Sicherer Bilder Upload mit PHP

== Lösung ==
Quellcode

```
$imageinfo = @getimagesize($_FILES['datei']['tmp_name']);
$allowed = array('image/gif', 'image/jpeg', 'image/png');
if(!$imageinfo || !isset($imageinfo['mime']) || !in_array($imageinfo['mime'], $allowed)) {
throw new Exception('bildformat nicht erlaubt');
}
$uploaddir = 'uploads/';
$uploadfile = $uploaddir . basename($_FILES['datei']['name']);
if (move_uploaded_file($_FILES['datei']['tmp_name'], $uploadfile)) {
echo "Upload erfolgreich.";
} else {
echo "Upload fehlgeschlagen.";
}
```

Alles anzeigen

== Erläuterungen ==
Der Code behandelt bereits einige Sicherheitsmaßnahmen. Der Content Typ von Bildern, der mittels $_FILES['datei']['type'] übermittelt wird ist nicht sicher. Dieser lässt sich mit wenigen Handgriffen einfach manipulieren. Auch auf die Dateiendung kann man sich nicht verlassen - zumal sich die Prüfung durch Einsatz von Steuerzeichen wie \0 umgehen lässt.
Durch den Aufruf von basename wird verhindert, dass man unerlaubte Sonderzeichen, wie Pfadangaben in den Dateinamen schmuggeln kann.

== Einschränkung ==
Das Verfahren sichert nur den Upload ab. Bietet aber weiterhin eine Angriffsfläche, falls andere Teile ihres Programmes unsicher sind.
Hinterlegt man z.b. Kommentare direkt im Bild (mit manchen Bildbearbeitungsprogrammen ist dies möglich) - so könnte man hier Code einfügen.
Dieser Code wird eingefügt, wenn man
a) erlaubt die Bilder per include zu laden
b) der Webserver die Ausführung von Dateien mit einer Bildendung an den PHP Interpreter weitergibt

== Mehr Sicherheit ==
Noch mehr Sicherheit erhält man eigentlich nur, indem man das Bild mit einer PHP Funktion (imagecreatefrom) selbst nochmal abspeichert.
Dies birgt jedoch die Gefahr, dass einige Bilder zu groß für den maximal für PHP verfügbaren Arbeitsspeicher sind. Und bildet über den Performanceengpass eine weitere Angriffsfläche.

== Literatur ==
Mehr Informationen in englischer Sprache finden Sie unter:

- scanit.be/uploads/php-file-upload.pdf

== Demo ==
Eine Live Demo des Upload Tools und den dazugehörigen Quelltext finden Sie hier: demo.easy-coding.de/php/sicherer-bilder-upload-mit-php.

Quellcode

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" dir="ltr" xml:lang="de">
<head>
<meta http-equiv="content-type" content="text/html; charset=UTF-8" />
<title>PHP: Sicherer Bilder Upload</title>
</head>
<body>
<?php
if(isset($_FILES['datei'])) {
$imageinfo = @getimagesize($_FILES['datei']['tmp_name']);
$allowed = array('image/gif', 'image/jpeg', 'image/png');
if(!$imageinfo || !isset($imageinfo['mime']) || !in_array($imageinfo['mime'], $allowed)) {
throw new Exception('bildformat nicht erlaubt');
}
$uploaddir = 'uploads/';
$uploadfile = $uploaddir . basename($_FILES['datei']['name']);
if (move_uploaded_file($_FILES['datei']['tmp_name'], $uploadfile)) {
echo "Upload erfolgreich.";
} else {
echo "Upload fehlgeschlagen.";
}
}
?>
<form action="" method="post" enctype="multipart/form-data">
<fieldset>
<legend>Datei auswählen</legend>
<input type="file" name="hiddendata" name="datei" />
<input type="submit" value="Upload starten" />
</fieldset
</form>
</body>
</html>
```

Alles anzeigen