



WHITEPAPER

# TRACK THE UNTRACKABLE

Wie man trotz Ad-Blocker fast 100 % der Conversions in Google Analytics messen kann



## DER HINTERGRUND

**TRACK THE UNTRACKABLE**

Seit langer Zeit besteht das Problem, dass User, die einen Ad Blocker nutzen, nicht trackbar waren.

Heute ist es wieder möglich, fast 100 % der Conversions in Google Analytics zu messen.

Die Online-Werbebranche ist ein extremer Wachstumsmarkt. Alleine für 2017 rechnet sie mit Nettowerbeinvestitionen von insgesamt 1,91 Milliarden Euro. Grundlage für eine erfolgreiche Steuerung der Online Marketing Maßnahmen ist die Erhebung und Messung von Daten. Da sind Entwicklungen, wie z. B. die für 2018 geplante E-Pri- vacy Verordnung, die „intelligente Tracking Prevention“ im neuen Safari, Opera mit integriertem Adblocker und die Zusammenarbeit von Firefox mit disconnect.me eine ernstzunehmende Gefahr, da sie das Sammeln von Daten unterbinden und damit z. B. die Erfolgsmes- sung und Steuerung erschweren.

## Schneller browsen ohne Ads

Wechsle zu Opera mit integriertem Adblocker

 Downloaden

Kostenlos für Windows

OPERA MIT INTEGRIERTEM ADBLOCKER

Davon sind auch einige Tracking-Anbieter betroffen: z. B. Google Analytics (GA). Dieser Anbieter hält sich an den „do not track“ Mo- dus der Browser. Zusätzlich gibt es viele User, die in ihrem Browser einen Adblocker installiert haben, und damit neben Werbebannern auch gleich noch die oft gescholtene Datenkrake Google blocken. Mittlerweile können im Desktop Bereich zwischen 20-25 % der Userdaten nicht mehr erfasst und aufgezeichnet werden. Bei den mobilen Geräten ist es nur eine Frage der Zeit, bis auch dort ent- sprechende Blocker-Tools zur Verfügung stehen.

## DAS ZIEL

**SEITENBETREIBER**

Aber was bedeutet das für Seitenbetreiber? Müssen die nun mit 20- 25 % Abweichung bei Tracking Tools, im speziellen Google Analytics, leben?

Die Antwort lautet: Nein! Aber: 100 % Datenerhebung bedeuten mehr Aufwand. Und man muss technische Anpassungen vorneh- men, um die User wieder ins System zu bekommen. Wichtig hierfür sind Server Side Tracking und das Measurement Protocol von GA. Damit lassen sich die „Geblockten“ in GA sichtbar machen. Ob dies der geltende Datenschutz erlaubt, muss im Einzelfall geprüft werden. An Hand eines einfachen Beispiels wird nun gezeigt, wie in Google Analytics mit Hilfe von Eventtracking die Anzahl der User, die GA und GTM blockiert haben, ermittelt werden können.

## DIE LÖSUNG

**ADBLOCKER INSTALLIERT?**

Adblocker verhindern das Tracking und die Anzeige von Werbung im Browser. Im einfachsten Fall blockieren sie dazu den Request zur Domain des Tracking-Anbieters. Will man z. B. verhindern, dass Daten an GA gesendet werden, wird ein Filter erstellt, der das Laden von Javascript Dateien mit „google-analytics.com“ in der URL sperrt.

Dadurch kann im Browser das GA JavaScript Objekt nicht mehr ini- tialisiert werden. Dies lässt sich durch die folgende JavaScript-Zeile prüfen, die false zurückgibt, wenn GA nicht verfügbar ist:

```
1 (window.ga && ga.create) ? false : 'GA'
```

Das erweiterte Skript (s. u.) prüft, ob GA und GTM vorhanden sind. Da bei Eprofessional GA im Tag Manager eingebunden wurde, prüfen wir, ob der Tag Manager geladen wurde. Falls dieser nicht lädt, wird auch GA nicht geladen. Die Information, ob GA bzw. GTM blockiert wurde, wird zusammen mit dem Document Pfad und dem Document Title an den Image Request „collect.php“ angehängt und an den Server übermittelt.

```
1 window.addEventListener('load', function ()
2 {
3     setTimeout(function () {
4         var blService = window.google_tag_manager ?
5             ((window.ga && ga.create) ? false : 'GA') :
6             "GTM+GA";
7         if (blService)
8         {
9             var title =
10                document.getElementsByTagName("title")
11                [0].innerHTML;
12                var img = document.createElement('img');
13                im.setAttribute('style',
14                'display:none;');
15                img.src = '/collect.php?ec=Blocking&ea='
16                + blService + '&dp=' +
17                encodeURIComponent(location.pathname) +
18                '&dt=' + encodeURIComponent(title);
19                document.body.appendChild(img);
20            }
21        }, 1000);
22    }, false);
```

Auf dem Server werden diese Informationen ausgelesen und mit Hilfe des Measurement Protocolls an die Google Analytics Server weitergereicht.

## DIE HERANGEHENSWEISE

**MEASUREMENT PROTOCOL**

Das Measurement Protocol erlaubt es, User-Interaktionen aus beliebigen Environments mit Hilfe von HTTP Requests an die Google Analytics Server zu schicken. Dieses nutzen wir, um unser Tracking vom Client zum Server zu verlagern.

Damit dies funktioniert, müssen beim Request definierte Parameter übergeben werden. Die in der Tabelle aufgeführten Parameter de- finieren die Mindestanzahl, die übergeben werden muss, um valide Requests zu erzeugen.

Name	Parameter	Beispiel	Beschreibung
Protocol Version	V	v=1	Die "protocol version". Der Wert ist 1.
Tracking ID	Tid	tid=UA-123456-1	Google Analytics property ID
Client ID	cid	cid=xxxx	Eindeutige User ID
Hit Type	T	t=pageview	Art der Interaktion (Pageview, Event, Transaction usw)

Beispiel-URL für einen einfachen Pageview, der bereits alle notwendigen Parameter enthält:

```
https://www.google-analytics.com/collect?v=1&-
tid=UA-123456-1&cid=5555&t=pageview&dp=%2FpageA&z=123456
```

Diese URL lässt sich in die Adresszeile eines Browsers kopieren und aufrufen. Schon hat man den ersten Request über das Measurement Protocol an GA übermittelt. Wenn man zudem sehen will, ob der Request richtige Werte übergibt, kann man die Debug URL von Google Analytics nutzen:

```
https://www.google-analytics.com/debug/collect?v=1&-
tid=UA-123456-1&cid=5555&t=pageview&dp=%2FpageA&z=123456
```

Dieser Aufruf wird nicht von GA aufgezeichnet und gibt als Ergebnis ein JSON Objekt zurück, mit Infos zu den durchgeführten Verarbeitungsschritten und zu etwaigen Fehlern beim Parsen.

```
{
  - hitParsingResult: [
    - {
      valid: true,
      parserMessage: [ ],
      hit: "/debug/collect?v=1&tid=UA-123456-1&cid=5555&t=pageview&dp=%2FpageA&z=123456"
    },
  ],
  - parserMessage: [
    - {
      messageType: "INFO",
      description: "Found 1 hit in the request."
    }
  ]
}
```

ERGEBNIS EINES DEBUG REQUESTS AN GOOGLE ANALYTICS

Eine Liste aller möglichen Parameter der Measurement Protocol findet sich hier:

<https://developers.google.com/analytics/devguides/collection/protocol/v1/parameters>

Eine kurze Erklärung mit Hinweisen und Anforderungen zur Erstellung der Request URLs findet sich hier:

<https://developers.google.com/analytics/devguides/collection/protocol/v1/reference>

Noch eine kurze Bemerkung zu dem Parameter „cid“ (Customer ID) und dem übergebenen Wert: Falls man keine eigene Customer ID einsetzt, empfehlen wir folgende php-Routine zur Festlegung der ID:

```
1 $time = time();
2 $domainName = 'eprofessional.de';
3 $domainLength = count(explode('.', $domainName));
4 $cid = rand(100000000, 999999999) . '.' . $time;
```

Dieser Wert wird dann in einem Cookie gespeichert und steht bei den nachfolgenden Requests erneut zur Verfügung.

```
1 setcookie('_ga', "GA1.{$domainLength}." . $cid,
   $time + 63115200, '/', $domainName, false, false);
```

Damit dieser Request vom Server versendet werden kann, ist eine Library notwendig, mit der man HTTP Requests ausführen kann. In diesem Beispiel wird eine PHP- mit eingebundener Curl-Library verwendet.

```
1 function ga_call($params)
2 {
3     $url =
4         isset($_REQUEST['debug'])? 'https://www.google-
5         analytics.com/debug/collect': 'https://www.google-
6         analytics.com/collect';
7
8     $ch = curl_init();
9     curl_setopt($ch, CURLOPT_URL, $url);
10    curl_setopt($ch, CURLOPT_USERAGENT,
11    isset($_SERVER['HTTP_USER_AGENT']) ?
12    $_SERVER['HTTP_USER_AGENT'] : '');
13    curl_setopt($ch, CURLOPT_HTTPHEADER,
14    array('Content-type: application/x-www-form-
15    urlencoded'));
16    curl_setopt($ch, CURLOPT_HTTP_VERSION,
17    CURL_HTTP_VERSION_1_1);
18    curl_setopt($ch, CURLOPT_POST, TRUE);
19    curl_setopt($ch, CURLOPT_POSTFIELDS,
20    utf8_encode(http_build_query($params, null,
21    ini_get('arg_separator.output'),
22    PHP_QUERY_RFC3986)));
23    $content = curl_exec($ch);
24    curl_close($ch);
25    return $content;
26 }
```

Mit diesem Request werden Parameter versendet, deren Werte teilweise über die Server Variablen und durch den Image Aufruf im Browser des Clients definiert sind.

```
1 $tid = 'UA-40636790-4';
2 $server_name = isset($_SERVER['SERVER_NAME']) ?
3 strtolower($_SERVER['SERVER_NAME']) : '';
4 $server_protocol = isset($_SERVER['REQUEST_SCHEME'])
5 ? strtolower($_SERVER['REQUEST_SCHEME']) : '';
6 if (substr($server_name, 0, 4) == 'www.')
7 {
8     $domain = substr($server_name, 4);
9 }
10 $paras = array(
11     'v' => 1, //Protocol Version
12     'tid' => $tid, //Tracking Id
13     'aip' => 1, //Anonymize IP
14     'ds' => 'web', // Datasource (optional)
15     'cid' => get_cid($domain), //client ID
16     'uip' => get_ip(), // IP Override
17     'ua' => isset($_SERVER['HTTP_USER_AGENT']) ?
18     $_SERVER['HTTP_USER_AGENT'] : '', // User Agent
19     'dr' => isset($_SERVER['HTTP_REFERER']) ?
20     $_SERVER['HTTP_REFERER'] : '', //Document
21     Referrer
22     't' => 'event', //'pageview', 'screenview',
23     'event', 'transaction', 'item', 'social',
24     'exception', 'timing'.
25     'ni' => '0', //Non-Interaction Hit
26     'dl' => $server_protocol . "://" . $server_name
27     . $dp, //Document location URL
28     'dh' => $server_name, //Document Host Name
29     'dp' => isset($_GET['dp']) ? $_GET['dp'] : '',
30     //Document Path
31     'dt' => isset($_GET['dt']) ? $_GET['dt'] : '',
32     //Document Title
33     'ec' => isset($_GET['ec']) ? $_GET['ec'] : '',
34     //Event Category
35     'ea' => isset($_GET['ea']) ? $_GET['ea'] : '',
36     //Event Action
37     'el' => isset($_GET['el']) ? $_GET['el'] : '',
38     //Event Label
39     'ev' => isset($_GET['ev']) ? $_GET['ev'] : 0,
40     //Event Value
41     'z' => rand(1000000000, 2147483647) // Cache
42     Buster
43 );
```

Im Anschluss werden die Parameter an GA gesendet und, falls man nicht im Debug-Modus ist, gibt das Skript ein einfaches Bild zurück.

```
1  $result = ga_call($paras);
2
3  if (isset($_REQUEST['debug']))
4  {
5
6      print $result;
7  }
8  else
9  {
10     header('Content-Type: image/png');
11     echo
12     base64_decode('R0lGODlhAQABAIAAAAUeBAAACwAAAAAQABAAACakQBADs');
13 }
```

#### AUSBLICK

### NAHEZU 100 % ALLER CONVERSIONS IN GOOGLE ANALYTICS?

Bei diesem Beispiel handelt es sich um eine leicht verständliche Variante. Es ist möglich diese Technologie zu nutzen, um z. B. beim Ecommerce-Tracking alle Conversions wieder in GA sichtbar zu machen. Notwendig ist in diesem Fall dann jedoch eine valide Kanalzuordnung, die sich auf der betreffenden Seite implementieren lässt. Mit der JavaScript Bibliothek Sourcebuster lassen sich viele Informationen, so auch die Kanalzuordnung, in einem Cookie ablegen und dann bei der Transaktion erneut auslesen.

#### ÜBER EPROFESSIONAL

### WE ARE DIGITAL EXPERTS

Als eine der erfahrensten Digital Marketing Agenturen in Deutschland mit mehr als 130 Experten am Standort Hamburg, gehören wir seit 1999 zu den Top-Adressen für Digitales Marketing.

Eprofessional ist Partner starker Marken, führender Web Portale, performanter Web Shops und erfolgreicher Multifilialisten in Europa. Wir stehen für kanalübergreifendes Online Marketing und beste Performance.

Möchten Sie gern mehr über unsere Leistungsbereiche erfahren, oder sind Sie an einer Zusammenarbeit interessiert?

Dann freuen wir uns über Ihre Nachricht.