LINFO1341 - Réseaux Informatiques 2024

Victor Carballes - 34472100
Krystian Targonski - 42942000

***Abstract*—Le but de ce projet est d'analyser le trafic réseau généré lors de l'utilisation du service de stockage en ligne de Microsoft, OneDrive. Pour ce faire, nous allons comparer deux cas : l'utilisation de l'application fournie par Microsoft et l'utilisation du site web.**

I. Introduction

OneDrive est un service de stockage en ligne proposé par Microsoft. Il permet aux utilisateurs de sauvegarder, synchroniser et partager des fichiers et des dossiers via le cloud. Il offre un espace de stockage en ligne accessible depuis n'importe quel appareil connecté à Internet. Ce service est disponible sous plusieurs formes, que ce soit par un accès simple via un navigateur ou par une application (mobile ou non). C'est pourquoi nous avons également décidé de nous pencher sur les potentielles différences entre les requêtes que ce service pourrait effectuer en fonction de l'endroit où il est utilisé.

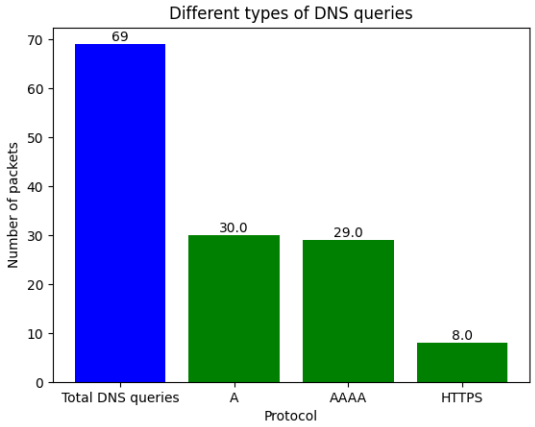
II. Procédé d'analyse

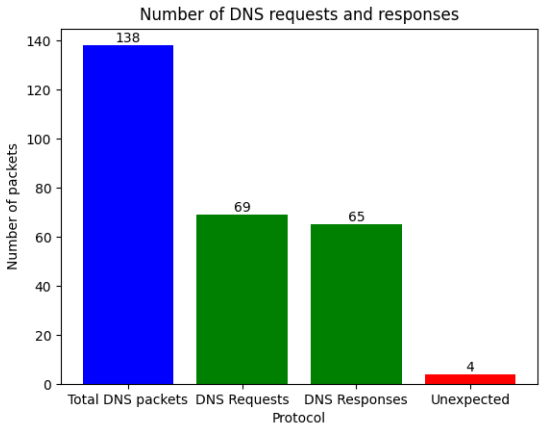
- Application Windows 11 (Par Wifi UCL et maison)
  - Routine de synchronisation (Syncing)
  - Ajout d'un fichier aux répertoires
  - Suppression permanente d'un fichier
  - Téléchargement d'un fichier (Par requête d'ouverture du fichier)
  - Garder le fichier sur l'ordinateur (Option 'Keep On This Device')
  - Relâcher le fichier de l'ordinateur (Option 'Free Up Space')
  - Re-synchronisation d'un grand fichier (Changement d'une partie du contenu)
  - Partage de fichier (Par mail)
  - Terminer le processus de l'application (Par le gestionnaire de tâches)
  - Démarrer l'application
- Site web
  - Connexion de l'utilisateur
  - Chargement de la page d'accueil
  - Chargement d'un dossier partagé
  - Chargement d'un fichier en ligne
  - Téléchargement d'un fichier (sans son ouverture)
  - Téléchargement d'un fichier déjà ouvert
  - Déconnexion de l'utilisateur
  - Partage d'un fichier
  - Ajout d'un fichier par un autre utilisateur
  - Changement du nom d'un fichier par un autre utilisateur
  - Suppression d'un fichier par un autre

III. Analyse DNS

Lors de nos différentes captures, nous avons remarqué, en utilisant les commandes whois et dig, que les requêtes sont principalement dirigées vers des serveurs de Microsoft, avec des exceptions telles que Akamai Technologies. En ce qui concerne les temps de réponse, en moyenne, nous avons observé environ 32 ms, avec des pics allant de 19 ms à 78 ms, et des temps TTL moyens de 198 secondes. De plus, le comportement était généralement prévisible, à l'exception d'un cas qui nous a surpris lors de la résolution effectuée par l'application (voir la suite pour plus de détails). Étant donné leur nombre assez important, nous allons analyser en détail les requêtes les plus fréquentes, tandis que les autres seront répertoriées dans un tableau à la fin du rapport.

Voici quelques graphes pour illustrer l'entièreté des requêtes DNS faites par l'application sur Windows ainsi que leurs types:





A. Analyse Application

Les requêtes DNS sont toutes effectuées à l'initialisation/destruction de l'application, lors de la demande de partage de fichiers ou de téléchargement de fichiers, ainsi que lors de certaines synchronisations (lors de l'expiration de la connexion ou du changement d'IP/port). En tout, l'application a effectué 88 demandes/réponses vers des domaines différents. Nous avons tout de même trouvé des instances de demandes envoyées par HTTPS, ce qui nous a un peu surpris. Cependant, les demandes étaient de deux types, A et AAAA. D'après ce que nous avons pu trouver sur ce sujet, il s'agit d'un protocole nommé DoH (DNS over HTTPS), principalement utilisé pour améliorer la confidentialité des requêtes DNS, qui, par défaut, sont envoyées en clair. Un comportement moins fréquent sur le Web que sur l'application est le fait que cette dernière favorise vraiment la communication à travers les adresses IPv6 obtenues, alors que le service Web, bien qu'ayant également une préférence pour l'IPv6, utilise quand même plus d'adresses IPv4 résolues.

Voici quelques noms de domaines notables :

- doh.opendns.com : Ce domaine est résolu lors du partage d'un fichier sur l'application, généralement à plusieurs reprises avec une alternance de types A et HTTPS. Cependant, une surprise est survenue avec ce nom de domaine ! En effet, lorsque nous avons capturé nos paquets avec Wireshark, celui-ci nous a montré un champ d'autorité à 0, indiquant un manque de serveur autoritatif. Cependant, d'après nos tests en local avec la commande:

  ```
  dig doh.opendns.com -t HTTPS
  ```

  Nous avons reçu une réponse avec le flag autorité mis à 1 et le serveur autoritatif auth1.opendns.com apparaît. Cela peut être dû à des erreurs liées au fonctionnement de Wireshark.
- www.tm.ak.prd.aadg.trafficmanager.net : Ce nom de domaine était résolu à chaque fois que nous accédions à un dossier ou à un fichier dans le cloud. Le serveur correspondant à ce nom de domaine appartient à Microsoft et n'a pas de serveur autoritatif. Une particularité des résolutions de ce nom de domaine est le fait que nous n'avons jamais vu de type A, seulement des résolutions pour des adresses IPv6.

B. Analyse Web

D'un point de vue des requêtes effectuées par le navigateur Internet lors de l'accès et de la connexion au service OneDrive à travers nos comptes UCL, nous avons compté un total de 64 demandes qui ont été faites, soit un total de 128 paquets DNS échangés (2 demandes ont retourné des erreurs car le nom demandé n'existait pas). Dans ce cas, nous n'avons observé que des demandes de résolution de type A et AAAA, rien d'autre.

- uclouvain-my.sharepoint.com : Pour ce nom de domaine, deux types de requêtes sont effectuées : une pour l'adresse IPv4 (type A) et l'autre pour l'adresse IPv6 (type AAAA). La résolution de ce domaine se fait toujours dans les cas suivants : lors de la première connexion, lorsque notre navigateur doit étendre notre session (dans les cas d'inactivité prolongée) et lorsque la page est rafraîchie. Ce nom de domaine appartient à Microsoft, c'est ce que nous indique la commande whois avec l'adresse retournée par la résolution. Un point assez intéressant à noter est que même si les deux types d'adresses sont résolus, le service utilise majoritairement (voire totalement) l'adresse IPv6 pour communiquer avec le serveur concerné.
- res-1.cdn.office.net : Ce nom de domaine nous a paru assez particulier car, d'après nos premières analyses, il y avait des requêtes qui étaient faites pour connaître son adresse IPv4 (A), mais cette information n'était jamais vraiment utilisée. Ce nom de domaine appartient à Akamai Technologies, une entreprise qui fournit des services de cloud. Cette information nous a poussés à garder un oeil sur ce domaine et, au final, nous avons découvert qu'il y a en effet de la communication entre notre appareil et le serveur en question lors du chargement de fichiers ! C'est donc un domaine qui est résolu assez souvent plus on utilise le service.

IV. Couche réseau

Un comportement efficace observé dans les réseaux domestiques est l'omission de l'IPv4, car dans la plupart des cas, elle n'est pas utilisée par l'application pour éviter les problèmes liés aux NAT. Cependant, des paquets sont parfois envoyés via IPv4, notamment lors de l'utilisation de TLSv1.2. En IPv4, l'application et le service web utilisent la technique de UPnP (Universal Plug and Play) pour détecter automatiquement les objets connectables et établir des politiques de connexion avec les routeurs. De plus, ils accèdent toujours à des adresses spécifiques, principalement pour la synchronisation, mais également lors du démarrage / connexion. Pour les échanges de données, ils utilisent exclusivement TLSv1.2 et tous les paquets TCP sont configurés en mode No Fragment, avec le Checksum désactivé, ce qui assure une transmission fluide même en cas d'échec TCP.

Voici un tableau répertoriant les adresses IP vers lesquelles les paquets sont envoyés:

IP	Position	ISP
20.42.73.28:443	USA	Mcr. corp.
20.199.120.182:443	France	Mcr. corp.
20.135.20.1:443	Irlande	Mcr. Azure
20.223.35.26:443	Irlande	Mcr. Azure
40.79.167.35:443	Japon	Mcr. corp.
20.44.10.123:443	USA	Mcr. corp.
52.138.229.66	Italie	Mcr. corp.
1drv.ms (13.107.42.12:443)	USA	Mcr. corp.
$[2603:1026:2400:0:0:0:0:2]:443$	France	Mcr. corp.
$[2620:1ec:42:0:0:0:0:132]:443$	France	Mcr. corp.
$[2603:1026:3000:c8::7]$	Italie	Mcr. corp.
$[2620:1ec:8f8::10]$	USA	Mcr. corp.

Dans l'ensemble de ces adresses, une tendance très spécifique émerge : toutes les destinations sont des serveurs Microsoft (et parfois Microsoft Azure), la plupart étant approvisionnées par Microsoft en tant que fournisseur de services internet, que ce soit dans l'application ou sur le service web. De plus, il est remarquable que toutes les adresses pointent vers le port 443, un port standard pour les interfaces HTTP(s). Cette configuration est probablement due à l'utilisation de WebDAV pour le transfert de fichiers via OneDrive.

Comme mentionné, les échanges avec Microsoft se font toujours via le port 443, que ce soit en TCP/TLS ou en UDP/Quic (bien que OneDrive n'utilise pas UDP, seulement Quic). Les adresses IP des serveurs sont variées, mais toutes appartiennent à Microsoft. Les pays fréquemment connectés incluent les Pays-Bas, la France, Washington, la Suisse et l'Irlande ou même encore le Japon. Avec l'Irlande étant connue comme la position du serveur de référence pour nos données d'utilisateur selon le pacte GDPR.

Une adresse qui nous a paru un peu hors du commun était justement celle du Japon. En effet, Wireshark nous montrait qu'il s'agissait bien de paquets provenant de Tokyo, mais whois ne reflétait pas cette information et nous montrait que le serveur correspondant se trouvait aux États-Unis. Ceci était dû au fait que whois ne vérifie pas les ASN. Après une recherche un peu plus poussée, nous avons découvert que celui-ci appartenait à l'AS8075, géré par Microsoft.

V. Couche transport

Ici, on s'intéresse aux protocoles qui sont utilisés dans la couche transport. Les voici :

Protocol	Utilisation
TCP (RFC 9293)	Utilisé pour l'établissement de la connexion et l'envoi des ACK et SYN dans le cadre des divers paquets TLS, puisque TLS est basé sur TCP.
TLSv1.2/1.3 (et réponses en TLSv1.0 dans certains cas) (RFC 8446)	Transmission de données en général, cryptée par (de préférence) TLS_AES_256_GCM_SHA384.
QUIC (UDP) (RFC 9000)	Échange de clés et d'identifiants de connexion.
MDNS (Multicast Domain Name Service) (RFC 6763)	Service utilisé pour découvrir des imprimantes sur Windows 10, mais aussi les services DNS par la suite.

Lors de cette analyse, nous nous sommes posé la question de la quantité de connexions vers un nom de domaine, notamment dans le cas où plusieurs actions sont effectuées simultanément. Il est ainsi intéressant de déterminer s'il existe plusieurs connexions vers un même nom de domaine et d'en comprendre les raisons. Dans notre cas, il a été observé plusieurs connexions TLSv1.3 vers microsoft.com, impliquant diverses opérations telles que la connexion, la transmission de données, ou encore la gestion des événements et des paramètres.

Nous avons aussi eu l'occasion de voir du trafic QUIC passer, ce qui soulève des questions importantes telles que quelles versions sont utilisées ou encore quelles extensions sont négociées dans le handshake. En réponse, nous avons constaté que la seule version de QUIC utilisée est la version 1. Les extensions identifiées dans le handshake comprennent Initial, 0-RTT (incompatible avec Protected), Handshake, et Protected (TLSv1.3/TCP). De plus, il est à noter que les paquets QUIC contiennent souvent des éléments tels que PADDING, CRYPTO et PING après le payload, ce qui apporte des informations supplémentaires sur la nature de la communication en cours.

VI. Chiffrement et sécurité

D'un point de vue de sécurité, comme mentionné précédemment dans la partie DNS, la majorité des requêtes DNS sont effectuées sans encryption / sans protection, à l'exception de celles qui passent par HTTPS, ces dernières étant encryptées, ce qui ajoute une couche de sécurité. De plus, il y a vérification des empreintes des sites grâce à Let's Encrypt.

En ce qui concerne le transfert de données, nous avons observé que TLSv1.2 est principalement utilisé pour le transfert de fichiers, tandis que TLSv1.3 est privilégié pour les connexions à Microsoft Azure et pour QUIC.

Nous avons également vérifié la durée de vie des différents certificats, qui varie de 103 à 128 secondes, et nous avons dressé une liste des algorithmes de chiffrement utilisés, que voici :

Suite de chiffrement	Échange de Clefs	Chiffrage/Hash
AES_256_GCM_SHA384	ECDHE_ECDSA	AES_256_GCM_SHA384
AES_128_GCM_SHA256	ECDHE_ECDSA	AES_128_GCM_SHA256
AES_256_GCM_SHA384	ECDHE_RSA	AES_256_GCM_SHA384
AES_128_GCM_SHA256	ECDHE_RSA	AES_128_GCM_SHA256
AES_256_CBC_SHA384	ECDHE_ECDSA	AES_256_CBC_SHA384
AES_128_CBC_SHA256	ECDHE_ECDSA	AES_128_CBC_SHA256
AES_256_CBC_SHA384	ECDHE_RSA	AES_256_CBC_SHA384
AES_128_CBC_SHA256	ECDHE_RSA	AES_128_CBC_SHA256
AES_256_CBC_SHA	ECDHE_ECDSA	AES_256_CBC_SHA
AES_128_CBC_SHA	ECDHE_ECDSA	AES_128_CBC_SHA
AES_256_CBC_SHA	ECDHE_RSA	AES_256_CBC_SHA
AES_128_CBC_SHA	ECDHE_RSA	AES_128_CBC_SHA
AES_256_GCM_SHA384	RSA	AES_256_GCM_SHA384
AES_128_GCM_SHA256	RSA	AES_128_GCM_SHA256
AES_256_CBC_SHA256	RSA	AES_256_CBC_SHA256
AES_128_CBC_SHA256	RSA	AES_128_CBC_SHA256
AES_256_CBC_SHA	RSA	AES_256_CBC_SHA
AES_128_CBC_SHA	RSA	AES_128_CBC_SHA

Enfin, en ce qui concerne le trafic UDP, un examen approfondi a été effectué pour évaluer son niveau de sécurité. Il a été constaté que le trafic UDP semble être sécurisé via QUIC, qui utilise TLSv1.3 pour garantir la confidentialité et l'intégrité des données échangées. Tout ceci est valable aussi bien pour l'application que pour le service sur le navigateur web.

VII. Application

Ici, nous avons observé le comportement du service lors du transfert de nouveaux fichiers par rapport à la modification de fichiers existants. Nous avons remarqué une séquence de synchronisation en premier lieu, suivie de la transmission incomplète des modifications, celles-ci étant seulement partiellement transmises à la fin du fichier. Notons également

que la synchronisation se fait automatiquement pour tous les utilisateurs et que les modifications génèrent des synchronisations plus rapides étant donné que la quantité de données à transmettre est plus basse.

Quant au volume de données échangées pour chacune des fonctionnalités, nous avons répertorié cela dans le tableau suivant :

Action	Rate (Peaks)
Routine de synchronisation	3 - 60 [#/sec]
Ajout d'un fichier aux répertoires	864 [#/sec]
Suppression permanente d'un fichier	17 [#/sec]
Téléchargement d'un fichier	2031 [#/sec]
Garder le fichier sur l'ordinateur	650 [#/sec]
Relâcher le fichier de l'ordinateur	3 [#/sec]
Re-synchronisation d'un grand fichier	25 [#/sec]
Partage de fichier	1284 [#/sec]
Terminer le processus de l'application	75 [#/sec]
Démarrer l'application	175 [#/sec]
Connexion de l'utilisateur	106 [#/sec]
Chargement de la page d'accueil	285 [#/sec]
Chargement d'un dossier partagé	123 [#/sec]
Chargement d'un fichier en ligne	247 [#/sec]
Téléchargement d'un fichier (sans son ouverture)	119 [#/sec]
Téléchargement d'un fichier déjà ouvert	96 [#/sec]
Déconnexion de l'utilisateur	85 [#/sec]
Partage d'un fichier	64 [#/sec]
Ajout d'un fichier par un autre utilisateur	33 [#/sec]
Rennomage du fichier par un autre	61 [#/sec]
Supression d'un fichier par un autre	53 [#/sec]

VIII. Appendice

Un point qui nous a paru intéressant à vérifier était la présence éventuelle de serveurs relais, et nous avons remarqué que dans le cas de OneDrive, l'application peut communiquer directement avec les serveurs de Microsoft et ses services tels qu'Azure. De plus, lorsqu'il s'agit d'interagir avec un utilisateur sur le même réseau Wi-Fi ou Ethernet, plusieurs observations sont à noter. Par exemple, nous avons constaté que lorsque plusieurs utilisateurs partagent le même réseau, le nombre de requêtes DNS générées peut être réduit, ce qui peut contribuer à une meilleure efficacité du trafic applicatif.

En ce qui concerne l'utilisation de serveurs relais, il est à noter que les principales destinations utilisées sont souvent les clusters de serveurs de Microsoft, tels qu'Azure, plutôt que des relais intermédiaires. Cette approche favorise une communication directe entre l'application et les serveurs de Microsoft, ce qui peut contribuer à des performances optimales et à une meilleure intégrité des données.

De plus, Microsoft pourrait avoir mis en place des mécanismes de mise en cache de l'information, ce qui pourrait expliquer la rapidité des résolutions DNS observées, notamment à l'UCL. Ces mécanismes pourraient inclure l'utilisation de Multicast DNS (MDNS) pour récupérer localement les informations pertinentes et réduire ainsi la dépendance aux requêtes DNS externes.

En outre, il est plausible que Microsoft ait implémenté des stratégies internes pour faciliter la communication entre ses propres services, en gardant en mémoire les destinations pertinentes et en minimisant ainsi la latence et la dépendance à l'égard de serveurs relais externes.

Tableau des différents noms de domaines résolus :

Nom de domaine	Serveur autoritatif	Entreprise
uclouvain-my.sharepoint.com	sharepoint.com	Microsoft Corp.
shell.cdn.office.net	akamaiedge.net	Akamai Technologies
www.tm.ak.prd.aadg.trafficmanager.net	trafficmanager.net	Microsoft Corp.
login.microsoftonline.com	msidentity.com	Microsoft Corp.
login.msa.msidentity.com	msidentity.com	Microsoft Corp.
www.tm.lg.prod.aadmsa.akadns.net	akadns.net	Akamai Technologies
www.a.lg.prod.aadmsa.akadns.net	akadns.net	Akamai Technologies
prdv4a.aadg.msidentity.com	msidentity.com	Microsoft Corp.
login.live.com	live.com	Microsoft Corp.
clo.footprintdns.com	footprintdns.com (Domain Administartor)	Microsoft Corp.
fp-afdx-bpdee4gtg6frejfd.z01.azurefd.net	azurefd.net	Microsoft Corp.
star-azurefd-prod.trafficmanager.net	trafficmanager.net	Microsoft Corp.
part-0039.t-0009.t-msedge.net	msedge.net	Microsoft Corp.
ams03pap001.storage.live.com	live.com	Microsoft Corp.
ecs.office.com	office.com	Microsoft Corp.
api.onedrive.com	onedrive.com	Microsoft Corp.
common-afdrk.fe.1drv.com	1drv.com	Microsoft Corp.
roaming.officeapps.live.com	live.com	Microsoft Corp.
arc.msn.com	msn.com	Microsoft Corp.
g.live.com	live.com	Microsoft Corp.
displaycatalog.mp.microsoft.com	microsoft.com	Microsoft Corp.
oneclient.sfx.ms	sfx.ms	Microsoft Corp.
skydrive.wns.windows.com	windows.com	Microsoft Corp.
self.events.data.microsoft.com	microsoft.com	Microsoft Corp.
doh.opendns.com	opendns.com	Cisco Technology, Inc.
skyapi.live.net	live.net	Microsoft Corp.
x1.c.lencr.org	lencr.org	Let's Encrypt (CloudFlare, Inc.)

IX. Références

1) DoH - DNS over HTTPS, RFC 8484
2) Cisco IP Journal Issue 7-3
3) QUIC, RFC 9000
4) TLSv1.3 - RFC 8446
5) mDNS - Multicast DNS, RFC 6762
6) Lien au git contenant les différentes traces et code utilisé: https://github.com/Seito1090/LINFO1341