LINGI1341 : Réseaux informatiques
Projet 2 : Analyse du site web zalando.be

De Keersmaeker François
7367-1600

***Résumé*—Ce document présente les résultats de l'analyse de différents protocoles internet du site web zalando.be.**

INTRODUCTION

La connaissance des différents mécanismes qui régissent le fonctionnement d'un site internet est primordiale dans la formation d'informaticien. C'est pourquoi, dans le cadre du cours LINGI1341 : Réseaux informatiques de l'EPL, nous avons analysé le fonctionnement de plusieurs sites.

Ce document présente l'analyse du site web zalando.be, qui est un site de vente de vêtements en ligne. Plus précisément, cette analyse portera sur les protocoles DNS, HTTP, TLS et TCP.

I. DNS

A. Adresses IP

Le but de cette première partie est de retrouver les adresses IP correspondant au domaine zalando.be. Celui-ci existe en deux versions, www.zalando.be en néerlandais et fr.zalando.be en français. En utilisant la commande `dig` [2] sous Linux, on peut retrouver les adresses IPv4 et IPv6 de ces domaines. Ces adresses sont reprises dans la table I.

TABLE I
ADRESSES IP DES DOMAINES DE ZALANDO

Domaine	Adresse IPv4	Adresse IPv6
zalando.be	130.211.9.113	2a02:cbf0:20:1:7b6b:4a32:5ee7:aebd
www.zalando.be	23.57.82.93	/
fr.zalando.be	23.57.82.93	/

Les domaines en néerlandais et en français ont la même adresse IP.

Ces adresses sont les mêmes quel que soit le résolveur DNS utilisé. Il n'y a donc pas de partage de charge entre plusieurs serveurs.

Le TTL de ces 2 dernières adresses est initialement de 60 secondes, et il diminue avec le temps quel que soit le résolveur utilisé, excepté les résolveurs de Google (8.8.8.8), où le TTL est fixe à 59 secondes), et Quad9 (9.9.9.9), (où le TTL semble prendre des valeurs aléatoires).

Le domaine fr.zalando.be est un CNAME pour le domaine fr.zalando.be.ipv4.edgekey.net, qui est lui-même un CNAME pour le domaine e10634.b.akamaiedge.net.

B. Serveurs DNS

L'analyse sera maintenant faite uniquement sur le domaine fr.zalando.be.

Le domaine fr.zalando.be est supporté par 8 serveurs DNS, nommés nxb.akamaiedge.net où x est un chiffre de 0 à 7. Ces serveurs sont tous accessibles en IPv4, mais seul n0b.akamaiedge.net est accessible en IPv6. Les adresses de ces serveurs DNS sont les mêmes quel que soit le résolveur DNS utilisé.

Le TTL des serveurs DNS est le même pour tous les serveurs, et est initialement de 3600 secondes (1 heure) et diminue avec le temps, quel que soit le résolveur utilisé, excepté les résolveurs de Google (8.8.8.8), et Quad9 (9.9.9.9), où le TTL semble prendre des valeurs aléatoires.

C. Records additionnels

En utilisant la commande `dig`, des records de type SOA sont parfois apparus dans la section AUTHORITY. Ces records SOA (Start of Authority) représentent le serveur DNS principal. Chaque domaine peut avoir plusieurs serveurs DNS, mais un seul serveur SOA.

II. HTTP

L'analyse HTTP du site a été effectuée par le navigateur Mozilla Firefox, version 63.0.3, avec un cache et des cookies vides, dans deux cas différents différents :

— Protection contre le pistage activée
— Protection contre le pistage désactivée

Le scénario d'utilisation est le suivant : accès au site, choix du français, connexion, ajout d'un article au panier, achat du panier (jusqu'à l'étape de sélection de la carte de paiement).

A. Domaines contactés

Les domaines contactés lors de ce scénario d'utilisation sont différents pour les deux cas.
Protection contre le pistage activée :

— www.zalando.be (domaine de base, en néerlandais)
— fr.zalando.be (domaine de base, en français)
— mosaic0x.ztat.net (serveurs de Zalando utilisés pour stocker du contenu)
— s3.eu-central-1.amazonaws.com (serveur contenant mosaic, stocke du contenu)
— secure-skin.ztat.net (fournit des ressources pour le site)
— w.usabilla.com (site tiers permettant le feed-back par l'utilisateur)
— cf.metrigo.com (service de pistage)

— creativefactory.zalando.be
— a-content-static.ztat.net (fournit des ressources pour le site)
— checkout.payment.zalando.com (domaine utilisé pour la commande)
— card-entry-service.zal-payments.com (domaine utilisé pour la prise en compte de la carte de paiement)

Le cas sans protection contre le pistage a contacté tous les domaines mentionnés ci-dessus, ainsi que les suivants (entre autres) :

— www.google-analytics.com (service d'analyse d'audience)
— www.googletagmanager.com (service de pistage de Google)
— www.googleadservices.com (service de publicité de Google)
— adservice.google.com (service de publicité de Google)
— www.google.com
— www.google.be
— connect.facebook.net
— www.facebook.com
— x.metrigo.com (service de publicité de Zalando)
— zalando-be.nuggad.net et zalando-be-dp.nuggad.net (service de pistage de Zalando)
— x.mathtag.com (service de publicité et de pistage)
— x.x.doubleclick.net (service de publicité et de pistage développé par Google)
— match.adsrvr.org (service de pistage)
— pixel.rubiconproject.com (service de publicité)
— Encore beaucoup d'autres services de publicité et de pistage

On peut en déduire que seuls les domaines contactés lors du premier cas sont nécessaires au bon fonctionnement du site, et que tous les domaines contactés en plus lors du second cas sont utilisés pour la publicité et le pistage. On remarque en effet que de nombreuses requêtes envers ces domaines concernent des images de 1x1 pixel qui servent à traquer l'utilisateur. Cependant, même dans le premier cas, on retrouve quelques unes de ces images de pistage.

Une requête envoyée au domaine ads.bluelithium.com, un service de pub de Yahoo !, retourne un code 502 (voir figure 1), ce qui représente une erreur du côté du serveur. Cette erreur apparaît à chaque chargement de la page, ce qui est normal puisque le service a été stoppé en 2007. Il est intéressant de remarquer que la requête est toujours envoyée lors de la navigation sur ce site alors que le serveur distant n'existe plus.

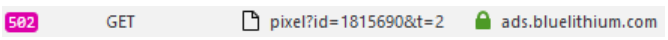

FIGURE 1. Requête HTTP vers le domaine bluelithium

B. Ressources utilisées

L'outil en ligne *HTTP Archive Viewer 2.0.17* [8] a été utilisé pour analyser l'archive HTTP (HAR) provenant de Firefox, avec et sans protection contre le pistage. La figure 2 représente les ressources utilisées par la page pour les deux cas.

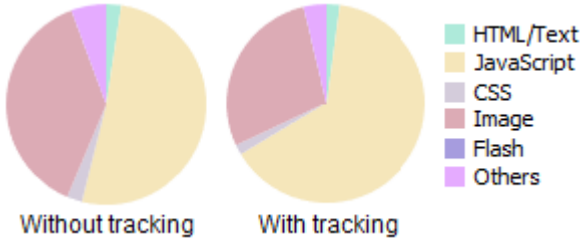

FIGURE 2. Ressources utilisées par la page

Avec la protection contre le pistage activée, on remarque que la majorité des ressources utilisées sont du code JavaScript, utilisé pour le bon fonctionnement du site, et des images utilisées pour l'apparence du site.

Les ressources utilisées lorsque la protection contre le pistage est désactivée sont relativement semblables, mais la portion de ressources JavaScript est plus grande. Cela est dû aux scripts utilisés pour la publicité et le pistage, qui sont absents dans le premier cas.

C. Ports TCP

Le seul port utilisé, et ce pour les deux cas, est le port 443. Puisque ce port est le port par défaut des requêtes en HTTPS, et que toutes les requêtes en sont, il est normal que les requêtes utilisent toutes ce port. Il existe aussi des requêtes pour des fichiers qui avaient été mis en cache ; ces fichiers ont été directement récupérés sans passer par le réseau, et ne sont donc pas passés par le port 443.

D. Analyse des requêtes

Les requêtes seront désormais envoyées avec deux navigateurs différents pour comparer leurs différences : Mozilla Firefox (navigateur par défaut) et Google Chrome (navigateur jamais utilisé).

Tout d'abord que, puisque Firefox est le navigateur par défaut et a déjà visité le site, il possède déjà des cookies pour ce site et peut les envoyer dès la première requête. Sur Chrome, le domaine www.zalando.be a envoyé un header `set-cookie` pour paramétrer un cookie pour ce client. Cela implique que la page affichée en appelant l'URL zalando.be n'est pas la même sur les deux navigateurs :

— Firefox : page en français, avec mon compte déjà connecté, sur le catalogue des hommes
— Chrome : page en néerlandais, aucun compte connecté, aucun catalogue spécifique

Le cookie de Firefox avait donc enregistré toutes les préférences pour ce site, tandis que Chrome n'avait aucune information.

Les requêtes vers les domaines de Zalando contiennent certaines en-têtes non-standard, dont la signification peut être trouvée grâce aux *MDN Web Docs* [6] :

— `Accept` : indique quels types de contenu le client peut interpréter.

— `Accept-Encoding` : définit les compressions que le client peut interpréter.
— `Accept-Language` : langues préférées par le client.
— `Cache-Control` : paramètres de cache.
— `DNT` : définit si le client préfère autoriser son suivi (0) ou pas (1).
— `Pragma` : utilisé pour la compatibilité avec HTTP/1.0.
— `TE` : indique l'encodage des transferts.
— `Upgrade-Insecure-Requests` : indique que le client peut accéder au site sécurisé.

E. Analyse des réponses

Les réponses HTML sont soit en HTTP/2.0 pour les réponses provenant des domaines fournissant le contenu du site en lui-même (les serveurs zalando.be et mosaic.0x.ztat.net), soit en HTTP/1.1 pour les services de publicité et de pistage.

Les réponses des domaines de Zalando contiennent des entêtes non-standard, dont la signification peut être trouvée grâce aux *MDN Web Docs* [6] :

— `Access-Control-Allow-Origin` : indique si les ressources peuvent être partagées avec une origine donnée.
— `etag` : indique la version de la ressource. Le client n'aura pas besoin de la télécharger si il possède déjà la même version.
— `expires` : date/heure à laquelle la réponse expire.
— `X-Firefox-Spdy` : indique le support du protocole de trafic HTTP SPDY.

III. TLS

Le protocole TLS permet de sécuriser les échanges entre le client et le serveur. Lorsque le protocole HTTP est utilisé par dessus le protocole TLS, on dit que le site utilise le protocole HTTPS. Le site zalando.be utilise ce protocole et est donc (théoriquement) sécurisé. L'analyse de TLS a été effectuée en utilisant le navigateur par défaut Mozilla Firefox, avec la protection contre le pistage désactivée (de cette façon, il est possible de comparer la sécurité des services du site en lui-même avec celle des services de publicité et de pistage).

Avant toute autre analyse, on regarde si le site peut être accédé en HTTP non sécurisé, en tapant l'adresse `http://www.zalando.be` dans la barre d'adresse. Cette adresse est accessible mais redirige directement vers la version sécurisée du site, `https://www.zalando.be`. Il est donc impossible d'accéder à une version du site n'utilisant pas HTTPS.

Les outils de développement web de Firefox ont été utilisés pour analyser le fonctionnement de TLS sur le site. La version de TLS utilisée est la version 1.2, comme la grande majorité des sites utilisant HTTPS.

A. Suites de chiffrement

La suite de chiffrement utilisée par les services de Zalando fournissant du contenu utile est `TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384`.
Celle utilisée par les services de paiement est `TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256`.
L'algorithme d'échange des clés pour ces deux suites est `ECDHE_RSA`, un échange de clés RSA utilisant l'algorithme de Diffie-Hellman sur une courbe elliptique. Cet algorithme possède la propriété de *Perfect Forward Secrecy*[1] (PFS). L'avantage de cet algorithme, par rapport à un Diffie-Hellman classique, est que l'exécution est beaucoup plus rapide.

La suite de chiffrement utilisée par les services de publicité et de pistage varie en fonction du site. Par exemple, la suite est `TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256` pour le domaine www.google-analytics.com, ou `TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256` pour le domaine adservice.google.com. Ce dernier utilise l'algorithme d'échange des clés `ECDHE_ECDSA`, un échange de clés utilisant l'algorithme de signature ECDSA et l'algorithme d'échange de clés de Diffie-Hellman sur une courbe elliptique. ECDSA utilise aussi les courbes elliptiques, et a deux avantages par rapport à RSA : des longueurs de clés plus courtes et des opérations plus rapides. `ECDHE_ECDSA` possède la propriété PFS.

La commande OpenSSL [9] `s_client -connect fr.zalando.be:443` nous donne les mêmes informations quant à la suite de chiffrement des services de Zalando.

B. HTTP Strict Transport Security

HTTP Strict Transport Security (HSTS) est un mécanisme qui permet aux sites d'indiquer qu'ils ne peuvent être rejoints qu'en HTTPS, et donc d'éviter des attaques diverses.

HSTS n'est pas activé sur le site zalando.be. C'est un problème de sécurité car cela le rend plus vulnérable aux attaques, notamment de type *man-in-the-middle*.

HSTS est activé pour les domaines www.google.com et www.google-analytics.com, ce qui est compréhensible puisque Google est à la pointe en termes de sécurité informatique. Il est cependant assez ironique de constater qu'un service de pistage soit plus sécurisé que le site de base.

C. Certificats

Les certificats de sécurité ont été émis à Zalando SE par l'organisation DigiCert Inc., une entreprise tierce de confiance, habilitée à délivrer des certificats. Elle a notamment délivré des certificats à des géants comme Facebook ou PayPal, on peut donc supposer qu'il s'agit d'une entreprise de confiance.

D. Réutilisation des clés

On veut maintenant savoir si le serveur garde en cache les clés choisies pendant un certain temps, ce qui permettrait aux clients de se reconnecter rapidement en utilisant les mêmes clés.

La commande OpenSSL `s_client -connect fr.zalando.be:443 -reconnect` permet de se connecter plusieurs fois au serveurs à la suite. On remarque la

1. Un algorithme possède la propriété de *Perfect Forward Secrecy* si les clés utilisées lors des anciennes sessions ne peuvent pas être retrouvées, même si un hacker parvient à trouver la clé privée du serveur. Cette propriété est importante pour la sécurité des échanges TLS.

ligne montrée sur la figure 3 dans la réponse, ce qui signifie que les clés négociées sont stockées par le serveur pendant un court laps de temps et peuvent être réutilisées pour réouvrir un connexion avec le même client.

```
Reused, TLSv1.2, Cipher is ECDHE-RSA-AES256-GCM-SHA384
```

FIGURE 3. Ligne indiquant la réutilisation des clés

E. Paquets échangés

Le logiciel de capture *Wireshark* [7] a été utilisé pour analyser les paquets échangés entre le client et le serveur. On retrouve bien les message `ClientHello` et `ServerHello` utilisés pour négocier les options de connexion, par exemple les suites de chiffrement. Cette dernière a été annoncée comme étant `TLS_ECDHE_RSE_WITH_AES_256_SHA384`, ce qui correspond à ce qui avait été trouvé avec les outils de développement web de Firefox.

IV. TCP

Transmission Control Protocol (TCP) est un protocole de transport de données fiable et nécessitant une connexion entre les deux appareils, utilisé pour plus de 90% du trafic de paquets sur Internet. Les protocoles HTTP et TLS analysés ci-dessus sont en fait utilisés par dessus TCP.

A. Connexions simultanées

Pour augmenter la vitesse de la connexion, certains navigateurs effectuent plusieurs connexions au serveur en parallèle. Une capture des paquets de début de connexion (paquets SYN) avec *Wireshark* [7] permet de voir que le nombre de connexions simultanées peut monter jusque 6, ce qui est la valeur maximale pour Firefox, le navigateur utilisé.

B. Options négociées

Lors de la connexion entre le client et le serveur, certaines options améliorant l'utilisation de TCP sont négociées. Si le client et le serveur sont compatibles avec ces options, elles sont utilisées. On peut retrouver ces options dans les paquets SYN et SYN+ACK du début de connexion.

Une capture de paquets avec *Wireshark* [7] permet de voir que les options négociées entre le client (navigateur Mozilla Firefox 63.0.3) et le serveur (fr.zalando.be) sont :

- Maximum segment size : le navigateur annonce 1460 bytes, et le serveur annonce 1452 bytes. C'est la valeur la plus faible qui est choisie.
- SACK permitted : permet l'utilisation d'acquis sélectifs.
- Timestamps : permet de connaître précisément l'ordre des paquets. Lorsqu'un paquet de données est envoyé, il a un timestamp spécifique (dans son champ `TSval`), et le paquet ACK de ce paquet de données contient le même timestamp (dans son champ `TSecr`).
- Window scale : permet d'utiliser des fenêtres de plus de 65535 bytes, en décalant le nombre indiquant la taille de la fenêtre de S bits vers la droite, où S est le *scaling factor*. Le navigateur et le serveur annoncent tous les deux $S = 7$.

L'utilité des options Timestamps et Window scale est expliquée dans *RFC7323* [10].

En utilisant l'outil *Scapy* [11], qui permet de créer ses propres paquets et de les envoyer à la destination voulue, il est possible d'analyser l'utilisation de deux autres options :

- TCP Fast Open : permet d'envoyer des données dès le paquet de début de connexion. Un paquet contenant cette option est envoyé et on regarde si le paquet de réponse contient également l'option. Pour fr.zalando.be, l'option n'est pas supportée.
- TCP Explicit Congestion Notification : permet de marquer les paquets s'ils ont connu de la congestion lors de la transmission. Cette option n'est pas négociée en la plaçant dans les options du paquet, mais en mettant les flags CWR et ECN-Echo à 1. Si le paquet de réponse a aussi ces flags mis à 1, l'option est activée pour la transmission. Pour fr.zalando.be, l'option n'est pas supportée.

C. Flags additionnels

La capture de paquets via *Wireshark* [7] a mis en évidence un flag peu utilisé : PSH. Ce flag est mis à 1 lorsque l'hôte n'a pas attendu d'avoir un paquet contenant MSS bytes pour l'envoyer. Ce flag indique que le paquet a été envoyé directement, et doit être transmis directement à l'application.

D. Round-trip time

Le *round-trip time* (RTT) indique la durée entre l'envoi d'un paquet de données et la réception du ACK correspondant. La capture de paquets via *Wireshark* [7] permet de calculer le RTT. Celui-ci n'est pas constant et varie autour de 0.01 ms. La figure 4 illustre un paquet de données et l'ACK correspondant, pour lesquels le RTT vaut 0.014 ms (le champ Time indique les secondes écoulées depuis la capture du dernier paquet).

No.	Time	Source	Destination	Protocol	Info
2029	0.002438650	23.61.5.250	192.168.1.30	TLSv1.2	Application Data,
2030	0.000014386	192.168.1.30	23.61.5.250	TCP	36604 → 443 [ACK]

FIGURE 4. Paquets dont le RTT vaut 0.014 ms

CONCLUSION

L'analyse du site zalando.be a permis de découvrir le fonctionnement des protocoles utilisés lors de transmission de paquets sur le réseau. Les découvertes les plus intéressantes sont les suivantes :

- Il y a énormément de services de publicité et de pistage sur les sites internet, et ces derniers utilisent des images invisibles de 1x1 pixel pour traquer l'utilisateur.
- Les cookies permettent de lancer une page web calibrée aux habitudes de l'utilisateur.
- Il existe de nombreuses en-têtes non standard dans les requêtes et réponses HTTP.
- Les sites internet ne sont pas toujours à la pointe en matière de sécurité.
- Il est possible d'établir plusieurs connexions simultanées vers le même domaine pour accélérer le téléchargement.

RÉFÉRENCES

[1] O. Bonaventure, *Computer Networking : Principles, Protocols and Practice, 2nd edition*, Université Catholique de Louvain, 2011 : http://cnp3book.info.ucl.ac.be/2nd/html/index.html

[2] Linux man page of `dig` : https://linux.die.net/man/1/dig

[3] O. Bonaventure, *A first analysis of an HTTP server*, 15 novembre 2018 : https://obonaventure.github.io/cnp3blog/http-analysis/

[4] O. Bonaventure, *A first analysis of a TLS server*, 22 novembre 2018 : https://obonaventure.github.io/cnp3blog/tls-analysis/

[5] O. Bonaventure, *A first analysis of a TCP server*, 28 novembre 2018 : https://obonaventure.github.io/cnp3blog/tcp-analysis/

[6] Mozilla Developer Networks Web Docs (MDN Web Docs) : https://developer.mozilla.org/fr/docs/Web

[7] Wireshark : https://www.wireshark.org/

[8] HTTP Archive Viewer 2.0.17, Jan Odvarko : http://www.softwareishard.com/har/viewer/

[9] OpenSSL : https://www.openssl.org/

[10] D. Borman, B. Braden, V. Jacobson, R. Scheffenegger, *RFC7323 : TCP Extensions for High Performance*, Internet Engineering Task Force (IETF), Septembre 2014 : https://tools.ietf.org/html/rfc7323#section-3

[11] Scapy : https://scapy.readthedocs.io/en/latest/index.html