z. B. Lesen und Schreiben) muss man einfach die Nummern addieren.

datei.txt **0777**

Prefix 0
Besitzer
Gruppe
Alle

Rechte

7 = Voll (alle Rechte)
6 = Lesen und Schreiben
5 = Lesen und Ausführen
4 = Nur Lesen
3 = Schreiben und Ausführen
2 = Nur Schreiben
1 = Nur Ausführen
0 = Keine



Die erste Ziffer steht für SetUid, SetGid und Sticky-Bits. Für unsere PHP Aufgaben reicht das Prefix 0.

PHP



chmod ()

chmod () ändert die Zugriffsrechte von Dateien. Der erste Parameter ist der Dateiname, der zweite das Oktal als Zahl.

Im Beispiel wird die Datei geheim.txt durch Zugriffsrechte geschützt. Ruft man die Datei im Browser auf, bekommt ein **403 – Zugriff verweigert** – ausgegeben.

```
<?php
$txtString = 'For your Eyes only';
file_put_contents('geheim.txt', $txtString);
chmod('geheim.txt', 0200);
?>
```



Beispiel für einen kurzzeitigen Zugriff auf eine Datei!

```
<?php
chmod('geheim.txt', 0400);
echo file_get_contents('geheim.txt');
chmod('geheim.txt', 0200);
?>
```