LINGI 1341 - Analyse du site web *LinkedIn*

Merlin Camberlin · 09441700

I. INTRODUCTION

LinkedIn est de loin le réseau social professionnel le plus connu et le plus utilisé. Créé en 2004, *LinkedIn* compte à ce jour plus de 500 millions d'utilisateurs.[1] Bien que fort utilisé, combien se sont demandés comment ce site fonctionnait-il, comment diffusait-il son contenu, comment s'assurait-il de la sécurité des échanges entre lui et ses clients ?

L'objectif de ce rapport sera de répondre à ces différentes questions. Pour ce faire, le rapport sera divisé en trois parties. La première concernera le protocole `HTTP`, la seconde le protocole `DNS` et la dernière abordera le protocole `TLS`.

II. ANALYSE HTTTP

Dans cette section, les requêtes et réponses `HTTP` lorsqu'un client consulte le site web *LinkedIn* seront analysées. En particulier, cette partie répondra aux différentes questions suivantes: quelles sont les requêtes qui sont effectuées? Quelles ressources sont téléchargées, quels serveurs sont interrogés et comment le protocole `HTTP/2` est utilisé.

A. Noms de domaines, numéros de ports et ressources utilisées

Lorsque l'on se rend sur *LinkedIn*.com avec le navigateur Firefox (en ayant préalablement vidé le cache et les données de navigation), une série de requêtes `HTTPS` sont effectuées. Ci-après, sont énumérés les différents noms de domaines contactés. Etonnemment une grande série d'entre-eux ne font pas partie des services de *LinkedIn*.

- *linkedin.com*: pour afficher le contenu et services proposé par *LinkedIn*.
- *youtube.com*: pour afficher la vidéo présente sur la page d'accueil.
- *dmp.demdex.net* et *lnkd.demdex.net*: pour recueillir des informations sur le client qui consulte le site. Derrière le nom de domaine *demdex*, se cache *Adode Audience Manager*, une platforme proposant un service de recueil d'information pour fournir une vue uniformisée de l'audience du site.[2] Les préfixes `dpm` et `lnkd` sont les abréviations respectives de *linked* et de *Data Provider Match*. Ce dernier préfixe indique aux systèmes internes d'*Adobe* qu'un appel de *Audience Manager* ou qu'un service d'identification transmet les données du client pour une synchronisation ou pour une demande d'identification.
- *static-exp1.licdn.com* est un sous-domaine de *licdn.com* appartenant à *LinkedIn*. Ce domaine est également en charge de l'analyse de l'audience du site web.

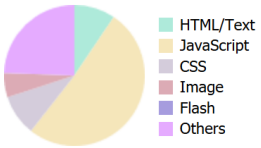
D'un navigateur à un autre, le nombre de noms de domaines contactés peut varier. A la liste précédente, peuvent s'ajouter

- stats.g.doubleclick.net
- google-analytics.com
- sb.scorecardresearch.com
- yt3.ggpht.com

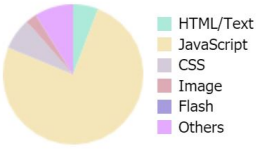
Les numéros de port numéro 443 et 80 correspondant respectivement à celui d'`HTTP` et celui d' `HTTPS` sont ouverts. A ce sujet, aucune requête ou réponse n'est faite en `HTTP`. Même en consultant `http://www.linkedin.com`, on est redirigé vers la version `HTTPS` avec un status *HTTP 301 Moved Permanently*. En outre, en consultant `https://www.linkedin.be`, on est égalemenent redigéré avec le même status mais cette fois vers `https://be.linkedin.com/`.

B. Ressources utilisées

La FIGURE 1 ci-après illustre la répartition des ressources échangées lors du chargement de la page d'accueil et lors du chargement du fil d'actualité de l'auteur.



(a) Lors du chargement de la page d'accueil



(b) Lors du chargement du fil d'actualités

Fig. 1: Résumé des différentes ressources téléchargées [3]

En grande partie, c'est du code *JavaScript* qui est échangé et d'autant plus lorsque l'on consulte le fil d'actualité. On se serait attendu à avoir une augmentation du nombre d'image en consultant le fil d'actualité mais l'augmentation n'est pas significative. Ceci s'explique par le fait que les images et post sont chargées au fur et à mesure de la lecture dans le fil.

C. Requêtes HTTP

En utilisant le navigateur *Mozilla Firefox*, en ayant préalablement vidé le cache et les données de navigation,

quelques requêtes sont effectuées en consultant la page d'accueil.

Les en-têtes des requêtes effectuées contiennent quelques champs intéressants:

- **Accept-Language** indique quelles sont les langues que le client est capable de comprendre pour que le serveur adapte son contenu à sa langue parlée. Alors que toutes les données de *Mozilla Firefox* ont été vidée, le navigateur présumait déjà que la langue préférée par le client était le français. (*fr,fr-FR;q=0.8,en-US;q=0.5,en;q=0.3*)
- **Cache-Control** permet de définir la politique de cache des contenus utilisée dans les requêtes et dans les réponses. Pour désactiver la mise en cache, no-cache a systématiquement été spécifié.
- **Connection** indique si la connection réseau doit rester ouverte après la fin de la transaction courante ou non. Si keep-alive est spécifié, la connexion sera persistante et les requêtes suivantes se feront en utilisant cette même connexion.
- **TE** spécifie les encodages de transfert que le client est prêt à accepter. Avec la version 2 de `HTTP`, pour être valide, ce champ doit être *trailers* indiquant un codage de transfert fragmenté.
- **Upgrade-Insecure-Requests** indique serveur les préférences du client pour une réponse chiffrée et authentifiée.
- **User-Agent** indentifie le type d'application, le système d'exploitation, le fournisseur de logiciel ou la version logicielle du client. Etonnement, même en utilisant chrome, le user-agent est spécifié avec Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/78.0.3904.108 Safari/537.36.
- **Referer** contient l'adresse de la page précédente visitée sur laquelle un lien a été suivi. Ce champ permet aux serveurs d'identifier la provenance des visiteurs. Il est notamment présent dans les requêtes contactant les noms de domaine recueillant des informations sur le visiteur du site à des fins analytiques.
- **Origin** indique la provenance de la requête. C'est une variante du champ *Referer* qui, contrairement à ce dernier, n'inclut pas le chemin complet mais seulement le nom du serveur.
- **Cookie** contient les cookies `HTTP` précédemment envoyé par le serveur.
  [4]

Sur le navigateur habituel de l'auteur (à savoir *Google Chrome*), en conservant les données de navigation et en désactivant le cache, plusieurs nouveaux champs sont présents:

- **DNT** pour *do not track* indique aux applications web que l'utilisateur ne désire pas être suivi. DNT vaut 1 si l'utilisateur ne veut pas être suivi, sinon 0. [5]
- **Sec-fetch-mode** expose le mode d'une requête à un serveur
- **Sec-fetch-site** expose la relation entre l'origine d'un initiateur de requête et l'origine de sa cible.
- **Sec-fetch-user** indique si une demande de navigation a été déclenchée ou non par l'activation de l'utilisateur

Alors que la toute première requête de *LinkedIn* ne contenait pas de champ cookie avec la navigateur *Firefox*, avec le navigateur *Chrome*, la requête en contient directement un.

D. Réponses `HTTP`

Les en-têtes des réponses des domaines contactés contiennent quelques champs intéressants:

- **Content-Security-Policy** permet aux administrateurs d'un site web de contrôler les ressources que l'agent utilisateur est autorisé à charger pour une page donnée. Ce champ permet de se protéger contre les attaques de *cross-site scripting*.
- **HTTP Strict Transport Security** permet de forcer les navigateurs à exclusivement utiliser le protocole HTTPS pour communiquer avec le serveur. Ce champ permet d'éviter une attaque dite de l'homme du milieu. Si un connexion HTTP est acceptée et que le serveur la redirige vers une connexion HTTPS, le visiteur, pendant cette redirection peut potentiellement être exposé à une attaque de l'homme du milieu. En effet, un pirate pourrait détourner cette redirection pour envoyer le visiteur vers un site malveillant au lieu de la version sécurisée attendue.
- **X-XSS-Protection** bloque le chargement des pages lorsque une attaques XSS (cross-site scripting) est détectée.
- **Set-Cookie** envoie un cookie du server vers le client. Ce cookie sert d'identifiant unique du visiteur du site. Lorsque le client revient sur ce dernier, il renvoie son cookie en complétant le champ Cookie des requêtes `HTTP` qu'il effectue. Sur *mozilla Firefox*, lors de la première requête vers *LinkedIn*.com, 5 cookies sont instaurés.
- **x-firefox-spdy** indique que le protocole `SPDY` est utilisé. Ce protocole a été conçu par *Google* pour augmenter les performances d'`HTTP` sans devoir le remplacer. Depuis mai 2015, l'IETF l'a intégré dans la version d'`HTTP/2`. Ce champ n'a donc plus vraiment d'intérêt si ce n'est pour indiquer que la version `HTTP/2` est utilisée.
- **x-li-pop** indique quel `POP` (Point of Presence) a été utilisé pour distribuer le contenu du site. Ce champ est propre à *LinkedIn*. De plus amples informations seront données à ce sujet dans la SECTION F.

E. Implémentation HTTP/2

LinkedIn supporte `HTTP/2` depuis l'année 2017. Les deux principales fonctionnalités de `HTTP/2` qu'il utilise sont le multiplexage de flux et la priorisation des flux. Cette dernière permet d'établir de définir une préférence dans l'ordre de chargement des ressources. Ainsi, *LinkedIn* préférencie le chargement du code `HTML` en premier étant donné que celui-ci est la première ressource dont un navigateur a besoin pour diffuser une page Web. Quant au multiplexage de streams,

LinkedIn s'en sert pour permettre, sur une seule connexion TCP, de charger plusieurs ressources.

Les domaines contactés recueillant des informations sur les visiteurs du site, quant à eux utilisent la version HTTP 1.1 .

A ce jour, *LinkedIn* n'utilise pas le serveur push. En effet, l'analyse chronologique des requêtes et réponses HTTP le confirme en montrant que les dépendances de la page ne sont pas téléchargées avant la fin du téléchargement du fichier HTML.

F. Optimisation implémentée

LinkedIn sert son contenu différemment selon son type statique ou dynamique. Le contenu dynamique (HTML, JSON,...) est distribué par leur propre POP (Point Of Presence) tandis que le contenu statique (CSS, images etc.) est fourni à partir de réseaux de diffusion de contenu tiers (CDN). Pour améliorer la distribution de son contenu dynamique, *LinkedIn* utilise des PoPs (*Point of Presence*) et un *Load Balancer*. La FIGURE 2 ci-après les illustre. Les PoPs sont des centres de données à petite échelle qui sont distribués spatialement. Leur utilisation permet de réduire le temps d'aller-retour entre le client et le serveur web. [6]

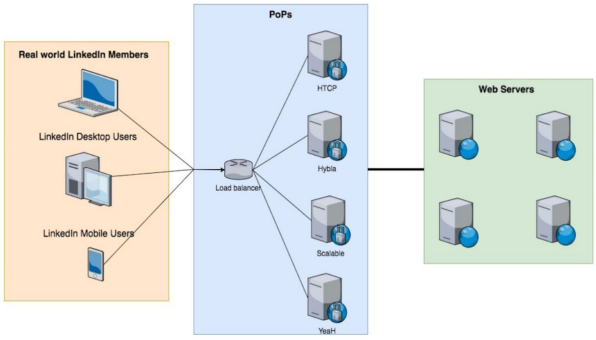


Fig. 2: Représentation des différents acteurs lors du chargement de pages[6]

III. ANALYSE DNS

Cette deuxième partie se concentrera sur le DNS. Elle répondra notamment aux questions suivantes: quelles sont les adresses IP qui permettent de contacter *LinkedIn*, quelles sont les serveurs responsables de *LinkedIn*, combien de temps une reponse DNS peut-elle rester en cache.

A. Serveurs DNS responsables

Le site de *LinkedIn* est sous l'autorité de 8 serveurs DNS. La plupart d'entre-eux supportent IPv4 et IPv6. Leurs noms et adresses sont donnés sur la FIGURE 3 ci-après.

Nom	Adresse IPv4	Adresse IPv6
dns1.p09.nsone.net	198.51.44.9	2620:4d:4000:6259:7::9
dns2.p09.nsone.net	198.51.45.9	2a00:edc0:6259:7::9
dns3.p09.nsone.net	198.51.44.73	2620:4d:4000:6259:7::90
dns4.p09.nsone.net	198.51.45.73	2a00:edc0:6259:7::90
ns1.p43.dynect.net	208.78.70.43	2001:500:90:1::43
ns2.p43.dynect.net	204.13.250.43	/
ns3.p43.dynect.net	208.78.71.43	2001:500:94:1::43
ns4.p43.dynect.net	204.13.251.43	/

Fig. 3: Noms et adresses des différents domaines qui ont aurotité sur *LinkedIn*

B. Serveurs HTTP contactés

Que *LinkedIn* soit contacté à partir du campus UCL, du resolver DNS de google ou du domicile de l'auteur, ses deux adresses sont systématiquement identiques et sont: 108.174.10.10 en IPv4 et 2620:109:c002:0:0:0:6cae:a0a en IPv6. Chacune a un TTL (*time to live*) de l'ordre d'une heure.

Les servers HTTP contactés (ceux énumérés dans la SECTION II-A) sont tous accessibles en IPv4 et en IPv6. D'un resolver à l'autre, leur adresse IP varie à l'exception des domaines suivants: *linkedin.com* et *sb.scorecardresearch.com*.

En revanche, lorsque le domaine lnkd.demdex.net est contacté, avec le resolveur UCL ou avec le resolver de google, toutes les adresses retournées sont différentes.

C. Utilisation de CNAME

Aucun alias n'est présent en contactant *LinkedIn*.com . En revanche, lorsque l'on contacte *www.LinkedIn.com*, on rencontre une série d'alias:
www.linkedin.com. → www-src.linkedin.com. → 2-01-2c3e-003c.cdx.cedexis.net. → pop-tln1-alpha.www.linkedin.com. → 185.63.144.1
Les autres domaines cités en SECTION II-A en utilisent également. le domaine *lnkd.demdex.net.* utilisent des alias qui mène à des services proposés par *amazonaws* (des services web d'*amazon*). En effet, la suite d'alias provenant de *demdex.net* est :
lnkd.demdex.net. → gslb-2.demdex.net. → edge-irl1.demdex.net. → dcs-edge-irl1-876252164.eu-west-1.elb.amazonaws.com. → 52.17.215.83; 34.247.58.231; 34.253.43.81; 34.240.143.140; 34.241.149.220; 34.240.220.248; 34.247.192.223; 52.16.220.22 . On peut remarquer que le dernier alias ne pointe pas vers une seule adresse, mais vers huit et ce, pour partager la charge entre les différents serveurs.

D. TTL (Time-To-Live)

Le TTL est la durée pendant laquelle le résultat d'une requête DNS peut rester en mémoire cache. Selon le resolver utilisé, les TTL des réponses DNS peuvent varier. Par exemple, si on utilise le resolver de son domicile, la valeur du TTL est toujours la même. Au contraire, en utilisant le resolver public de *Google*, comme illustré sur la FIGURE 4, les TTL varient d'une requête à l'autre.

```
dig @8.8.8.8 +noall +answer d-opt +ttlunits linkedin.com -t A
          linkedin              24m35s  IN      A       108.174.10.10
dig @8.8.8.8 +noall +answer d-opt +ttlunits linkedin.com -t A
          linkedin              43m34s  IN      A       108.174.10.10
dig @8.8.8.8 +noall +answer d-opt +ttlunits linkedin.com -t A
          linkedin              23m30s  IN      A       108.174.10.10
dig @8.8.8.8 +noall +answer d-opt +ttlunits linkedin.com -t A
          linkedin              41m6s   IN      A       108.174.10.10
dig @8.8.8.8 +noall +answer d-opt +ttlunits linkedin.com -t A
          linkedin              51m43s  IN      A       108.174.10.10
dig @8.8.8.8 +noall +answer d-opt +ttlunits linkedin.com -t A
          linkedin              21m23s  IN      A       108.174.10.10
dig @8.8.8.8 +noall +answer d-opt +ttlunits linkedin.com -t A
          linkedin              1m22s   IN      A       108.174.10.10
dig @8.8.8.8 +noall +answer d-opt +ttlunits linkedin.com -t A
          linkedin              53m24s  IN      A       108.174.10.10
```

Fig. 4: Variation du TTL avec le resolver public 8.8.8.8

E. Particularités

Des records de types SOA ont été trouvés. SOA est une abréviation pour *Start of Authority* et contient des informations concernant la zone DNS auquel il se rapporte, notamment concernant le transfert de zone.

Le DNS utilisé pour contacter *LinkedIn* contient un enregistrement particulier. Il s'agit de l'enregistrement HINFO qui indique le type de processeur et le système d'exploitation de l'hôte. A ce record était répondu HINFO 13 RFC 1035.

IV. ANALYSE TLS (OU SSL)

Cette dernière partie concernera TLS. Elle répondra à la question suivante: Quelles sont les stratégies mises en place par *LinkedIn* pour sécuriser et authentifier ses échanges avec ses clients ?

Sur le site *LinkedIn*, TLS est utilisé sur HTTP pour sécuriser les échanges de données. *LinkedIn* supporte actuellement TLS 1.1 et TLS 1.2 mais pas encore la version TLS 1.3.

A. Handshake

Lors de l'établissement d'une connexion TLS, client et le serveur entrent dans la phase dite de *handshake*. Dans cette phase, ils négocient les futurs paramètres de sécurité qui seront utilisés pour sécuriser et authentifier les échanges. En pratique, ils s'échangent leur liste de suites cryptographique, par ordre de préférence, qu'ils supportent. Chaque suite comporte, dans cet ordre, un algorithme d'échange de clés, un algorithme d'authentification, un algorithme de chiffrement par bloc et un algorithme d'authentification de message (MAC).

Dans sa version TLS 1.2, *LinkedIn* supporte

- 3 algorithmes d'échange de clés: ECDHE, DHE et RSA
- 1 algorithme d'authentification : RSA
- 3 algorithmes de chiffrement par bloc : AES en 128 et 256 bits en mode GCM, CBC et CCM, CHACHA20_POLY1305, ARIA en 128 et 256 bits en mode GCM
- 3 algorithmes d'authentification de message: SHA384, SHA256, SHA

La suite cryptographique préférée par les serveurs IPv4 et IPv6 de *LinkedIn* est : ECDHE, RSA, AES en 128 bits en mode CM, SHA256.

En comparant la suite cryptographique proposée par *LinkedIn* avec celle préférée par son navigateur, l'auteur a remarqué que celles-ci différaient d'un navigateur à l'autre.

Le navigateur *Firefox* v71.0 préférencie AES_128_GCM_SHA256 tandis que le navigateur *Chrome* préférencie la suite GREASE_IS_THE_WORD_3A. Cette dernière suite semble particulière tant par son nom que par son format qui ne respecte pas celui annoncé ci-avant. En réalité, GREASE vient de *Generate Random Extensions And Sustain Extensibility* et est un mécanisme de prévention de défaillances d'extensibilité dans TLS. Pour rester extensibles, les serveurs devraient ignorer les suites cryptographiques inconnues. Ce nom de suite permet de vérifier que les serveurs réagissent bien correctement.

B. Perfect forward secrecy

La propriété de *Perfect forward secrecy* garantit qu'une information chiffrée aujourd'hui restera confidentielle même si la clef privée d'un des deux correspondants est compromise. Pour y parvenir, serveur et client négocient un secret partagé indépendant de la clé privée du serveur à l'aide d'algorithmes Diffie-Hellman. Parmi les algorithmes précédemment cités, DHE_RSA, ECDHE_RSA et ECDHE_ECDSA vérifient cette propriété. DHE vient de *Diffie-Hellman Ephemeral* et ECDHE de *Elliptic Curve Diffie-Hellman Ephemeral*.

V. CONCLUSION

A l'issu de ce rapport, l'auteur a été surpris à plusieurs reprises. Sa surprise la plus inattendue à notamment été celle de découvrir la présence de *Google* pour son service *google analytics*, d'*Adobe* pour son service *audience manager* et enfin *Amazon* pour ses services web dans le fonctionnement du site de *LinkedIn* alors que ceux-ci sont indépendants.

REFERENCES

[1] Wikipédia. LinkedIn [s.l.n.d.] Disponible sur : https://fr.wikipedia.org/wiki/LinkedInl. (15/12/19)

[2] Adobe Corporation. Understanding Calls to the Demdex Domain [s.l.n.d.] Disponible sur : https://docs.adobe.com/content/help/en/audience-manager/user-guide/reference/demdex-calls.html. (22/11/19)

[3] Jan Odvarko. HTTP Archive Viewer [s.l.] Disponible sur : http://www.softwareishard.com/har/viewer/. (22/11/19)

[4] MDN Web Docs. HTTP headers [s.l.n.d] Disponible sur : https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/ (23/11/19)

[5] Wikipédia. Do/ Not/ Track [s.l.n.d] Disponible sur : https://fr.wikipedia.org/wiki/Do_Not_Track (23/11/19)

[6] Ritesh Maheshwari. How LinkedIn used PoPs and RUM to make dynamic content download 25% faster, 25 juin 2014 [s.l.] Disponible sur : https://engineering.linkedin.com/performance/how-linkedin-used-pops-and-rum-make-dynamic-content-do (30/11/19)