[www.MaFli.NET]

Wie kann man sich vor "Bilder-Klau" im Web schützen?

Dieser Beitrag erläutert die gängigsten Methoden, mit denen man versuchen kann, Bilder auf seiner Homepage vor dem „Bilder-Klau“ zu schützen. Es wird jedoch zu jeder Methode ebenfalls darauf eingegangen, wie man die Bilder trotz aller Schutzmaßnahmen abspeichern kann.

Für alle Ungeduldigen

Wer sich nicht alles durchlesen möchte, dem sei hier kurz und knapp auf die Frage:
„***Kann ich Bilder auf meiner Homepage 100%ig schützen?***“
die einzig mögliche Antwort gegeben: „***Nein!***“ *(siehe auch Punkt 11).*

Inhalt

1. Einleitung

Wer Bilder ins Internet stellt macht sie der Öffentlichkeit zugänglich, somit muss man damit rechnen, dass diese von anderen Nutzern ungefragt weiter verwendet werden. Nachfolgend werden die Möglichkeiten zur Verhinderung des „Bild-Klaus“ an Beispielen beschrieben sowie ihre Grenzen aufgezeigt.

2. Schutz mittels JavaScript

Hier soll ein Überblick über einige Versuche mit Hilfe von JavaScript ein Bild zu schützen gegeben werden.

2.1 Sperren des Kontextmenüs

Diese Art des Schutzes („Hausfrauen-Methode“) ist eine anscheinend weit verbreitete unter jenen Webmastern, die sich nicht wirklich mit dem Thema auseinander setzen, denn sie ist weiterhin die nervigste und uneffektivste Art seine Bilder zu schützen.
Bei dieser Art wird meist versucht die Bilder der Homepage durch ein JavaScript zu schützen, was die rechte Maustaste und somit das Kontextmenü blockiert. Dabei lassen viele Webmaster zusätzliche ein kleines Popup anzeigen, in dem meist ein Hinweis auf das Verbot steht. Mit der folgenden Anweisung im `<body>`-Tag des HTML-Dokumentes ist es möglich, auf der gesamten Seite das Kontextmenü zu unterbinden:
`<body onContextMenu="return false;">` und mit einem kleinen Zusatz wird das Popup mit dem Text eingefügt: `<body onContextMenu="alert('Du darfst den Quelltext nicht anschauen!');return false;">`

2.2 Sperren der rechten Maustaste

Diese Art des versuchten Schutzes ist im Prinzip nichts anderes, als die eben beschriebene Sperrung des Kontextmenüs, hier will ich jedoch beschreiben, wie man die rechte Maustaste und somit das Kontextmenü nur über sämtlichen Bildern auf der Webseite sperrt. Dazu muss in den `<head>`-Tag der Webseite folgendes JavaScript eingefügt werden:

```
<script language="JavaScript">
function schutz(e) {
  alert("Dieses Bild ist geschützt.");
  return false;
}
function bildschutz() {
  if(document.images)
    for(i=0;i<document.images.length;i++)
      document.images[i].onmousedown =
schutz;
}
</script>
```

Damit dieses Script wirksam wird muss in den `<body>`-Tag folgender Code eingefügt werden:

```
<body onLoad="bildschutz();">
```

2.3 Bildschutz durch die „Bild-weg?“-Methode

Mittels JavaScript hat man ebenfalls noch die Möglichkeit, das beim „darüberfahren“ mit der Maus, das Bild einfach komplett verschwindet. Somit kommt man auch nicht über die rechte Maustaste und dann über „Eigenschaften“ an den Pfad des Bildes. Man kann es sich nur betrachten, wenn sich die Maus nicht über dem Bild befindet. Der Befehl kann zum Beispiel so lauten:

```
<span onMouseOver="mein_bild.style.visibility='hidden'"
onMouseOut="mein_bild.style.visibility='visible'">
<img src="bilder/bild.jpg" id="mein_bild">
</span>
```

2.4 Was bewirken diese beiden Scripte bei Usern?

Außer, dass der Besucher wohl sehr verärgert sein wird, da man ihm damit die Funktion des Browser einschränkt, wird so ein Script nicht viel Wirkung zeigen. Ohne Kontextmenü ist es dem Besucher ebenfalls nicht gestattet per Mausklick die Seite neu zu laden bzw. zu wählen, ob Links auf der Seite in einem neuen Fenster geöffnet werden sollen oder dass er die Seite mit wenig Aufwand zu seinen Bookmarks hinzuzufügen kann. Und gerade mit dem letzten Punkt schießt man sich eigentlich selbst ins Bein...
Wer so ein Script auf einer Webseite „findet“, überlegt sich bestimmt in Zukunft zweimal, ob er noch einmal wiederkommt.

2.5 Wie kann man diesen Schutz umgehen?

Nunja, wie gesagt, wer als Webmaster so ein Script einsetzt, der sollte wirklich noch einmal darüber nachdenken, denn Möglichkeiten den Schutz zu umgehen gibt es viele, unter anderem:

- Indem man einen Screenshot macht.
- Indem man über das Menü geht und sich den Quelltext anzeigen lässt.
- Indem man ENTER drückt und dabei die rechte Maustaste gedrückt hält, wird das Kontextmenü trotzdem angezeigt.
- Indem man im Browser JavaScript abschaltet:
  Wobei solche "No-Rechtsklick-Scripte" in vielen Mozilla und Netscape-Versionen auch bei aktiviertem JavaScript wirkungslos sind.
  JavaScript abschalten:
  Internet Explorer: Menü "Extras" -> "Internetoptionen" -> Register "Sicherheit", Zone "Internet" anklicken, dann "Stufe anpassen", nach unten scrollen bis Bereich "Scripting" erscheint, und dort "Deaktivieren".
  Netscape Navigator (und Mozilla): Menü "Bearbeiten" -> "Einstellungen" -> Kategorie "Erweitert", auf der rechten Seite das Häkchen bei "JavaScript aktivieren" entfernen.
  Opera (ab v7.0): Menü "Datei" -> "Eigenschaften" -> Häkchen bei "JavaScript aktivieren" entfernen.

3. Schutz mittels HTML

Hier soll ein Überblick gegeben werden, wie versucht werden kann, mit HTML-Mitteln Bilder zu schützen.

3.1. Bildschutz mit einem „Blind.gif“

Ein so genanntes *Blind.gif* wird über das eigentliche Bild "gelegt", das hat zur Folge, beim abspeichern wird nicht das richtige Bild gespeichert, sondern das *Blind.gif*.
Vorgang: mittels eines Bildbearbeitungsprogramms erstellt man eine neue Datei mit der Größe von 1x1 Pixel und transparenten Hintergrund. Diese speichert man im GIF-Format ab. Nun erstellt man in der HTML-Datei - an der Stelle wo das Bild erscheinen soll - eine Tabelle (eine Spalte, eine Zeile), als Tabellenhintergrund wird nun das Originalbild eingefügt. Das *Blind.gif* wird in die Zelle eingefügt, und auf die Größe des Originalbildes vergrößert. Der Quelltext könnte folgendermaßen aussehen:

```
<table background="bilder\originalbild.jpg" cellspacing="0"
cellpadding="0">
<tr><td><img src="images\blind.gif" height="150" width="150"></td></tr>
</table>
```

Wenn nun der Besucher das Bild abspeichert, speichert er nur das *Blind.gif* ab, nicht aber das eigentliche Bild. Deshalb könnte man sich noch überlegen, ob man dem *Blind.gif* auch den Namen „*Blind.gif*“ gibt, denn beim Versuch das Bild abzuspeichern wird der Bildname ja im „Speichern unter…“-Dialog angezeigt (Betriebssystem sei Dank), sodass sich jemand mit bösen Absichten denken kann, was los ist ;-) und somit gleich eine andere Methode wählt.

3.1.1. Schutz umgehen?

Weiß man, dass der Webmaster *Blind.gifs* nutzt, kann man sich im Quelltext der Homepage den Pfad zu dem Bild heraussuchen, in die Adresszeile des Browsers kopieren und das Bild direkt aufrufen. Hat der Webmaster nicht an den Browsercache gedacht (siehe Punkt 5.6) hat man das Bild sowieso schon auf dem Rechner, und muss nur noch wissen, wo man hin greifen muss. Außerdem hilft der bewährte Screenshot.

3.2. Bildschutz durch Bildteilung

Bei dieser Methode teilt man das Bild in mehrere Teile auf – ähnlich einem Puzzle. Man „zerschneidet“ das Bild mit einem Bildbearbeitungsprogramm in beispielsweise drei Teile. Mittels einer unsichtbaren Tabelle kann man nun die Teile auf der Webseite wieder als ganzes zusammensetzen. Ein positiver Effekt bei dieser Art: Das Bild wird schneller geladen. Beispielcode:

```
<table border="0" cellpadding="0" cellspacing="0">
<tr><td><img src=""fotos\bild_teil1.jpg"" border="0"></td></tr>
<tr><td><img src=""fotos\bild_teil2.jpg"" border="0"></td></tr>
<tr><td><img src=""fotos\bild_teil3.jpg"" border="0"></td></tr>
</table>
```

3.2.1. Schutz umgehen:

Die Teilung des Bildes ist kein Schutz im eigentlichen Sinne, sondern nur ein Hindernis, um das Bild auf ganz plumpe Weise komplett kopieren. Man muss sich nur alle Teile des gesamten Bildes herunterladen, und mit einem Bildbearbeitungsprogramm einfach wieder zusammensetzen. Oder auch hier: Einfach mittels Screenshot abspeichern.

4. Signaturen, Wasserzeichen und Copyright-Hinweise

Die weitere Art seine Bilder zu schützen bietet die digitale Bildsignatur und die Signatur mittels Wasserzeichen. Mit diesen Arten kann zwar ebenfalls nicht der "Bild-Klau" verhindert werden, jedoch kann bei der digitalen Signatur rechtlich gegen Personen vorgegangen werden, welche die Bilder unrechtmäßig bzw. wider Willen des Fotografen publizieren. Bei einem Wasserzeichen hat der „Dieb“ keine Freude am Bild, da ständig das Wasserzeichen zu sehen ist.

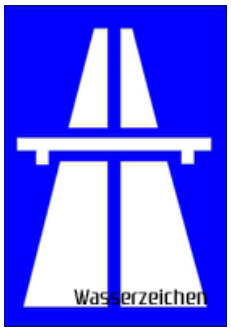
4.1. Digitale Bildsignatur

Wie bereits kurz angedeutet, ist diese Art die effektivste, um sich rechtlich gegen den „Bild-Klau“ zur Wehr zu setzen, allerdings gibt es diesen Schutz nicht umsonst. Bei der Signatur werden Informationen –unsichtbar- in das Bild eingebracht, ähnlich der digitalen Signatur bei eMails. Eine Firma die diesen Schutz anbietet, ist Digimarc (http://www.digimarc.com). Man muss eine Urheber-ID beantragen. Die gibt es kostenlos, allerdings nur für knapp 100 Bilder und für ein Jahr – eine Art Testlizenz. Danach muss für diesen Service gezahlt werden, zurzeit ist man mit $49 für maximal 100 Bilder (in der Personal-Edition), die mit Namen und Adresse versehen werden, als Wasserzeichen dabei. Digimarc verspricht allerdings, dass man das Wasserzeichen so gut wie gar nicht sieht, und dass das Wasserzeichen selbst einen Ausdruck und ein erneutes Einscannen überlebt.

4.2. Wasserzeichen

Wasserzeichen kann man mit einem beliebigen, besseren Bildbearbeitungsprogramm (für Links siehe Punkt 93) selbst erstellen/einfügen. Dabei „legt“ man ein Bild seiner Wahl (meist Logo, oder Schriftzug) mit einer höheren Transparenz einfach über das zu schützende Bild, sodass man mit dem nun entstandenen „Bild im Bild“ nicht mehr viel anfangen kann, wenn man es „klaut“ und seinerseits publizieren möchte. Da das Wasserzeichen sichtbar ist, ist das natürlich auch ein Nachteil beim betrachten – ja nach Wasserzeichengröße. Dafür kostet diese Methode im Gegensatz zur digitalen Bildsignatur nichts, lässt sich relativ einfach erstellen und ist auch wirkungsvoll.

Beispiele für Wasserzeichen:



4.2.1. Copyright als Wasserzeichen

Eine weitere Möglichkeit ohne das ganze Bild mittels eigenem Wasserzeichen zu „verschandeln“ ist die Einbettung eines Copyrighthinweises. Dabei fügt man wiederum mit einem Bildbearbeitungsprogramm einfach seinen Namen, oder seine eMail-Adresse bzw. seine Internetadresse in das Bild ein. Theoretisch müsste man den Hinweis ebenfalls quer über das ganze Bild einfügen. Wird er nämlich am Rand eingefügt, besteht immer noch die Möglichkeit, dass man das Bild bis dahin abschneidet, und weiterverwendet.

Ein Beispiel

5. Bildschutz durch .htaccess und PHP

Mit .htaccess wird ein Mechanismus geboten, den Zugriff auf Webserver zu kontrollieren. Zum Beispiel kann man ein bestimmtes Verzeichnis mit einem Passwort schützen. Man kann so seine Bilder in ein .htaccess-geschütztes Verzeichnis auf den Webserver laden. Wird dann das Bild direkt über die Adresszeile (URI) des Browsers angesprochen kommt die Aufforderung Name und Passwort einzugeben. Somit ist kein Zugriff auf das Bild möglich. Allerdings unterstützen nicht alle (vor allem die wenigsten Anbieter von kostenlosem Webpspace) diese Techniken, da sie auf dem Server eingerichtet werden müssen. Einen sehr guten Artikel über den Bildschutz mit .htaccess und PHP (mit Beispiel) gibt es von Marco Glatz unter: http://www.ideenreich.com/php/php_bilderklau.shtml.

5.1.1. Wie kann man den Schutz umgehen?

Indem man einen Screenshot vom Bildschirm macht.

6. Bildschutz durch FLASH

Manchmal kommt es im Web auch vor, dass Webmaster den Bilder-Klau verhindern möchten, indem sie die Bilder einfach in ein FLASH-File einbinden. Hier sollte als erstes bedacht werden, dass FLASH Clientläufig ist, somit wird das ganze FLASH-File noch vor dem Anzeigen in den Cache des Browsers geladen.

6.1. Wie kann man den Schutz umgehen?

Mit einem Screenshot kommt man an die Bilder heran, dazu muss man noch nicht einmal weiter Online bleiben, da das Flash-File im Browsercache zwischengespeichert ist.

7. Bilder außerhalb des Document-Root-Verzeichnisses

Durch verschieben der Bilder in einen Ordner vor dem „Document-Root“-Verzeichnis (d.h. in einen anderen Ordner, als der, in dem die komplette Homepage gespeichert wird), ist kein Zugriff per HTTP möglich. Das bedeutet, wenn man den Bildpfad in die Adresszeile kopiert, wird das Bild nicht gefunden, da es nicht im Stammverzeichnis der Homepage liegt (man kann die Bilder allerdings problemlos mittels relativer Pfadangabe in seinen Scripten einbinden). Diese Art des Schutzes kann man allerdings nur nutzen, wenn man seinen Speicherplatz bei einem Webspace-Anbieter hostet, der diesen Zugriff auch gewährt. Bei kostenlosen Anbietern ist so etwas nicht möglich.

7.1. Wie kann man den Schutz umgehen?

Indem man auch hier wieder einen Screenshot vom Bildschirm macht oder im Browsercache nachschaut.

8. Mögliche Schutzmaßnahmen

Hier soll noch einmal eine Schutzmaßnahme erklärt werden, die den Bilder-Klau erschwert, aber eben auch nicht verhindern kann. Dabei wird alles vorher beschriebene berücksichtigt.

Cachen von Dateien verhinden

Um das cachen zu verhindern, fügt man folgende Meta-Tags ein in den `<head>`-Bereich der HTML-Seite ein:
`<meta http-equiv="pragma" content="no-cache">` - verhindert das Speichern der Seite im Cache des Proxy-Servers
`<meta http-equiv="cache-control" content="no-cache">` - verhindert das Speichern der Seite im Browsercache.
Somit werden die Seiten und auch die entsprechenden Bilder nicht auf dem Rechner des Benutzers gespeichert.

Bilder außerhalb von „Document-Root“

Wer die Möglichkeit besitzt, sollte seine Bilder außerhalb des „Document-Root“-Verzeichnisses ablegen (siehe Punkt 7).

Statuszeilentext unterdrücken

Wird ein Link direkt auf ein Bild gesetzt, ist der Pfad in der Statuszeile des Browsers sichtbar (sofern die Statuszeile angezeigt wird - Benutzerabhängig), dies kann man mit dem folgenden Befehl in einem Link verhindern:
`<a href="bilder/originalbild.jpg" onMouseOver="status='Mein Bild'" onMouseOut="''">` - dies funktioniert jedoch wiederum nur, wenn im Browser JavaScript aktiviert ist.

Prüfen ob JavaScript aktiviert ist

Bezüglich JavaScript (JS) sollte man überlegen, ob man den Zugriff auf die Homepage ohne aktiviertes JS überhaupt zulässt (wer viele JScripte nutzt, wird sich das vielleicht auch schon überlegt haben). Somit würde sogar die „Hausfrauenmethode“ (Punkt 2.1) in fast allen Browsern laufen, da JS aktiviert sein muss. Wer bei seinem nächsten Besuch im Internet einmal bewusst JS abschaltet, wird merken, wie viele Homepagebetreiber und auch Firmen den Zutritt ohne JS einschränken bzw. ganz verwähren.
Prüfen, ob JS im Browser aktiviert ist, lässt sich direkt mit HTML-Mitteln. Dazu existiert der Tag: `<noscript></noscript>`.

Der Inhalt wird nur ausgeführt, wenn kein JS im Browser aktiviert ist, sonst bewirkt dieser Tag nichts. Mit folgendem Quelltext kann bspw. eine Weiterleitung auf eine alternative Seite erfolgen, um einen Hinweis auszugeben.

```
<noscript>
<meta http-equiv="refresh" content="0; URL=kein_js_aktiviert.html">
</noscript>
```

Quellcode kodieren (UNICODE)

Mit Hilfe des Unicodes ist es möglich, den gesamten Quelltext so zu kodieren, dass er ohne externes Programm nicht wirklich verständlich gelesen werden kann. Man kann natürlich jedes Zeichen selbst kodieren (Liste: http://de.selfhtml.org/html/referenz/zeichen.htm#benannte_iso8859_1) oder aber ein Programm nutzen, was durchaus sinnvoller erscheint. In diversen Programmen lässt sich wiederum angeben, ob man eine numerische oder hexadezimale Darstellungsform nutzt. Auf diese Weise lassen sich auch sehr gut eMail-Adressen kodieren, sodass diese nicht von entsprechenden Programmen ausgelesen werden können.
Einen HTML-UNICODE Konverter gibt es hier: http://www.mafli.net/programme/unicode/

9. Bandbreitenklau (Bandwidth Stealing)

Das Laden von Inhalten von einem anderen Server aus, so dass sie wie Inhalte der eigenen Webseite erscheinen, nennt man Bandbreitenklau. Typischerweise sind dies oft Bilder oder Musik. Jeder Aufruf eines Dokumentes (egal ob HTML-Datei, Bilder, Musik,...) auf dem Server verursacht so genannten Traffic (Datentransfer). Dieser ist je nach Hostingangebot, welches der geschädigte Webmaster nutzt, eingeschränkt. Somit ist klar, wenn immer nur auf das entsprechende Dokument verlinkt wird, wird die Trafficgrenze schneller erreicht. Wird sie überschritten, muss der Webmaster sogar noch draufzahlen, was meist nicht ganz so billig ist. Vor dieser ungewollten Verlinkung kann man sich –zumindest bei Bildern– schützen.

9.1. Bandbreitenklau aufdecken

Meistens bleiben solche Fälle unaufgedeckt. Bei vielen Webhostern gibt es allerdings zu jeder Site eine Log-Datei (meist access.log), in der unausgewertet alle Zugriffsdaten stehen. Diese Datei kann man sich herunterladen und, beispielsweise, in einer Tabellenkalkulation anschauen. Wenn nun hinter dem Namen einer Grafik eine fremde URL steht, ist davon auszugehen, dass das Bild von jemand anderem verlinkt wurde. Durch Aufruf der dort stehenden URL kann man einfach nachschauen, ob das stimmt.

9.2. Bandbreitenklau verhindern (.htaccess)

Mittels der Datei .htaccess kann ebenfalls abgefragt werden, woher die Anfrage an die Bilddatei kommt. Wenn diese nicht mit einer vordefinierten URL übereinstimmt, kann ein rotes X (keine Grafik gefunden), oder alternativ eine Austauschgrafik angezeigt werden. Dazu muss der folgende Code in die Datei .htaccess geschrieben werden.
Diese Methode funktioniert nur, wenn beim Server die Variable **Rewrite_mod** aktiviert ist!

```
RewriteEngine on
RewriteCond %{HTTP_REFERER} !^$
RewriteCond %{HTTP_REFERER} !^http://(www\.)?mafli\.net(/.*)?$ [NC]
RewriteRule \.(gif|jpg|jpeg|png|GIF|JPG|JPEG|PNG)$
http://www.mafli.net/images/ersatz.gif [R,L]
```

Die grünen Einträge müssen durch das entsprechende ***eigene*** Domain und dem ***eigenen*** Pfad ersetzt werden.

10. „Bild-Klau“ trotz aller Schutzmaßnahmen

Der Bildklau lässt sich –wie ihr hoffentlich bisher mitbekommen habt– nie verhindern können - der „Druck“-Taste auf der Tastatur sei Dank! Möchte man also ein Bild auf seinem Rechner haben, macht man einfach einen Screenshot vom aktuellen Bildschirminhalt. Den muss man letztendlich nur noch in einem Grafikprogramm (einfachster Weise ***Paint*** bei Windows) oder einem anderen Bildbetrachter (siehe Punkt Links 43) über das Menü ***Bearbeiten à Einfügen*** einfügen, und abspeichern. Schon hat man das Bild erfolgreich „geklaut“. Dagegen ist jeder der hier aufgeführten Methoden zwecklos.
Eine weitere Methode ist, die Seite mittels allem vorhandenen Grafiken auf seinem Rechner abzuspeichern (bis auf Variante „htaccess“).
Im Internet-Explorer/Netscape/Mozilla über Menü ***Datei à Speichern unter… à Dateityp: komplette Website*** die Seite mit allem Bildern und Texten auf der lokalen Festplatte sichern. Oder –bequemer- man nutzt einfach Programme wie WebStripper. Dort kann man noch einige Einstellungen mehr vornehmen, so kann man sich beispielsweise nur alle Bilder bis zur zweiten Pfadebene herunterladen lassen usw…

11. Was haben wir gelernt?

Es gibt keinen Schutz, um Bilder vor dem „Bild-Klau“ zu schützen, es existiert lediglich ein rechtlicher Schutz mittels der digitalen Signatur. Wer möchte, dass niemand seine Bilder klauen kann, sollte sie sich ausdrucken, zu Hause in ein Fotoalbum kleben und im Schrank einschließen, aber nicht im Internet publizieren.

12. Nützliche Links zum Thema:

Kostenloser Bildbetrachter http://www.irfanview.de
Kostenloses Bildbearbeitungsprogramm http://www.gimp.org

Digitaler Bildschutz (Wasserzeichen) http://www.bildschutz.de
Digitale Wasserzeichen http://www.dicimarc.com

HTML-Dokumentation: http://de.selfhtml.org
PHP-Software: http://www.php.net
Webserver (Voraussetzung für PHP): http://www.apache.org

[www.MaFli.NET]

WebStripper (Download kompletter Homepages): http://download.pchome.de/448.htm

MaFli HTML-UNICODE Konverter: http://www.mafli.net/programme/unicode/

13. Weitere informative Seiten

http://www.trafficklau.de
http://www.bildschutz.de
http://www.computerhilfen.de

Gruß Mario (mario@mafli.net)
http://www.mafli.net

Hinweise
Wenn ich hier den Begriff des Webmasters verwende, bezieht sich dieser ebenfalls auf alle Webmaitressen :-)
Script-Version: 1.0 (03.04.05)