

Social-Media-Buttons datenschutzkonform nutzen

Notizbuch: Webshoprecht

Erstellt: 14.03.2016 14:07

Schlaawö... Datenschutz, Social Media

Quell-URL: <http://www.heise.de/ct/ausgabe/2014-26-Social-Media-Buttons-datenschutzkonfo...>

Schützen und teilen

Social-Media-Buttons datenschutzkonform nutzen



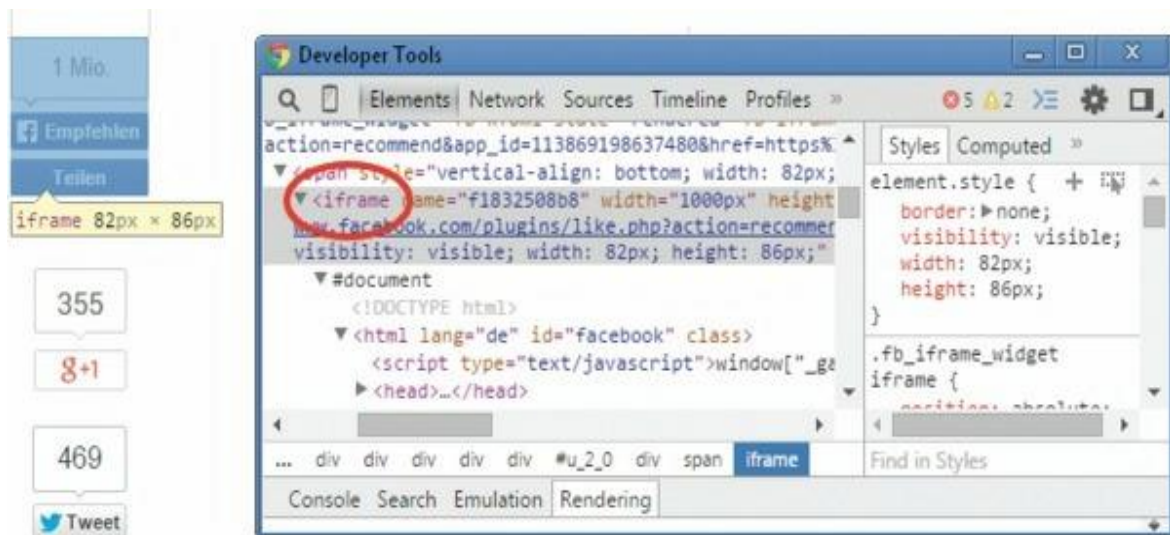
[Daniel Berger [28.11.2014

[Datenschutz, Sharing, Social Media, Social-Media-Buttons, Soziale Medien

Lädt eine Website die offiziellen Teilen-Buttons von Facebook, Google+ und Twitter, übertragen diese persönliche Daten. Dass es auch anders geht, zeigt Shariff: Der von uns entwickelte Nachfolger der bisherigen Zwei-Klick-Lösung erspart einen Klick und ist dennoch datenschutzkonform.

So ziemlich alles im Netz lässt sich in sozialen Netzwerken mögen und teilen: Selbst auf Pornoseiten sind Facebooks „Gefällt mir“-Buttons zu finden – oft begleitet von Googles „+1“-Knopf und dem Twitter-Vögelchen. Die meisten User wollen eher andere Inhalte teilen, nämlich witzige Videos, spannende Artikel und süße Katzen-GIFs. Für Seitenbetreiber lohnt es sich, dazu Sharing-Funktionen anzubieten: Insbesondere junge Internetnutzer verteilen ihre Web-Entdeckungen großzügig auf Facebook und stoßen dort wiederum auf andere Empfehlungen. Wessen Inhalte oft geteilt werden, der kriegt mehr Besucher, mehr Traffic, mehr Feedback.

Kleine Verfolger



Die offiziellen Buttons zum Teilen von Inhalten auf Facebook, Google+ und Twitter sind auf vielen Websites als iframe eingebettet. Darunter leiden Datenschutz und Privatsphäre.

Üblicherweise sind die Empfehlungsbuttons mit iframes in Webseiten eingebunden. Der Code in diesen Rahmen stammt direkt von Facebook, Google und Twitter – und

das ist ein Problem: Bereits beim Laden der Teilen-Knöpfe sendet der Browser persönliche Daten wie die IP-Adresse oder lokal abgelegte Cookies an die sozialen Dienste. Das passiert auch dann, wenn ein Webseitenbesucher gar kein Konto bei Facebook, Google und Twitter hat oder die Teilen-Funktion nicht nutzen möchte.

Noch gläserner sind eingeloggte Nutzer: Facebook etwa identifiziert sie eindeutig und erfährt so, auf welchen Seiten sie unterwegs sind. Das US-Unternehmen wertet die URL der besuchten Seite sowie den Zeitpunkt des Besuchs aus. Diese Informationen nutzt Facebook beispielsweise für „Werbeanzeigen basierend auf deiner Nutzung von Webseiten oder Apps außerhalb von Facebook“. Besuchen Sie also die Site Ihrer Lieblingsband, auf der ein Like-Button eingebettet ist, sehen Sie später bei Facebook eine Anzeige, die deren neues Album bewirbt. Twitter verfährt ähnlich bei seinen „Tailored Ads“, für die das Unternehmen Webseitenbesuche mit Hilfe der Twitter-Buttons auswertet. Zwar erlauben Facebook und Twitter ein Opt-out, damit das Surfverhalten nicht als Grundlage für Werbeanzeigen dient. Das verhindert aber kein Tracking.

Zwei Klicks für mehr Datenschutz



Zwei Klicks für mehr Datenschutz: Die Teilen-Buttons bleiben deaktiviert und übertragen keine persönlichen Daten an Facebook, Google und Twitter – bis der Nutzer sie einschaltet.

Zu den neugierigen Teilen-Buttons, die Daten schon beim Aufruf übertragen, gibt es datenschutzfreundliche Alternativen, welche die Privatsphäre schützen. Eine davon

ist die Zwei-Klick-Lösung, die wir vor drei Jahren in c't vorgestellt haben [1]. Unter dem Motto „Zwei Klicks für mehr Datenschutz“ hat sich die Lösung schnell verbreitet: So nutzen tausende andere Websites unsere Technik, darunter etwa die Stadtportale von München und Stuttgart.

Bei der Zwei-Klick-Variante ruft der Nutzer erst durch einen Klick die eigentlichen Teilen-Knöpfe auf. Vorher findet keine Datenübertragung statt; ein Tracking durch Dritte ist deshalb nicht möglich – jedenfalls nicht über die Knöpfe. Durch den Klick erklärt sich der Anwender mit der Datenübertragung einverstanden und kann dann Inhalte empfehlen, teilen, twittern.

Diese Lösung stellt immer noch eine sinnvolle Alternative zu den offiziellen Standard-Buttons dar. Allerdings haben sich im Produktiveinsatz kleine Nachteile für Webseitenbetreiber offenbart: Zum einen verleiten die Buttons nicht allzu sehr, Inhalte impulsiv und schnell zu teilen – zwei Klicks sind eben doch eine Hemmschwelle. Zwar bietet die Zwei-Klick-Variante die Möglichkeit, einzelne Buttons dauerhaft zu aktivieren. Aber durch diese Bequemlichkeit werden wieder Daten übertragen, selbst wenn Besucher die Funktion gar nicht nutzen.

Das zweite Problem ist die unauffällige Gestaltung der Empfehlungsbuttons: Die ausgegrauten Flächen verdeutlichen zwar den inaktiven Zustand, springen aber nicht so sehr ins Auge wie die bunten Originale von Facebook, Twitter und Google. Letztere ermuntern Besucher viel stärker dazu, Inhalte zu teilen. Unsere Kollegen in der Heise-Webentwicklung haben sich deshalb Gedanken gemacht und mit Shariff eine verbesserte Version der Zwei-Klick-Lösung programmiert.

A New Shariff in Town

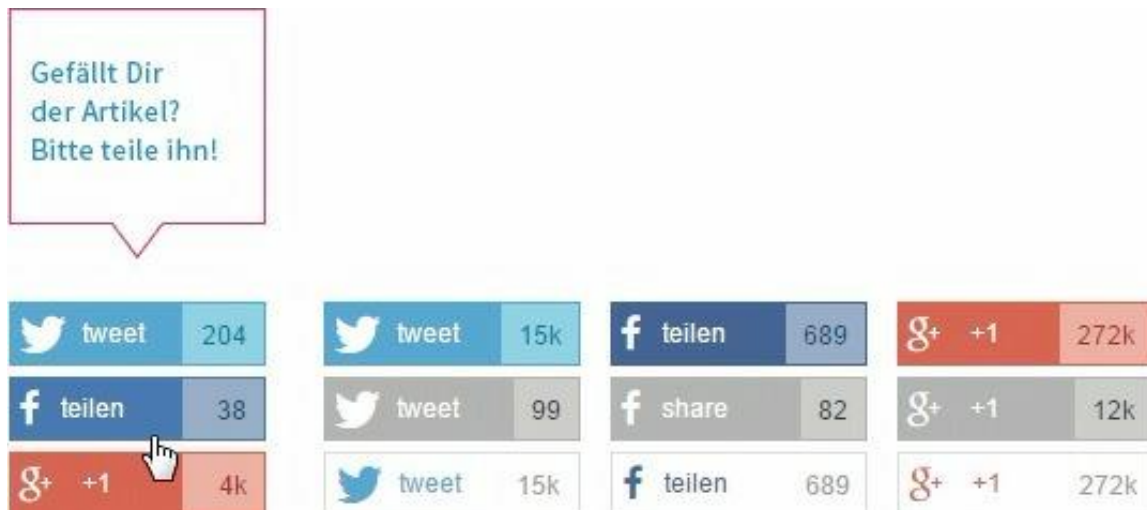
Die Social-Media-Buttons von Shariff fallen auf – und schützen die Privatsphäre des Nutzers ebenso gut wie das Zwei-Klick-Verfahren. Bei den neuen Buttons handelt es sich um einfache HTML-Links, die mit CSS individuell gestalten werden können. Eine Einbettung über iframes ist nicht nötig, weshalb auch die Aktivierung der Knöpfe entfällt. Das erspart einen Klick und das Teilen geht ein bisschen schneller.

Ein Skript ruft ab, wie oft eine Seite bereits geteilt oder getwittert wurde. Es nimmt über die Programmierschnittstellen (APIs) der Dienste zu diesen Kontakt auf und ruft die Zahlen ab. Die Abfrage geschieht also vom Server aus; statt der IP-Adresse des Besuchers wird lediglich die Server-Adresse an Facebook, Google und Twitter übertragen. Solange der Nutzer nicht auf den Link drückt, um Inhalte zu teilen, bleibt er zumindest für Facebook & Co. unsichtbar.

Thilo Weichert, Datenschutzbeauftragter des Landes Schleswig-Holstein, begrüßt unsere neuen Buttons als „datenschutzfreundlichere Technologie“. Weichert ist der Meinung, dass diese Art von Sharing-Buttons für Webseitenbetreiber obligatorisch

sein sollten, um zu verhindern, dass soziale Netzwerke das „Internetverhalten ohne aktiven Beitrag des Users mitverfolgen und speichern“.

Shariff für Webmaster



Shariff ist der Nachfolger der ebenfalls von Heise entwickelten Zwei-Klick-Variante und schützt Ihre Privatsphäre. Das Teilen geht trotzdem ganz einfach mit nur einem Klick.

Für Webmaster gibt es Shariff zum kostenlosen Download bei GitHub, siehe c't-Link am Ende des Artikels. Um Ihre eigene Website mit den datenschutzkonformen Buttons auszurüsten, laden Sie unter „Releases“ die nötigen Dateien als ZIP-Archiv herunter. Shariff besteht aus einer CSS- und einer JavaScript-Datei. Zusätzlich gibt es ein Backend in mehreren Sprachen, nämlich PHP und Perl. Auch für Node.js [2] steht eine Fassung bereit.

Für die einfachste Variante benötigen Sie nur das „Frontend“, also shariff.min.js sowie shariff.min.css. Die Buttons enthalten dann keine Angaben darüber, wie oft Inhalte bereits geteilt wurden – die holt erst das optionale Backend. Entpacken Sie den Inhalt des Archivs und laden Sie die Dateien auf Ihren Server. Die CSS-Datei binden Sie in die HTML-Dokumente im <head>-Bereich ein und die JavaScript-Datei direkt vor </body>. An allen Stellen, wo die Teilen-Funktion zu sehen sein soll, platzieren Sie einen <div>-Container mit der Klasse shariff. In der einfachsten Form sieht der Seiten Quelltext so aus:

```
<!DOCTYPE html>
<html>
<head>
  [...]
  <title>Meine feine Website</title>
```

```
<link href="shariff.min.css" rel="stylesheet">
</head>
<body>
  <h1>Mein Artikel</h1>
  <p>[...]</p>
  <div class="shariff"></div>
  <script src="shariff.min.js"></script>
</body>
</html>
```



Video: Nachgehakt

Mitteilsame Links

Einfache Links übermitteln die Inhalte an Facebook, Google+ und Twitter. Jeder Dienst nimmt dazu passende Parameter entgegen. Bei Twitter sehen diese so aus:

```
https://twitter.com/intent/tweet?text=[...]&url=[
  ...]&via=[...]
```

Der Parameter `url` übermittelt die Seitenadresse, `text` eine vorgeschlagene Botschaft, die als Tweet erscheint. Unter `via` kann man einen Twitter-Account als Quelle angeben, also etwa `@heiseonline` für Inhalte von `heise.de`. Ähnlich läuft das bei den Teilen-Links von Facebook und Google+. Dort übermitteln u beziehungsweise `url` die Seite, die man teilen möchte:

```
https://www.facebook.com/sharer/sharer.php?u=[...]
https://plus.google.com/share?url=[...]
```

Diese statischen Links lassen sich in einer Webseite ganz einfach via HTML einbinden und mit CSS individuell gestalten. Damit Sie die URL nicht bei jedem Link per Hand eintragen müssen, identifiziert Shariff die Seitenadresse mittels JavaScript und schreibt sie in den Quelltext. Als URL dient entweder der Link in der Adressleiste des Browsers oder aber – sofern vorhanden – ein Link, der im Head-Bereich des Dokuments mit `rel=canonical` festgelegt ist. Diese „kanonische URL“ kommt bei Webseiten zum Einsatz, deren Inhalte über unterschiedliche URLs aufzurufen sind. Suchmaschinen wissen so, welche URL sie bevorzugt indexieren sollen. Amazon etwa gibt in seinem Quelltext eine Google-freundliche, von Parametern bereinigte URL an.

```
uri: function() {
    var uri = global.document.location.href;
    var canonical = $('link[rel=canonical]').attr('href');
    if (canonical && canonical.length > 0) {
        if (canonical.indexOf('http') < 0) {
            canonical = global.document.location.protocol + '//'
+ global.document.location.host + canonical;
        }
        uri = canonical;
    }
    return uri;
}
```

Die Variablen `uri` und `canonical` werden mit der URL gefüllt. Die erste if-Abfrage prüft, ob `canonical` länger als null Zeichen ist – also im Quelltext überhaupt existiert. Die zweite Prüfung testet, ob die ersten Zeichen ein „http“ enthalten, was auch `https://` mit einschließt. Die Methode `indexOf()` gibt die Position einer Zeichenkette zurück; in diesem Fall das gesuchte „http“. Taucht der Wert nicht auf, spuckt die Methode den Wert `-1` aus, der kleiner als null ist. In dem Fall wird die URL aus `document.location` zusammengesetzt und als `uri` gespeichert. Das Skript liest außer der URL auch den Seitentitel aus und bettet ihn in Twitter-Tweets ein.

Schließlich schreibt Shariff den Code der Teilen-Knöpfe in die Seite. Im Quelltext ist nur `<div class="shariff"></div>` zu sehen. Wenn Sie das Element hingegen mit den Entwicklertools des Browser betrachten, sehen Sie auch den Code, den das JavaScript erzeugt hat:

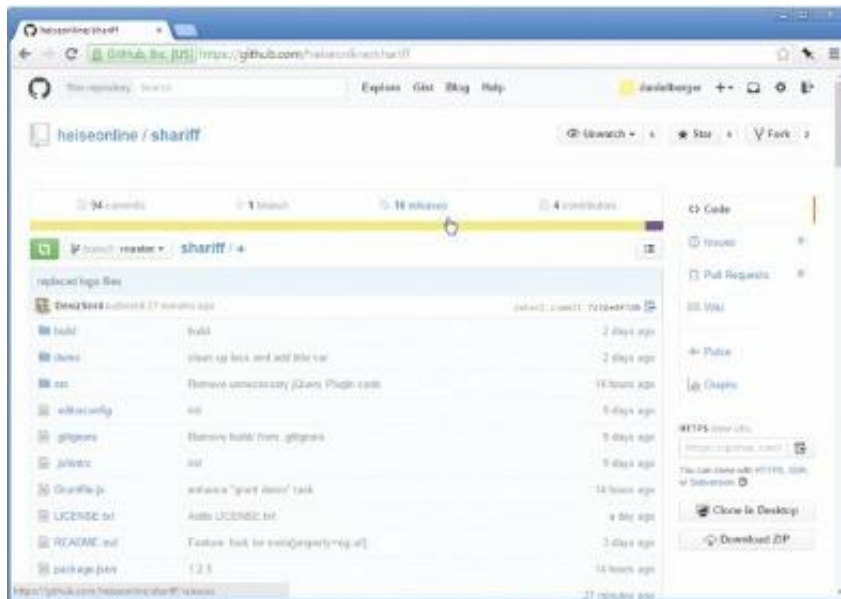
```
<ul>
  <li class="button facebook">
    <a href="https://www.facebook.com/sharer/
sharer.php?u=http://heise.de/"><span class="share
_text">teilen</span></a>
  </li>
```

```
[...]  
</ul>
```

Die Buttons sind als Listenelemente in einem ``-Container angeordnet. Gestalten lassen sie sich über `shariff.css`. Die Klassen `button` sowie `facebook`, `googleplus` und `twitter` verleihen den Teilen-Links ihr Aussehen. Ihnen kann man unterschiedliche Hintergrundfarben zuweisen, die Schriftgröße anpassen und vieles mehr. Eine graue Fassung der Buttons ist bereits eingebaut und lässt sich mit einem Daten-Attribut aktivieren: `<div class="shariff" data-theme="grey">`
`</div>`. Mit dem Attribut `data-services` können Sie außerdem festlegen, welche Buttons zu sehen sein sollen. Deren Ausrichtung steuern Sie mit `data-orientation`; der Standardwert ist `horizontal`.

Damit die Schaltflächen nicht so öde aussehen, ergänzen die Logos der Dienste den Text. Es handelt sich dabei nicht um Grafiken, sondern um flexible Vektor-Icons von Font Awesome. Die lassen sich entweder auf dem eigenen Server ablegen oder über das Content-Distribution-Network MaxCDN einbinden. Die Icons bieten den Vorteil, dass sie sich skalieren lassen, also auch auf hochauflösenden Displays knackscharf bleiben.

Gefallen finden



To Share and Protect: Shariff steht unter MIT-Lizenz auf GitHub zum kostenlosen Download bereit.

Wie viele Menschen bereits den Teilen-Button auf einer Webseite benutzt haben, ermittelt die optionale Server-Komponente. Laden Sie bei GitHub die benötigte Fassung herunter; zur Auswahl stehen Perl, PHP und Node.js. Liegt Ihre Site bei einem großen Massenhoster, ist die PHP-Fassung am einfachsten einzurichten. Der Webspace muss mindestens PHP in der Version 5.4 unterstützen, damit das Skript funktioniert. Schieben Sie den Inhalt des Ordners build auf Ihren Server, beispielsweise in ein neues Verzeichnis namens shariff-backend. Tragen Sie anschließend die URL zum Backend als Data-Attribut `data-backend-url` im div-Container ein:

```
<div class="shariff" data-backend-url="/[  
shariff-backend/"></div>
```

Bei der PHP-Version ist die Datei backend.php das Herzstück der Server-Komponente. Hinzu kommen ausgelagerte Module für Facebook, Google+ und Twitter. Dank des modularen Aufbaus lassen sich leicht weitere soziale Netzwerke hinzuzufügen. Das Backend schickt über die APIs eine Anfrage an die Server der Dienste. Diese liefern als Response je nach Anbieter mehr oder weniger Metadaten zurück – für die Teilen-Buttons ist aber nur interessant, wie oft eine URL bereits geteilt worden ist. Diese Information extrahiert das PHP-Skript aus den Daten und speichert sie für 60 Sekunden im File Cache, ehe das Skript sie erneut über die APIs abrufen. Die Angaben auf Vorrat garantieren eine schnelle Ausgabe der Zahlen, denn weitere Abfragen beantworten die sozialen Netzwerke relativ langsam.

Shariff und seine Helfer

Wie schon bei der Zwei-Klick-Lösung ermuntern wir Sie, unsere Methode auszuprobieren, an eigene Bedürfnisse anzupassen und zu verbessern. Denkbar wäre beispielsweise ein WordPress-Plug-in, welches den Einsatz in Blogs vereinfacht.

Wir haben Shariff unter der MIT-Lizenz veröffentlicht. Sie dürfen das Projekt also für private sowie kommerzielle Zwecke nutzen. Alle nötigen Dateien für den Einbau und die Weiterentwicklung finden Sie auf der GitHub-Projektseite. Dort finden Sie Shariff zum Download und eine detaillierte Anleitung für den Einbau der Perl- und Node.js-Backends in Ihre Website. Entwickler können nach Herzenslust klonen oder forken sowie eigene Commits einreichen. (dbe)

Literatur

- [Jürgen Schmidt, Zwei Klicks für mehr Datenschutz, c't 20/11, Seite 44 und <http://heise.de/-1333879>
- [Herbert Braun, Node-Lösung, JavaScript-Anwendungen für den Server mit Node.js, c't 4/14, Seite 172