A Benchmark for HTTP 2.0 Header Compression

Christian Grothoff

July 13, 2013

Abstract

The ongoing development of the HTTP 2.0 standard requires decisions on the compression algorithm that is to be used to compress HTTP headers in HTTP 2.0. This paper presents a public benchmark consisting of traces of HTTP headers based on real-world data with the goal of supporting discussions on HTTP 2.0 compression standards with realistic data. We present how the data was collected and preprocessed for publication and describe some of the characteristics of the resulting benchmark.

1 Introduction

IETF's "HTTPBIS" working group is currently developing the standard for HTTP 2.0 based on Google's SPDY protocol. The goals for HTTP 2.0 are reduced latency and bandwidth consumption and better security compared to HTTP 1.1. The primary proposed method for bandwidth reduction is the use of HTTP header compression (bodys can already be compressed in HTTP 1.0). Currently, the compression algorithm has not yet been finalized.

Another method that will most likely be used for reducing latency and bandwidth consumption in HTTP 2.0 is multiplexing multiple requests and interleaving multiple responses over the same TCP connection. Improving on HTTP 1.1 request pipelining, this will enable HTTP 2.0 to use a single TCP connection for many requests, which should reduce the number of parallel TCP connections created by browsers and thus minimize TCP setup costs. Furthermore, if many requests use the same TCP connection, state can be transmitted once per connection, avoiding the costly re-transmissions of client information in the "stateless" HTTP 1.1 sessions.

Given these two key features of HTTP 2.0, a benchmark for HTTP header compression needs to not only include individual, isolate HTTP headers but traces of HTTP headers. A *trace* is a series of HTTP header requests that are made from the same HTTP client to the same HTTP server within a limited amount of time. While in HTTP 1.1, multiple TCP streams (with different source ports) might be used to transmit a trace of HTTP headers, these requests would all be transmitted via the same TCP connection in HTTP 2.0. As a result, the compression algorithm can benefit from this situation, for example by only transmitting the differences between the headers for each request. The

benchmark presented in this paper thus provides traces of HTTP headers instead of just individual HTTP headers.

The goal of this work was to create a realistic benchmark for HTTP 2.0 header compression. As HTTP 2.0 is likely to be widely used, providing a purely synthetic benchmark for the compression algorithm is unlikely to adequately reflect the complex realities that HTTP 2.0 will experience in practice. A benchmark that can guide developers to an algorithm that in practice performs only a tiny fraction better would likely result in dramatic savings on a global scale over the lifetime of HTTP 2.0. Thus, the benchmark presented in this paper is based on real-world data collected from tens of thousands of actual users (Section 2).

However, publishing the raw data is not possible, as observing actual users HTTP traffic — even just the meta data from the HTTP headers — creates privacy issues.[1] Thus, prior to making such data public, it needs to be cleaned to remove the personally identifiable information. This paper shows that we can achieve this without destroing the utility of the data as a representative benchmark (Section 3).

As a result of this process, we obtained five data sets with traces containing approximately one million HTTP requests or responses each. The data is[2] publicly available from our website and can be freely used by anyone as long as they properly acknowledge the source of the data (by citing this paper). The website also contains the source code used to generate, clean and analyze the data.

2 Data Collection

We collected the HTTP trace data using `libpcap` on a high speed network link of a large ISP with tens of thousands of users. Only IPv4 TCP traffic involving port 80 was captured. Instead of reconstructing TCP streams, we assume that HTTP requests and responses (almost) always start at the beginning of an IP packet and thus simply checked if a given IP packed contained a valid HTTP request or response header. The IP addresses and the TCP payload of those packets that matched our HTTP request/response signature were then written to a database, together with a timestamp (in milliseconds). Using this method, our capture process was able to process packets quickly enough to drop less than one per million packets. As a result, we are optimistic that most collected HTTP traces are complete.

For further processing, the raw traces were then partitioned (by timestamp) into groups of one million HTTP requests or responses each.

[1] I disagree with GCHQ and NSA on this matter.

[2] Or rather, will be, as this is a draft report.

3 Data Cleaning

The most sensitive raw data are the collected source and destination IP addresses. Thus, in the first processing step, we ordered all traces by their respective source and destination IP address (and timestamp). All requests that have the same source and destination address and timestamps less than 5 minutes apart are then assigned the same trace identifier, which is generated by incrementing a global trace counter. Source and IP addresses are then discarded and HTTP headers are simply stored under the new trace identifier instead.

The goal for the next step of data cleaning is to remove all personally identifiable data from the HTTP headers — without also destroying characteristics that might be significant for compression. In particular, it is common among headers to repeat certain values consistently between requests, such as the host name, user agent or cookies. Also, the host name often occurs in several areas of the HTTP header. However, these values may also include sensitive information and thus must be obscured.

The basic method chosen for the obfuscation is simplistic substitution. However, as we want to leave compression unaffected by the substitution, we generally substitute within character groups. Specifically, numbers are substituted only by numbers, hexadecimal values only by hexadecimal values, letters only by letters, and the capitalization / non-capitalization of letters is also preserved. We distinguish between two types of substitutions. During consistent substitution, we always apply the same substitution map for all headers within a trace. While we want to apply a substitution cipher to the "cookie" headers, we do want to preserve the fact that cookies tend to remain the same within a given trace. Thus, the same substitution is applied to the "cookie" values for the entire trace. Consistent substitution is also applied to URIs; however, using a different substitution map. Similarly, for the "user-agent" we want to obscure the browser version; this is done by applying a substitution cipher on the version numbers — and again substitutions are done consistently within the same trace.

In contrast, inconsistent substitutions are employed when no similarities between multiple requests within the same trace are expected; for example, the "content-length" header is obscured by replacing the number with a different number of the same length with no consistency among the different requests within the same trace.

Headers that are known to include the hostname (from "host") are all processed specially. Our method first applies a (consistent, within a trace) substitution on the host name itself — ensuring that say "example.com" always shows up as "ztkxwoz.mqx" in all of these headers. The remainder of the headers containing the hostname are also modified using the same substitution.

Going over the collected HTTP headers by hand, we found over 150 headers that needed randomization of the included data, including many custom headers that included, for example, hash codes of delivered content or (numeric) customer user identifiers. It was decided to keep time stamps ("date", "expires", "last-modified") without modifications as the proximity of dates in the headers

Table 1: Typical regular expressions for headers and their allowed values. Note that "pramga" is misspelled in the headers; various standard headers are misspelled on occasion in our data set.

Header	Regular expression
geoip-country-code	"[̂A-Z][A-Z]$"
pramga	"n̂o-cache$"
rating	"R̂TA-5042-1996-1400-1577-RTA$"
retry-after	"0̂$"
ua-cpu	"ÂMD64—ARM—x86$"
varnishset	"ŷes—nocc$"
x-width	"[̂0-9]*$"

might be a particularly suitable target for compression algorithms, and as the specific dates of the collection do not seem to be privacy-sensitive.

For the remaining headers that were determined to be non-critical and thus were not transformed, regular expressions were written to ensure that only headers that matched the expected content were included in the benchmark. By writing a regular expression, we first of all ensure that the headers that were manually inspected actually reflect the breadth of the collected data, and furthermore (typically) limit the set of possible accepted values to a finite set that is thus not characteristic for some individual user. Table 1 gives some characteristic examples for the regular expressions used by our filter. Note that we are using case-insensitive POSIX regular expression matching where "^" stands for the beginning and "$" for the end of a line. Tailing whitespace is stripped from the values prior to matching with the given regular expressions.

By manually inspecting values that did not match the specified regular expression for a header, we found a few instances where possibly private information was unexpectedly present in these headers. For example, our regular expression for the "From" header is

```
^googlebot\\(at\\)googlebot\\.com|
msnbot\\(at\\)microsoft\\.com|
support@search\\.yandex\\.ru|
bingot\\(at\\)microsoft\\.com|
webcontent@bloomberg\\.net|
284444-web5|
crawler@exabot\\.com|
ggportal to imstorage-user-api|
psbot@picsearch\\.com|
webmaster@page2rss\\.com|
n7abcmed01|
n7abcmwc0[0-9]\\.starwave\\.com$
```

Table 2: Key characteristics of the public benchmark data. Note that the first line of the HTTP request or response is included in the number of bytes of header data, but not in the other reported statistics.

Characteristic	Avg. value ± std. dev.
# headers	999,961 ± 18
# traces	41,103 ± 9,324
# trace length	25 ± 6
# header values (total)	7,981,058 ± 449,429
# different types of headers	390 ± 27
# bytes of header data	426 ± 54 MB

However, looking at the millions of HTTP headers, we found a handful of "From" headers where users were seemingly (still) sending their private e-mail addresses. Similarly, we found a caching header that typically was set to "HIT" or "MISS" but occasionally also included the destination of the original request. Categorically preserving those headers would have thus have leaked information that we intended to mask using the substitution algorithms.

All header values that were neither subjected to character substitution nor matched the given regular expression were discarded and not included in the public benchmark. However, we report statistical information on the discarded data in Section 4.

4 A First Look at the Data

The public benchmark consists of five sets of approximately one million HTTP headers each. The number is slightly lower than one million as some malformed HTTP headers (likely packets that were not actually HTTP headers in the first place are detected and filtered during data cleaning). Splitting the dataset into five databases was done to allow users of the benchmark to report statistical properties of the data sets or to use some sets for training and others for evaluation. Furthermore, this also keeps the download size for the individual files more reasonable. Table 2 summarizes key characteristics of the public benchmark, whereas Table 3 gives statistics on the information lost during data cleaning. The tables show that only a tiny fraction of the overall header data was removed during data cleaning; the main impact of the process is the number of different types of headers left in each of the data sets is reduced by a factor of more than three: from close to 1,500 different headers in the original data to at most 427.

One important question for HTTP 2.0 is how many requests would typically be seen per connection. Figure 1 shows a histogram listing the number of HTTP requests per trace included in the benchmark. Note that our definition of trace requires a request from the same source to the same destination within

a five minute time interval. If HTTP 2.0 sessions would time out before then, the expected trace lengths would be shorter. As the public benchmark includes timestamps, the available data will allow the community to compute trace length distributions for various timeout values (but only up to five minutes).

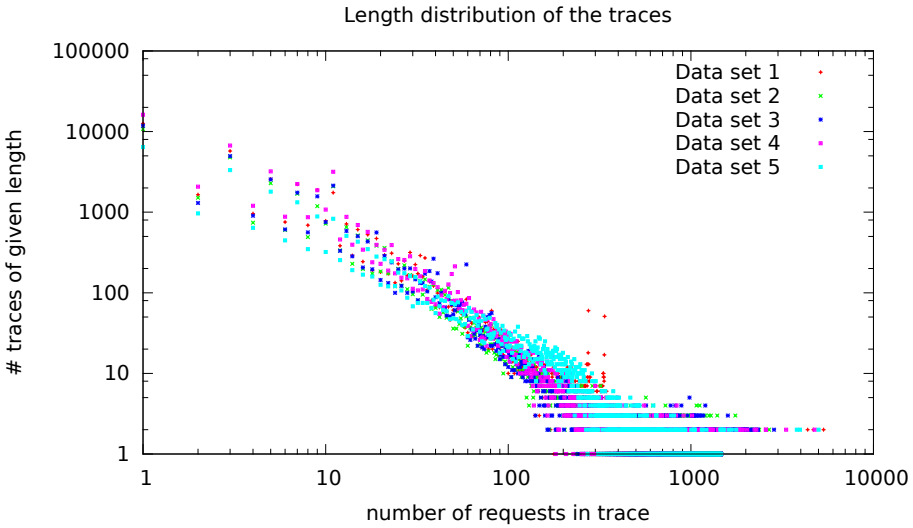

Figure 1: Length distribution for the traces. The length of a trace is defined as the number of headers in a given trace.

Another interesting question is how long HTTP headers are in general to begin with. Figure 2 shows the distribution of header lengths in bytes, and

Table 3: Amount of information removed during cleaning. We distinguish between headers that were dropped because our filter had no custom logic for the header (for example, we dropped "MS-Author-Via" and other uncommon headers for this reason) and headers where the value did not fit our regular expression (for example, we dropped "X-Cacheable: NO:Not Cacheable" for this reason). When counting the number of bytes removed, we include the string for the header type, the value and four bytes for colon, space and LFCR.

Characteristic	Value ± std. dev.
# header values suppressed	40,367 ± 9,128
# header types dropped (unknown)	958 ± 114
# header types dropped (regex missmatch)	24,4 ± 5,5
# bytes of header data removed	1,612 ± 42 KB

Figure 3 shows the number of key-value pairs found in the headers of the public benchmark.

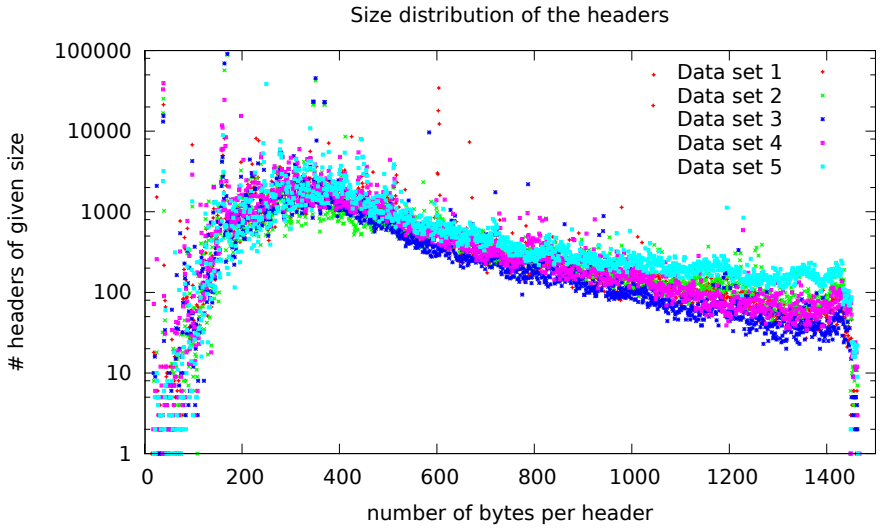

Figure 2: Number of bytes per header. Note that our collection method captures IP packets of at most 1500 bytes and thus HTTP headers that exceed 1400 bytes may not be fully captured; the exact threshold depends on the presence of IP and TCP options. This explains both the slight peak after 1400 bytes and the sharp drop shortly thereafter.

Finally, we present the results from using three compression algorithms on our traces. Each compression algorithm was initialized once per trace and given all of the header data from all of the requests in the trace. Headers for requests and responses were compressed separately. After each full header the respective algorithm had to "flush" its buffers, thus enabling the decoder to decode each header without seeing compressed data generated from the next header. Table 4 summarizes the results of this benchmark for the publicly available data. Table 5 summarizes the results for the original, non-public data prior to the data cleaning operations described in Section 3. It should be noted that in particular the compression ratios for the original data[3] are within less than one std. dev. (over the five subsets) of the results for the cleaned data for gzip. Consequently, it is reasonable to assume that the characteristics of the benchmark for compression are preserved by the cleaning process. The full calculations and scripts to gather the data are available at `https://gnunet.org/httpbenchmark/` together with the public benchmark.

[3]gzip achieves about 14% for request- and 18% for response-headers for the original data.

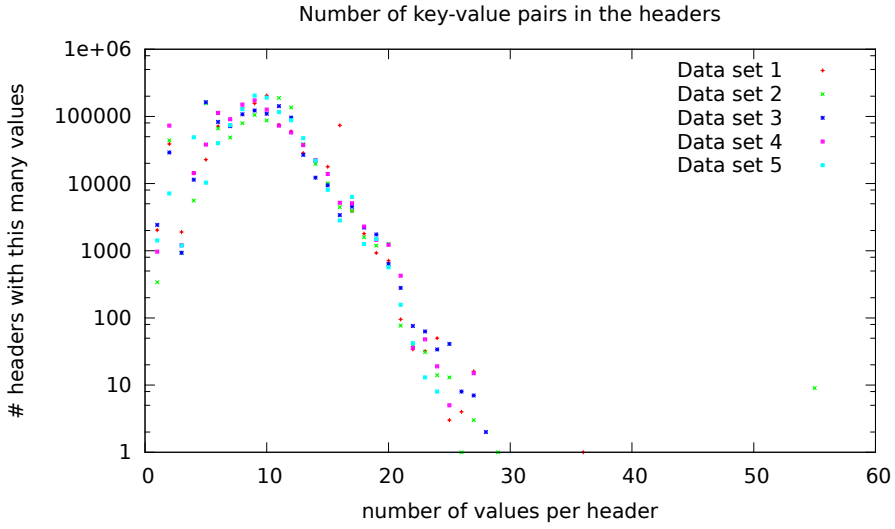

Figure 3: Number of key-value pairs in the headers.

5 Related Work

To the best of our knowledge, this is the first publicly available benchmark based on HTTP headers collected from real-world users. We are also not aware of any prior work on how to anonymize HTTP header data. The closest existing work on ensuring user privacy is the traditional best practices to minimize the collected data (which we do by eliminating IP addresses and HTTP payload). The research on k-anonymity, which tries to ensure that data cannot be linked to a group smaller than k individuals, is not really applicable as after removing IP addresses there is no way to linked any of the collected records back to an

Table 4: Performance of traditional compression algorithms for HTTP headers from the public benchmark for requests (req.) and responses (res.).

Compression algorithm	Compressed size	Comp. time	Decomp. time
`memcpy` req.	260 ± 44 MB	166 ± 6 ms	148 ± 3 ms
gzip req.	46 ± 18 MB	9,3 ± 2,2s	2 ± 0,5 s
bzip2 req.	215 ± 33 MB	103 ± 14 s	26 ± 4 s
`memcpy` res.	157 ± 12 MB	155 ± 5 ms	137 ± 3 ms
gzip res.	30 ± 4 MB	7,1 ± 0,7s	1,4 ± 0,1 s
bzip2 res.	138 ± 10 MB	70 ± 5 s	17 ± 1,2 s

individual without additional external data. Thus, our data cleaning focuses on making it difficult for an adversary with extensive access to external information to link our records back to individuals. In contrast, work on k-anonymity generally excludes the possibility of an adversary having additional information external to the data set that is being anonymized.

6 Conclusion

Removing personally identifiable data from traces that may contain arbitrary fields a difficult process, as it is not always clear which data might be personally identifiable to begin with. Formulating regular expressions can reduce the risk of accidentally including personal information.

Acknowledgements

I thank Georg Carle for helpful discussions on the research and Lothar Braun for his technical support.

A Sample headers

This is the output of "SELECT header FROM public WHERE trace=42" on the first benchmark set. Note that a typical trace contains both directions of the traffic (if they could be observed).

```
GET /fer.js HTTP/1.1
Cookie: ca_tyrv_izaq=f672a3d41d7c93ad33bb76153c8cc05e
Connection: Keep-Alive
Host: apaqvzbh.tdxvomdah.de
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/9.0 (compatible; MSIE 8.0; Windows NT 6.2; Trident/9.0)
Accept-Language: de-DE
```

Table 5: Performance of traditional compression algorithms for HTTP headers from the non-public benchmark for requests (req.) and responses (res.).

Compression algorithm	Compressed size	Comp. time	Decomp. time
`memcpy` req.	265 ± 45 MB	169 ± 4 ms	147 ± 3 ms
gzip req.	37 ± 6 MB	8,6 ± 1,4s	2,1 ± 0,2 s
bzip2 req.	210 ± 31 MB	103 ± 15 s	26 ± 3,9 s
`memcpy` res.	163 ± 12 MB	159 ± 3 ms	138 ± 3 ms
gzip res.	29 ± 4 MB	7,4 ± 0,7s	1,6 ± 0,1 s
bzip2 res.	138 ± 10 MB	71 ± 4,6s	17 ± 1,1 s

```
Referer: http://tnmeiawal.ea/dawmbttcgmssagwfpn.4.html
Accept: application/javascript, */*;q=0.8

HTTP/1.1 200 OK
Date: Thu, 18 Apr 2013 20:48:28 GMT
Server: Microsoft-IIS/6.0
ETag: "0483cd9c9ae47:975"
Accept-Ranges: bytes
Last-Modified: Thu, 09 Oct 2003 17:45:00 GMT
Content-Type: application/x-javascript
Content-Length: 10

GET /feot.zgz?l=898291034&xgfn=uqla:78&dsqcycfpzfihl=7&tqmwca=UfwnlawLnmeimpJlbqdqr&arcsmea=
%9Ddk.23989162%6Ce.Yms HTTP/1.1
Cookie: ca_tyrv_izaq=f672a3d41d7c93ad33bb76153c8cc05e
Connection: Keep-Alive
Host: imvxxdkn.kkwbrcgan.de
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/9.0 (compatible; MSIE 8.0; Windows NT 6.2; Trident/9.0)
Accept-Language: de-DE
Referer: http://tnmeiawal.ea/dawmbttcgmssagwfpn.4.html
Accept: application/javascript, */*;q=0.8

HTTP/1.1 200 OK
Content-Type: application/x-javascript
Date: Thu, 18 Apr 2013 20:48:28 GMT
Cache-Control: private, max-age=0, no-cache
Pragma: no-cache
X-Powered-By: Zend Core/5.9.5 PHP/9.5.6
Server: Microsoft-IIS/6.0
Date: Thu, 18 Apr 2013 20:48:28 GMT
Connection: close

GET /fesqh.zgz?dfllawie=7331&csialnie=6722&uqlaie=78&tqmwca=UfwnlawLnmeimpJlbqdqr&dsqcy=4&cf
Cookie: ca_tyrv_izaq=f672a3d41d7c93ad33bb76153c8cc05e
Connection: Keep-Alive
Host: imvxxdkn.kkwbrcgan.de
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/9.0 (compatible; MSIE 8.0; Windows NT 6.2; Trident/9.0)
Accept-Language: de-DE
Referer: http://tnmeiawal.ea/dawmbttcgmssagwfpn.4.html
Accept: image/png, image/svg+xml, image/*;q=0.8, */*;q=0.5

HTTP/1.1 200 OK
Content-Type: image/gif
Content-Length: 05
```

```
Date: Thu, 18 Apr 2013 20:48:28 GMT
Cache-Control: private, max-age=0, no-cache
Pragma: no-cache
X-Powered-By: Zend Core/5.9.5 PHP/9.5.6
Server: Microsoft-IIS/6.0
Date: Thu, 18 Apr 2013 20:48:28 GMT

GET /feot.zgz?l=414441207&xgfn=uqla:78&dsqcycfpzfihl=7&tqmwca=UfwnlawLnmeimpJlbqdqr&arcsmea=
Cookie: ca_tyrv_izaq=f672a3d41d7c93ad33bb76153c8cc05e
Connection: Keep-Alive
Host: imvxxdkn.kkwbrcgan.de
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/9.0 (compatible; MSIE 8.0; Windows NT 6.2; Trident/9.0)
Accept-Language: de-DE
Referer: http://tnmeiawal.ea/dawmbttcgmssagwfpn.4.html
Accept: application/javascript, */*;q=0.8

HTTP/1.1 200 OK
Content-Type: application/x-javascript
Date: Thu, 18 Apr 2013 20:48:28 GMT
Cache-Control: private, max-age=0, no-cache
Pragma: no-cache
X-Powered-By: Zend Core/5.9.5 PHP/9.5.6
Server: Microsoft-IIS/6.0
Date: Thu, 18 Apr 2013 20:48:28 GMT
Connection: close

GET /fesqh.zgz?dfllawie=7069&csialnie=6669&uqlaie=78&tqmwca=UfwnlawLnmeimpJlbqdqr&dsqcy=4&cf
Cookie: ca_tyrv_izaq=f672a3d41d7c93ad33bb76153c8cc05e
Connection: Keep-Alive
Host: imvxxdkn.kkwbrcgan.de
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/9.0 (compatible; MSIE 8.0; Windows NT 6.2; Trident/9.0)
Accept-Language: de-DE
Referer: http://tnmeiawal.ea/dawmbttcgmssagwfpn.4.html
Accept: image/png, image/svg+xml, image/*;q=0.8, */*;q=0.5

HTTP/1.1 200 OK
Content-Type: image/gif
Content-Length: 14
Date: Thu, 18 Apr 2013 20:48:28 GMT
Cache-Control: private, max-age=0, no-cache
Pragma: no-cache
X-Powered-By: Zend Core/5.9.5 PHP/9.5.6
Server: Microsoft-IIS/6.0
Date: Thu, 18 Apr 2013 20:48:28 GMT
```

```
GET /feot.zgz?l=150619908&xgfn=uqla:78&dsqcycfpzfihl=7&tqmwca=UfwnlawLnmeimpJlbqdqr&arcsmea=
Cookie: ca_tyrv_izaq=f672a3d41d7c93ad33bb76153c8cc05e
Connection: Keep-Alive
Host: imvxxdkn.kkwbrcgan.de
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/9.0 (compatible; MSIE 8.0; Windows NT 6.2; Trident/9.0)
Accept-Language: de-DE
Referer: http://tnmeiawal.ea/dawmbttcgmssagwfpn.4.html
Accept: application/javascript, */*;q=0.8

HTTP/1.1 200 OK
Content-Type: application/x-javascript
Date: Thu, 18 Apr 2013 20:48:28 GMT
Cache-Control: private, max-age=0, no-cache
Pragma: no-cache
X-Powered-By: Zend Core/5.9.5 PHP/9.5.6
Server: Microsoft-IIS/6.0
Date: Thu, 18 Apr 2013 20:48:29 GMT
Connection: close

GET /fesqh.zgz?dfllawie=7020&csialnie=6635&uqlaie=78&tqmwca=UfwnlawLnmeimpJlbqdqr&dsqcy=4&cf
Cookie: ca_tyrv_izaq=f672a3d41d7c93ad33bb76153c8cc05e
Connection: Keep-Alive
Host: imvxxdkn.kkwbrcgan.de
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/9.0 (compatible; MSIE 8.0; Windows NT 6.2; Trident/9.0)
Accept-Language: de-DE
Referer: http://tnmeiawal.ea/dawmbttcgmssagwfpn.4.html
Accept: image/png, image/svg+xml, image/*;q=0.8, */*;q=0.5

HTTP/1.1 200 OK
Content-Type: image/gif
Content-Length: 19
Date: Thu, 18 Apr 2013 20:48:29 GMT
Cache-Control: private, max-age=0, no-cache
Pragma: no-cache
X-Powered-By: Zend Core/5.9.5 PHP/9.5.6
Server: Microsoft-IIS/6.0
Date: Thu, 18 Apr 2013 20:48:29 GMT

GET /feot.zgz?l=536765318&xgfn=uqla:78&dsqcycfpzfihl=7&tqmwca=UfwnlawLnmeimpJlbqdqr&arcsmea=
Cookie: ca_tyrv_izaq=f672a3d41d7c93ad33bb76153c8cc05e
Connection: Keep-Alive
Host: imvxxdkn.kkwbrcgan.de
Accept-Encoding: gzip, deflate
```

```
User-Agent: Mozilla/9.0 (compatible; MSIE 8.0; Windows NT 6.2; Trident/9.0)
Accept-Language: de-DE
Referer: http://tnmeiawal.ea/dawmbttcgmssagwfpn.4.html
Accept: application/javascript, */*;q=0.8

HTTP/1.1 200 OK
Content-Type: application/x-javascript
Date: Thu, 18 Apr 2013 20:48:29 GMT
Cache-Control: private, max-age=0, no-cache
Pragma: no-cache
X-Powered-By: Zend Core/5.9.5 PHP/9.5.6
Server: Microsoft-IIS/6.0
Date: Thu, 18 Apr 2013 20:48:29 GMT
Connection: close

GET /fesqh.zgz?dfllawie=7062&csialnie=6662&uqlaie=78&tqmwca=UfwnlawLnmeimpJlbqdqr&dsqcy=4&cf
Cookie: ca_tyrv_izaq=f672a3d41d7c93ad33bb76153c8cc05e
Connection: Keep-Alive
Host: imvxxdkn.kkwbrcgan.de
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/9.0 (compatible; MSIE 8.0; Windows NT 6.2; Trident/9.0)
Accept-Language: de-DE
Referer: http://tnmeiawal.ea/dawmbttcgmssagwfpn.4.html
Accept: image/png, image/svg+xml, image/*;q=0.8, */*;q=0.5

HTTP/1.1 200 OK
Content-Type: image/gif
Content-Length: 91
Date: Thu, 18 Apr 2013 20:48:29 GMT
Cache-Control: private, max-age=0, no-cache
Pragma: no-cache
X-Powered-By: Zend Core/5.9.5 PHP/9.5.6
Server: Microsoft-IIS/6.0
Date: Thu, 18 Apr 2013 20:48:29 GMT

GET /feot.zgz?l=578772525&xgfn=uqla:78&dsqcycfpzfihl=7&tqmwca=UfwnlawLnmeimpJlbqdqr&arcsmea=
Cookie: ca_tyrv_izaq=f672a3d41d7c93ad33bb76153c8cc05e
Connection: Keep-Alive
Host: imvxxdkn.kkwbrcgan.de
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/9.0 (compatible; MSIE 8.0; Windows NT 6.2; Trident/9.0)
Accept-Language: de-DE
Referer: http://tnmeiawal.ea/dawmbttcgmssagwfpn.4.html
Accept: application/javascript, */*;q=0.8

HTTP/1.1 200 OK
```

```
Content-Type: application/x-javascript
Date: Thu, 18 Apr 2013 20:48:29 GMT
Cache-Control: private, max-age=0, no-cache
Pragma: no-cache
X-Powered-By: Zend Core/5.9.5 PHP/9.5.6
Server: Microsoft-IIS/6.0
Date: Thu, 18 Apr 2013 20:48:29 GMT
Connection: close

GET /fesqh.zgz?dfllawie=7095&csialnie=6620&uqlaie=78&tqmwca=UfwnlawLnmeimpJlbqdqr&dsqcy=4&cf
Cookie: ca_tyrv_izaq=f672a3d41d7c93ad33bb76153c8cc05e
Connection: Keep-Alive
Host: imvxxdkn.kkwbrcgan.de
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/9.0 (compatible; MSIE 8.0; Windows NT 6.2; Trident/9.0)
Accept-Language: de-DE
Referer: http://tnmeiawal.ea/dawmbttcgmssagwfpn.4.html
Accept: image/png, image/svg+xml, image/*;q=0.8, */*;q=0.5

HTTP/1.1 200 OK
Content-Type: image/gif
Content-Length: 79
Date: Thu, 18 Apr 2013 20:48:29 GMT
Cache-Control: private, max-age=0, no-cache
Pragma: no-cache
X-Powered-By: Zend Core/5.9.5 PHP/9.5.6
Server: Microsoft-IIS/6.0
Date: Thu, 18 Apr 2013 20:48:29 GMT

GET /feot.zgz?l=524030103&xgfn=uqla:78&dsqcycfpzfihl=7&tqmwca=UfwnlawLnmeimpJlbqdqr&arcsmea=
Cookie: ca_tyrv_izaq=f672a3d41d7c93ad33bb76153c8cc05e
Connection: Keep-Alive
Host: imvxxdkn.kkwbrcgan.de
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/9.0 (compatible; MSIE 8.0; Windows NT 6.2; Trident/9.0)
Accept-Language: de-DE
Referer: http://tnmeiawal.ea/dawmbttcgmssagwfpn.4.html
Accept: application/javascript, */*;q=0.8

HTTP/1.1 200 OK
Content-Type: application/x-javascript
Date: Thu, 18 Apr 2013 20:48:29 GMT
Cache-Control: private, max-age=0, no-cache
Pragma: no-cache
X-Powered-By: Zend Core/5.9.5 PHP/9.5.6
Server: Microsoft-IIS/6.0
```

```
Date: Thu, 18 Apr 2013 20:48:29 GMT
Connection: close

GET /fesqh.zgz?dfllawie=7362&csialnie=6743&uqlaie=78&tqmwca=UfwnlawLnmeimpJlbqdqr&dsqcy=4&cf
Cookie: ca_tyrv_izaq=f672a3d41d7c93ad33bb76153c8cc05e
Connection: Keep-Alive
Host: imvxxdkn.kkwbrcgan.de
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/9.0 (compatible; MSIE 8.0; Windows NT 6.2; Trident/9.0)
Accept-Language: de-DE
Referer: http://tnmeiawal.ea/dawmbttcgmssagwfpn.4.html
Accept: image/png, image/svg+xml, image/*;q=0.8, */*;q=0.5

HTTP/1.1 200 OK
Content-Type: image/gif
Content-Length: 79
Date: Thu, 18 Apr 2013 20:48:29 GMT
Cache-Control: private, max-age=0, no-cache
Pragma: no-cache
X-Powered-By: Zend Core/5.9.5 PHP/9.5.6
Server: Microsoft-IIS/6.0
Date: Thu, 18 Apr 2013 20:48:29 GMT

GET /feot.zgz?l=243466491&xgfn=uqla:78&dsqcycfpzfihl=7&tqmwca=UfwnlawLnmeimpJlbqdqr&arcsmea=
Cookie: ca_tyrv_izaq=f672a3d41d7c93ad33bb76153c8cc05e
Connection: Keep-Alive
Host: imvxxdkn.kkwbrcgan.de
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/9.0 (compatible; MSIE 8.0; Windows NT 6.2; Trident/9.0)
Accept-Language: de-DE
Referer: http://tnmeiawal.ea/dawmbttcgmssagwfpn.4.html
Accept: application/javascript, */*;q=0.8

HTTP/1.1 200 OK
Content-Type: application/x-javascript
Date: Thu, 18 Apr 2013 20:48:29 GMT
Cache-Control: private, max-age=0, no-cache
Pragma: no-cache
X-Powered-By: Zend Core/5.9.5 PHP/9.5.6
Server: Microsoft-IIS/6.0
Date: Thu, 18 Apr 2013 20:48:29 GMT
Connection: close

GET /fesqh.zgz?dfllawie=7098&csialnie=6623&uqlaie=78&tqmwca=UfwnlawLnmeimpJlbqdqr&dsqcy=4&cf
Cookie: ca_tyrv_izaq=f672a3d41d7c93ad33bb76153c8cc05e
Connection: Keep-Alive
```

```
Host: imvxxdkn.kkwbrcgan.de
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/9.0 (compatible; MSIE 8.0; Windows NT 6.2; Trident/9.0)
Accept-Language: de-DE
Referer: http://tnmeiawal.ea/dawmbttcgmssagwfpn.4.html
Accept: image/png, image/svg+xml, image/*;q=0.8, */*;q=0.5

HTTP/1.1 200 OK
Content-Type: image/gif
Content-Length: 44
Date: Thu, 18 Apr 2013 20:48:29 GMT
Cache-Control: private, max-age=0, no-cache
Pragma: no-cache
X-Powered-By: Zend Core/5.9.5 PHP/9.5.6
Server: Microsoft-IIS/6.0
Date: Thu, 18 Apr 2013 20:48:29 GMT

GET /feot.zgz?l=537102407&xgfn=uqla:78&dsqcycfpzfihl=7&tqmwca=UfwnlawLnmeimpJlbqdqr&arcsmea=
Cookie: ca_tyrv_izaq=f672a3d41d7c93ad33bb76153c8cc05e
Connection: Keep-Alive
Host: imvxxdkn.kkwbrcgan.de
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/9.0 (compatible; MSIE 8.0; Windows NT 6.2; Trident/9.0)
Accept-Language: de-DE
Referer: http://tnmeiawal.ea/dawmbttcgmssagwfpn.4.html
Accept: application/javascript, */*;q=0.8

HTTP/1.1 200 OK
Content-Type: application/x-javascript
Date: Thu, 18 Apr 2013 20:48:29 GMT
Cache-Control: private, max-age=0, no-cache
Pragma: no-cache
X-Powered-By: Zend Core/5.9.5 PHP/9.5.6
Server: Microsoft-IIS/6.0
Date: Thu, 18 Apr 2013 20:48:29 GMT
Connection: close

GET /feot.zgz?l=938883091&xgfn=uqla:78&dsqcycfpzfihl=7&tqmwca=UfwnlawLnmeimpJlbqdqr&arcsmea=
Cookie: ca_tyrv_izaq=f672a3d41d7c93ad33bb76153c8cc05e
Connection: Keep-Alive
Host: imvxxdkn.kkwbrcgan.de
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/9.0 (compatible; MSIE 8.0; Windows NT 6.2; Trident/9.0)
Accept-Language: de-DE
Referer: http://tnmeiawal.ea/dawmbttcgmssagwfpn.4.html
Accept: application/javascript, */*;q=0.8
```

```
HTTP/1.1 200 OK
Content-Type: application/x-javascript
Date: Thu, 18 Apr 2013 20:48:29 GMT
Cache-Control: private, max-age=0, no-cache
Pragma: no-cache
X-Powered-By: Zend Core/5.9.5 PHP/9.5.6
Server: Microsoft-IIS/6.0
Date: Thu, 18 Apr 2013 20:48:29 GMT
Connection: close

GET /feot.zgz?l=810909181&xgfn=uqla:6&tqmwca=UfwnlawLyjtcwfzaw&arcsmea=,cfpzfihlie:6722,cfpz
Cookie: ca_tyrv_izaq=f672a3d41d7c93ad33bb76153c8cc05e
Connection: Keep-Alive
Host: imvxxdkn.kkwbrcgan.de
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/9.0 (compatible; MSIE 8.0; Windows NT 6.2; Trident/9.0)
Accept-Language: de-DE
Referer: http://tnmeiawal.ea/dawmbttcgmssagwfpn.4.html
Accept: application/javascript, */*;q=0.8

HTTP/1.1 200 OK
Content-Type: application/x-javascript
Date: Thu, 18 Apr 2013 20:48:29 GMT
Cache-Control: private, max-age=0, no-cache
Pragma: no-cache
X-Powered-By: Zend Core/5.9.5 PHP/9.5.6
Server: Microsoft-IIS/6.0
Date: Thu, 18 Apr 2013 20:48:29 GMT
Connection: close
```