9gag.com : A Website Analysis

Florian Damhaut

***Abstract*—This rapport offers a detailed view of the information transfers mechanisms and protocols used on `9gag.com` as well as any interesting information found during the analysis of this Hong Kong-based website. `9GAG` is a social media on which people can share user-generated content.**

INTRODUCTION

A. State of the art

One of the past work [1] for this class had the same subject. Therefore, I tried to focus on the basic aspects as well as the changes that occurred in the past five years and everything interesting that was not included in his rapport. This rapport is still a standalone piece.

B. Tools & Methodology

For this paper, both a clean version on Mozilla Firefox 64.0 and Google Chrome were used to gather data. A modified version of Google Chrome[1][2] was also used to perform tests without advertisement and tracking. DNS look-up and trace were performed using an Ubuntu terminal as well as the Windows 10 PowerShell. SSLlabs was used to provide data about the TLS version used. Finally, WireShark has been used to analyse the transmission of packets.

I. DNS

A. Main site analysis

The research shows that `9gag.com` only uses 4 IPv4 addresses (A records) which did not change over the time of this analysis.

- `151.101.2.133`
- `151.101.66.133`
- `151.101.130.133`
- `151.101.194.133`

They all have a relatively small Time To Live (TTL), 300 seconds, to ensure better load-balancing and responsiveness in case of fail-over. On the other hand, the DNS authorities;

- `ns-1673.awsdns-17.co.uk`
- `ns-408.awsdns-51.com`
- `ns-1294.awsdns-33.org`
- `ns-629.awsdns-14.net`

have a much longer TTL to keep cost on the lower side and improve performance.

To access the main site, we need to connect to "fastly", the CDN in charge of the delivering the main web page. For that to happen, our Internet Service Provider (ISP) go through one of Amsterdam's internet exchange before reaching fastly.

Depending on the original connection the path is different, if the ISP's is **Proximus** then the connection go through *Belgacom*[3], *Telia* before reaching *fastly*. Whereas if the connection is initiated from a network connected to **Belnet**, it goes through `10.26.112.33`[4] before joining *Congento* and then finally reaching *fastly* in Amsterdam.

They also registered `www.9gag.com` as a CNAME for `9gag.com`.
Fastly's server also implement Server Name Indication. [5] (see Part IV. TLS)

B. Add-on sites

`img-9gag-fun.9cache.com`, `miscmedia-9gag-fun.9cache.com` and `assets-9gag-fun.9cache.com` are all Canonical NAMEs (CNAME) which refers to `*.cdn.cloudflare.net`. These sub-domains serve as a facade to link 9gag with another Content Delivery Network (CDN), cloudflare, that is used for storing and distributing the images and videos shown on 9gag.
They are linked to IPv4 addresses.
`accounts-cdn.9gag.com` is another CNAME to refering to `accounts-cdn.9gag.com.cdn.cloudflare.net`. It is the only part of the site capable of handling IPv6 which is wierd considering it's only used for account management.

If tested in belgium, all of the previous domain are delivered by cloudflare, a well-known CDN, through the Belgian National Internet eXchange (BNIX).

C. Advertising related domains

As of right now I found no less than 11 advertisement companies' domains directy linked to `9gag.com`.

There will not be an in depth inspection of all of them as it would not be of much interest. Nonetheless, they will all be listed during the HTTP ressources analysis.

1) Google AdSense: `adservice.google.be` is a CNAME for a *doubleclick* domain which support both IPv4 and IPv6 requests.

2) DoubleClick , a Google subsidiary: `cm.g.doubleclick.net` is also a CNAME for `pagead.l.doubleclick.net` but does not support the IPv6 protocol.
`googleads.g.doubleclick.net` is similar to `adservice.google.be`.

[1]Advertisement blocker : AdBlock plus & Ublock Origin
[2]Tracking blocker : Ghostery
[3]Proximus' previous name. Apparently they didn't bother changing their server's name
[4]A private IP that we strongly suppose is part of the Belnet network as every trace tested leaving it contains this IP.

3) Amazon Ad System: is divided between two services; `c.amazon-adsystem.com` is a CNAME for a cloudfront domain supposedly used for delivering Ads and `aax.amazon-adsystem.com` used for advertising info and cookies management.

4) `www.ora.tv`: is a CNAME for some randomly generated cloudfront domain hosting the site. Strangely enough, `ora.tv` is not a CNAME for `ora.tv` but instead refers to 2 IPv4 addresses according to amazon web service. Though, when we send an HTTP request for `ora.tv`, it's received with an 301 Error redirecting us to `www.ora.tv`.

D. Tracking related domains

Multiple tracking and data-gathering domains have been reached during the analysis, but none of them really stood out as interesting DNS-Wise. In the same way as Advertising's domains, they will be reviewed later in the HTTP section.

II. HTTP

9gag use the version 2.0 of the HTTP protocol with a security layer (HTTPS / TLSv1.2).

A. Initially Loaded Resources

Whenever loading the page, a typical browser send approximately 280 requests. As 9gag is all about sharing pictures, gif and short video, during a normal usage *meaning you scroll for quite some times*, most of the resources are loaded from one of the three cloudflare's related sub-domain. `img-9gag-fun.9cache.com` is in charge of providing the "user-created content", `miscmedia-9gag-fun.9cache.com` provides us with the others necessary pictures such as the full icons and banners and `assets-9gag-fun.9cache.com` used for small PNG icons and GIFs with a transparency layer.
The assets also cover CSS style-sheets, 9gag's javascript.

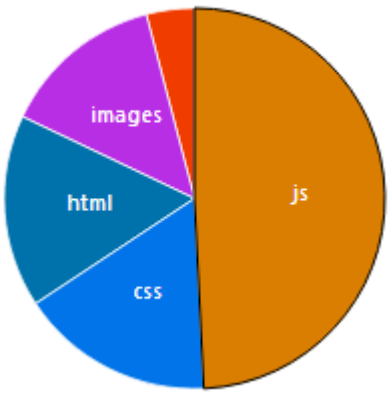

Figure 1. Initial load's resources

However, the amount of javascript needed to run every advertising and tracking resources as well as handling the site basic use is nothing to make light of. You can observe on 1 that almost half of the original load, meaning without using the infinite scrolling, is javascript code.

B. Images, gifs and videos

To store and access quickly the three main types of data used, 9gag use new web format for better quality and smaller size called WebP and WebM and more generic format such as jpeg.

WebP *[6] is a format of picture currently developed by Google that support both lossy and lossless compression. It has been shown to be more efficient than PNG.*

WebM *[7] is a video format based on mkv that is optimized for HTML5 usage. All of the "user-created content" with animation posted on 9gag is then converted to WebM before being available on the site. All the animated resources have a sister WebP resources to show the user before the video is needed and downloaded*

C. Cookies

When connection to 9gag, multiples cookies are used to register one's use of the site. They are used to track the user around the different part of the website.

On 9gag, there were 4 cookies used to track the client that were not used by 9gag directly ;

`_ga` *Google Analytics tracking cookie*
`fr` *Facebook tracking cookie*
`UID, UIDR` *Scorecard Research tracking cookies*
`_pk_id.7.f7ab` *Piwik JavaScript Tracking Client cookie*

D. TCP Ports

All requests are either using the 80th or the 443rd port as it is custom with HTTP and HTTPS request.
The latter being much more common.

E. Non-Standard HTTP headers

X-Cache *signal that their proxy has a valid copy of the page in its cache or not.*

Timing-Allow-Origin *specifies access to the Resource Timing API.*

Cf-cache-status *indicate where the resource come from; the CDN cache or the origin server, and the reason.*

Cf-ray *allows the website to track user through Cloudflare's network.*

F. Advertising & tracking related resources

When connected to 9gag, the browser receives an enormous amount of data unrelated to the actual content of the site. In fact, whenever using a web browser with ad-blocking software, the number of request plummet to roughly 140 out of the original 280. I have compiled below a list of all the domains I reached during the tests that I can prove, without a doubt, to be used for advertising purposes.

1) Google AdSense: `adservice.google.be`
2) DoubleClick , a Google subsidiary: `googleads.g.doubleclick.net`
3) Amazon Ad System: `*.amazon-adsystem.com`
4) Rubicon Project: `fastlane.rubiconproject.com`

5) DeployAds: `*.deployads.com`
6) Adnxs: `ib.adnxs.com`
7) Pubmatics: `ads.pubmatic.com`
which then lead to multiples others advertising domains;

- Adform
- ContextWeb
- Amobee

8) DistrictM: `cdn.districtm.io`
9) Adify: `ad.afy11.net`
10) OpenX: `ninegag-d.openx.net`
11) Programattik: `ads.programattik.com`

As for tracking, almost every GIFs downloaded are 1*1 uncoloured pixel that are used as web beacon. These are used to track the online behaviour of the user. Most social media button also act as web beacon. They also use XHR to transfer usage data from 9gag to their own sites.

G. Special advertising and embed javascript

`Ora.tv` is a special case of advertising. Whenever you connect to 9gag without using any form of ad-blocking software, you download video from there. They are then used between "user-created content" on the site. I strongly suspect that these embed videos are used to increase the view count on this specific site as it is, unfortunately, quite common practice on the internet.
One could suspect that the relationship between these two is quite strong as if you browser through 9gag with an ad-blocker, you can see two content-separating line one after the other, which is not something you can observe with the embed video showing.
What is more worrying is the javascript code that is downloaded and executed each time the site is opened. The source is quite clear, but the reason and capabilities of such a program are unknown.
This piece of code has been made to be incomprehensible by removing any indentation and using unintelligible variables and script names. As the script is only executed whenever a video is playing, we assume that it is to tally the number of time videos where watched. However, there is still no proof that it is its only function.

H. Easters eggs

During my analysis of 9gag, I found two hidden easters' eggs.
The first being their signature in their code (Figure : 2).

And the second is the name of their server, Potato 1.0, a popular meme on the platform.

III. TLS

The data gathered to write this part mostly come from SSLlabs'[5] analysis of both the main domain and sub-domains.

[5]www.ssllabs.com

Figure 2.

A. `9gag.com`

9gag only support the vesion 1.2 of the TLS protocol with an HSTS measure, meaning it can only be access using a secure HTTPS connection.
Fastly's server implement Server Name Indication, which is used for hosting multiple HTTPS domain on a single IP. It allows them to store and deliver 9gag on multiples IPs for cheaper as other site also use these IPs. [5] This technique is only useful here because the main site does not need to transfer a large amount of data.
It use the `TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256` cipher suite composed of [2]

TLS_ECDHE *which allows two parties to establish a shared secret over an insecure channel.*

RSA *for asymmetric encryption of the symetric encryption key.*

AES_128_GCM *for symetric encryption.*

SHA256 *: Hashing to prove integrity.*

This cipher suite provide froward secrecy.
9gag also implement OCSP stapling which assign the burden of CA proof to the server eliminating the need for the client to contact CA authorities and improving performances. [4]

B. `img-9gag-fun.9cache.com`

9gag's sub-domains are hosted on a different sub-domain, explaining the difference between them and the original site. They can support 3 versions of SSL : 1.0, 1.1 and 1.2, which allows for a wider range of connection possible but also weaker security. Indeed TLS v1.0 & v1.1 have weaker algorithms than v1.2. Fortunately, the servers are protected in case of protocol downgrade attacks with the TLS Fallback Signaling Cipher Suite Value (SCSV). [3]

All sub-domains are created equals.

IV. TCP

During a short, common use of 9gag, on average 220 TCP channel are opened. They are all either using the 80th or the 443rd port as it is custom with HTTP and HTTPS request.
The latter being much more common as it was used by more than 95% of the TCP streams on average. All the HTTP request were shutdown to leave place to their HTTPS counterparts.
Most of the streams are only started once during the first load of the page for a small amount of data (10kB).

9gag only support IPv4, Happy Eyeballs is thus not utilised. Most of the connection end gracefully.

A. Characteristics of the connection

1) Number of maximum concurrent streams: 6

2) Main site's RTT: 13ms

3) Image CDN's RTT: 9ms

*4) **Main site connection**:*

- Selective ACK permitted
- Max packet size : 1452 bytes

Window size Client

- Begin : 64240
- Scaling factor : 256
- End : 66304

Window size Server

- Begin : 26883
- Scaling factor : 256
- End : 36608

*5) **Image's CDN connection**:*

- Selective ACK permitted
- Max packet size : 1452 bytes

Window size Client

- Begin : 64240
- Scaling factor : 256
- End : 18884672

Window size Server

- Begin : 29200
- Scaling factor : 1024
- End : 43008

B. Effect of scrolling

To analyse how 9gag handled the infinite scrolling, tests were made with and without scrolling down. These allowed us to conclude that only one TCP stream is used to provide us with the data necessary for the infinite scrolling. It can be easily identified on the figure 3 because of the sheer difference in number of byte transmitted as well as the domains to which it is connected, the one directly identified before as the CDN for "user-generated content".

Address A	Port A	Address B	Port B	Bytes
192.168.1.13	50017	img-9gag-fun.9cache.com.cdn.cloudflare.net	443	93 M
192.168.1.13	50022	152.195.132.189	443	3454 k
2a02:a03f:38d8:c000:8d...	50084	s0-2mdn-net.l.google.com	443	253 k
192.168.1.13	50135	d22gnlv6tz69yc.cloudfront.net	443	164 k
192.168.1.13	50144	s1-eu.adformnet.akadns.net	443	162 k
2a02:a03f:38d8:c000:8d...	50156	ams15s30-in-x04.1e100.net	443	111 k
2a02:a03f:38d8:c000:8d...	50027	pagead46.l.doubleclick.net	443	82 k
192.168.1.13	50036	fra1-ib.adnxs.com	443	62 k
192.168.1.13	50102	e4937.d.akamaiedge.net	443	62 k
2a02:a03f:38d8:c000:8d...	50082	s0-2mdn-net.l.google.com	443	50 k

Figure 3. Number of bytes transferred in the top 10 TCP streams

C. Effect of advertisement and tracking

As mentioned before, 9gag is jammed with advertising and tracking, further proof of that is the number of TCP streams opened during an average use using multiple ad-blocking software is only of 34, about 170 less than without or only 15% of the original number.
If the scan is launched a few seconds after, only 100 TCP stream will be used for the same amount of time. We can thus imagine that some ads are refreshed every so often, but most of them will only be loaded once.

V. Conclusion

In a nutshell, 9gag is fairly optimized and all of the requests and response are forcibly secured. Their usage of a single TCP never-ending stream to handle the infinite scrolling is also interesting as it provides better performance than constant reloading. However, we found out that 9gag is ingrained with advertising and tracking which greatly lower its performance.

Personally, this analysis allowed to deepen my understanding of the course. On top of that, it provided me with the opportunity to learn a lot about Internet, a medium we use so much and know so little about.

References

[1] Hardy Simon, *9gag Analysis*
cdn.uclouvain.be/groups/cms-editors-ingi/lingi1341-rapports/9gag.pdf

[2] RoraZ *Cipher Explanation*
security.stackexchange.com/questions/65622/client-server-encryption-technique-explanation-tls-ecdhe-rsa-with-aes-128-gcm-s

[3] Network Working Group, B. Moeller *SCSV definition*
tools.ietf.org/id/draft-ietf-tls-downgrade-scsv-01.html

[4] Wikipedia *OCSP Stapling*
en.wikipedia.org/wiki/OCSP_stapling

[5] Wikipedia *SNI*
en.wikipedia.org/wiki/Server_Name_Indication

[6] Wikipedia *WebP*
en.wikipedia.org/wiki/WebP

[7] Wikipedia *WebM*
en.wikipedia.org/wiki/WebM