

Schutz gegen E-Mail-Spam

In diesem Tutorial möchte ich die Möglichkeiten aufzeigen, mit denen man sich gegen „Address-Harvester“ schützen kann.

„Address-Harvester“ sind Programme, welche eine Web-Site nach E-Mail-Adressen durchsucht um diese dann zu vermarkten bzw. für Werbezwecke zu verwenden.

Ich beschränke mich hierbei ausschließlich auf Möglichkeiten die in der Programmierung einer Web-Site Anwendung finden könnten.

Im Folgenden werde ich folgende Ansätze vorstellen:

1. Die bewusste „verfälschte“ Darstellen einer E-Mail-Adresse [Seite 2]
2. Das Darstellen einer E-Mail-Adresse per Bilder [Seite 3]
3. Die dynamische Darstellung der E-Mail-Adresse per JavaScript [Seite 4]
4. Die dynamische Darstellung der E-Mail-Adresse per PHP [Seite 5]
5. Die Darstellung einer E-Mail-Adresse mit CSS [Seite 6]

1. Die bewusste „verfälschte“ Darstellen einer E-Mail-Adresse

Bei einer „verfälschten“ Darstellung der E-Mail-Adresse soll dem Address-Harvestern die Möglichkeit genommen werden eine E-Mail-Adresse zu erkennen bzw. korrekt zu erkennen.

Vorteile

- einfach zu erstellen
- ohne JavaScript und PHP

Nachteile

- benötigt ggf. eine Erläuterung für den User
- nicht für einen Link geeignet

Auch dazu gibt es verschiedenen Möglichkeiten die allesamt recht einfach zu realisieren sind.

1.1 Man schreib anstelle eines @ die Zeichenfolge (at) in die E-Mail-Adresse.

Beispiel:

[Hans\(at\)Muster.de](#) anstelle von [Hans@Muster.de](#)

Für den User ist eine solche Lösung recht einfach zu verstehen und führt dabei kaum zu Missverständnisse.

1.2 Man erstellt eine E-Mail-Adresse in der einige Zeichen zuviel enthalten sind. Der User erhält dabei eine Erläuterung welche Zeichenfolgen zu entfernen sind.

Beispiel:

[HansXYZ@Muster.de](#) (XYZ entfernen)

Eine solche Lösung führt schon mal eher zu Missverständnissen da manche User eine Web-Site nicht genau lesen. Es ist somit unbedingt erforderlich die Erläuterung unmissverständlich und eindeutig dazustellen.

2. Das Darstellen einer E-Mail-Adresse per Bilder

Die Darstellung einer E-Mail-Adresse per Bild ist relativ einfach und effektiv. Sie benötigen dazu lediglich ein einfaches Bildbearbeitungsprogramm wie Paint. Mit einem solchem Programm erstellen Sie ein kleines Bild mit der darzustellenden E-Mail-Adresse. Dieses Bild wird anschließend an der entsprechenden Stelle im Quellcode eingefügt.

Vorteile

- einfach zu erstellen
- ohne JavaScript und PHP

Nachteile

- man benötigt ein Bildbearbeitungsprogramm
- nicht für einen Link geeignet

Beispiel:

Hans@Muster.de

Möchten Sie mehrere unterschiedliche E-Mail-Adressen auf Ihrer Web-Site einbringen, dann empfiehlt es sich nicht die ganze E-Mail-Adresse als Bild zu erstellen, sondern nur das @-Zeichen. Sie setzen nun das Bild mit dem @-Zeichen an der entsprechenden Stelle in Quellcode ein.

Beispiel:

Hans @ Muster.de

Bei einer solchen Lösung sollten Sie jedoch beachten, dass die Bilder dem Schriftgrad Ihrer Web-Site angepasst sind. Bei einer späteren Änderung sollten Sie jedoch bedenken, dass Sie ggf. auch die Bilder ändern müssen.

3. Die dynamische Darstellung der E-Mail-Adresse per JavaScript

Bei dieser Methode wird die E-Mail-Adresse über eine JavaScript-Funktion an der entsprechenden Stelle im Quelltext ausgegeben.

Dabei wird die Zeichenkette der E-Mail-Adresse aus vordefinierten Variablen zusammengesetzt. Da viele Address-Harvester angeblich kein JavaScript interpretieren können (derzeit zumindest), stellt diese Lösung einen relativ robusten Schutz dar.

Vorteile

- für einen Link geeignet

Nachteile

- der Browser muss JavaScript interpretieren können
- ein wenig Programmieraufwand

Beispiel:

```
<script type="text/javascript" language="JavaScript">
<!--
var name = "Hans";
var domain = "Muster.de";
document.write("<a href=\"mailto:" + name + "@" + domain + "\">");
document.write(name + "@" + domain + "</a>");
//-->
</script>
```

Die Ausgabe: Hans@Muster.de

Selbstverständlich muss der Browser den der User nutzt auch JavaScript verstehen. In der Regel ist da heutzutage von aus zu gehen.

4. Die dynamische Darstellung der E-Mail-Adresse per PHP

Bei dieser Methode wird die E-Mail-Adresse per PHP an der entsprechenden Stelle im Quellcode ausgegeben. Vorher wird die E-Mail-Adresse mit dem „hash-verfahren“ umgewandelt. Das Ergebnis ist eine E-Mail-Adresse im Quellcode, welche als solche nicht erkannt wird.

Unsere E-Mail-Adresse Hans@Muster.de wird dabei in die Zeichenfolge „%48%61%6E%73%40%4D%75%73%74%65%72%2E%64%65“ umgewandelt.

Da ein User diese Zeichenkette nicht interpretieren kann, ist es nicht möglich ihm diese E-Mail-Adresse als Link anzubieten. Anstelle dessen wird hier der Schriftzug „Kontakt“ als Link angeboten.

Vorteile

- für einen Link geeignet

Nachteile

- die E-Mail-Adresse ist im Browser nicht darstellbar
- ein wenig Programmieraufwand
- man benötigt einen Server der PHP interpretieren kann

Beispiel:

```
<?php
function no_spam($mail) {
    $str = "";
    $a = unpack("C*", $mail);
    foreach ($a as $b)
        $str .= sprintf("%%%X", $b);
    return $str;
}

$mail = no_spam("Hans@Muster.de");
$link = "<a href=\"mailto:\"" . $mail . "\">Kontakt</a>";
echo $link;
?>
```

Die Ausgabe: Kontakt

Fairerweise muss ich sagen, dass ich diesen tollen Lösungsvorschlag im Internet gefunden habe.

5. Die Darstellung einer E-Mail-Adresse mit CSS

Bei dieser Lösung wird die E-Mail-Adresse mit einen Text-String verändert. Dieser Text-String wird dabei per CSS ausgeblendet. Die Address-Harvester sind derzeit noch nicht in der Lage CSS zu interpretieren. Daher ist dieses eine recht effektive Lösung eine E-Mail-Adresse zu schützen.

Vorteile

- einfach zu erstellen
- ohne JavaScript und PHP

Nachteile

- nicht für einen Link geeignet

Beispiel:

```
Hans<span style="display:none">jetztKommtCSS</span>@Muster.de
```

Die Ausgabe: Hans@Muster.de