LINFO1341 Projet 1 : Analyse de Microsoft Teams

Nicolas BOUREZ
EPL
UCLouvain
Louvain-la-Neuve, Belgium
2846 2000

Martin BRANS
EPL
UCLouvain
Louvain-la-Neuve, Belgium
3829 2000

***Abstract*—Ce document reprend les résultats de notre analyse des protocoles utilisées par l'application Microsoft Teams en fonction des fonctionnalités proposées par celle-ci.**

***Index Terms*—computer networks, video-conference, Microsoft Teams**

I. Introduction

Les logiciels de vidéo-conférence ont connu un grand essor dans leur développement grâce à la pandémie de COVID-19 à partir de 2020 et ont impacté de manière significative son ressenti au quotidien des gens. Suite à la mise en place du confinement dans de nombreux pays, ils ont pu permettre aux gens de communiquer entre eux, aux entreprises de promouvoir le télé-travail et aux écoles et universités de poursuivre les cours à distance. Dans le cadre du Projet 1 du cours de Réseaux Informatiques de l'EPL, nous avons analysé le logiciel de vidéo-conférence Microsoft Teams, et plus particulièrement la façon dont cette application communique via Internet afin de mettre en oeuvre ses différentes fonctionnalités. Nous discuterons tout d'abord des différents serveurs contactés durant son utilisation via une analyse DNS. Nous regarderons ensuite les couches réseaux et transport qui servent à transmettre l'information communiquée. Enfin, nous parlerons également des techniques de chiffrement et de la sécurité de l'application, avant de conclure sur son comportement lors de situations spécifiques.

II. Analyse DNS

Tout d'abord, nous avons remarqué que les domaines résolus différaient fortement en fonction de la fonctionnalité utilisée. Nous avons donc analysé les 5 situations d'utilisation de Microsoft Teams qui nous paraissaient les plus pertinentes, à savoir :

- l'application inactive en arrière-plan
- un échange de messages textuels (avec emotes de réaction)
- un appel audio
- un appel vidéo (avec et sans partage d'écran)
- un échange de fichiers

A. Domaines résolus lorsque l'application est inactive

Lors d'une capture de paquets d'une trentaine de seconde avec simplement l'application ouverte sur notre ordinateur, on peut tout de même observer une centaine de paquets échangés :

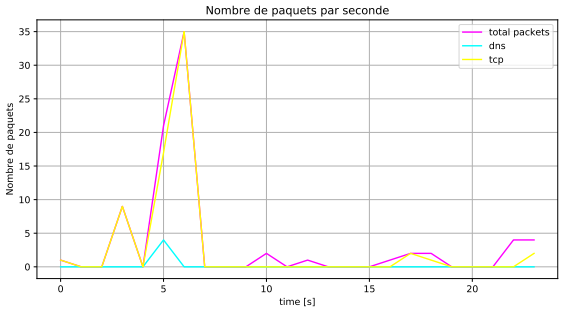

Fig. 1. Paquets capturés lorsqu'aucune fonctionnalité n'est utilisée

On aperçoit une requête DNS suivie d'un important échange de protocoles TCP. Le nom du domaine de cette requête est eu-teams.events.data.microsoft.com. Le serveur autoritaire est *ns1-201.azure-dns.com* et est géré par Microsoft Teams. Ce domaine est utilisé pour collecter des données telles que les interactions de l'utilisateur avec l'application, des informations sur les appareils utilisés, les performances de l'application, les erreurs rencontrées et d'autres données similaires.

B. Domaines résolus lors d'un échange de messages

Le domaine utilisé lors d'un échange de message est emea.ng.msg.teams.microsoft.com. Il est utilisé par Microsoft Teams pour acheminer les messages instantanés et les notifications de l'application, dans le cadre de l'infrastructure de messagerie synchrone de Teams, qui permet aux utilisateurs de communiquer en temps réel.
Ce domaine possède 4 serveurs autoritaires différents dont le domaine autoritaire est *cloudapp.net* :

- ns1-201.azure-dns.org
- ns2-201.azure-dns.com
- ns3-201.azure-dns.info

- ns4-201.azure-dns.net

Après l'analyse de chaque fonctionnalité, on remarquera que ces serveurs et ce domaine autoritaire sont utilisés par beaucoup de domaines de Teams.

C. Domaines résolus lors d'un envoi de fichier

Le domaine utilisé lors d'un envoi de fichier est uclouvain-my.sharepoint.com. Il est utilisé pour accéder au service SharePoint Online de Microsoft, qui est une plateforme de collaboration et de gestion de documents basée sur le cloud. Il permet aux utilisateurs de l'entreprise d'accéder à leur site SharePoint personnel, où ils peuvent stocker, partager et collaborer sur des fichiers et des documents, ainsi qu'utiliser d'autres fonctionnalités de collaboration SharePoint Online. Il possède également 4 serveurs autoritaires et un domaine autoritaire nommé *aa-rt.sharepoint.com* :

- ns1-222.azure-dns.org
- ns2-222.azure-dns.info
- ns3-222.azure-dns.net
- ns4-222.azure-dns.com

D. Domaines résolus lors d'un appel audio et vidéo

Contrairement aux 4 sous-points précédents, on remarque qu'il y a beaucoup plus de requêtes DNS de la part de Teams, comme le montre ce graphique (appel audio d'une cinquantaine de seconde):

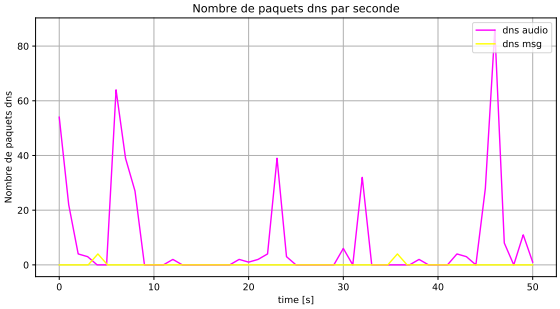

Fig. 2. Comparaison des paquets DNS capturés lors d'un appel audio et d'un échange de messages

Et nous avons alors une plus grande diversité dans les noms de domaines résolus :

1) *api-emea.flightproxy.teams.microsoft.com* : fourni un proxy pour les appels audio et vidéo en temps réel. Le "emea" est utilisé dans le cas où les utilisateurs viennent des régions d'Europe, du Moyen-Orient ou d'Afrique
2) *emea.cc.skype.com* : gère les connexions entre les utilisateurs. Il est également utilisé pour l'application Skype de Microsoft
3) *euaz.relay.teams.microsoft.com* : un relais de communication entre les clients de Teams et les services de Microsoft dans le cloud
4) *teams.microsoft.com* : utilisé pour se connecter à l'interface utilisateur web de Teams
5) *loki.delve.office.com* : utilisé par l'application Delve de Microsoft Office 365 pour stocker et traiter des données de profil utilisateur
6) *eu-mobile.events.data.microsoft.com* : utilisé pour collecter des données d'utilisation et de performance.

Parmis tous ces domaines, deux méritent une attention particulière à savoir le domaine *teams.microsoft.com* et le domaine *euaz.relay.teams.microsoft.com*. Le premier car c'est le domaine principal de Teams et le second car il n'a pas d'adresse IP fixe. L'adresse IP du premier est `52.113.194.132` (obtenue via la commande dig(2) de Linux et confirmée par *WireShark*) et son domaine autoritaire est *s-msedge.net*. La plateforme de ce domaine est *Azure Front Door*, qui est un service de livraison de contenu fourni par Microsoft Azure. Microsoft Azure est une plateforme de cloud de Microsoft.
Le second domaine est assez particulier car on n'obtient pas toujours la même adresse IP pour ce domaine. Cela vient du fait que c'est un CNAME pour le domaine *euaz.relay.teams.trafficmanager.net*. Traffic Manager est est un service Azure qui permet de gérer la distribution du trafic entrant pour améliorer la disponibilité et les performances de l'application. Il va donc distribuer le trafic à différents serveurs en fonction de notre région géographique ou bien de la charge de ceux-ci. Les domaines autoritaires sélectionnés dans notre cas sont :

- westeurope.cloudapp.azure.com
- japaneast.cloudapp.azure.com
- swedencentral.cloudapp.azure.com
- francecentral.cloudapp.azure.com

Des cas similaires ont été observés pour les domaines 5 et 6 qui utilisent aussi Traffic Manager.

De manière plus générale, l'ensemble des domaines est géré par une entreprise appelée MarkMonitor Inc [6]. C'est une société spécialisée dans la sécurité et la fourniture de domaines.

E. Type de requêtes effectuées

La majorité des requêtes sont de type 'A', ce qui signifie qu'elles cherchent à connaître l'adresse IP (IPv4) du serveur par lequel les données vont transiter. Celle de type 'AAAA' sont pour les adresses IPv6.

On remarque également dans les réponses DNS qu'il y a des requêtes de type CNAME. Ces requêtes sont utilisées pour trouver le nom canonique d'un domaine spécifié. Cela signifie que la première requête qu'on peut apercevoir dans Wire Shark n'utilise pas le nom canonique (mais aliasé) et qu'il y a des requêtes intermédiaires pour trouver l'adresse IP.

Les domaines autoritaires utilisent des requêtes de type NS. Ces requêtes sont utilisées pour fournir des informations sur les serveurs de noms autoritaires pour un domaine spécifique.

III. Couches Réseaux

A. Gestion des NATs

De nombreux routeurs à domicile et firewalls utilisent un système de ”traduction d'adresses réseaux” (ou ”NAT”) [1]. Ce dernier ajoute une couche de sécurité supplémentaire en cachant les adresses IPv4 privées des appareils reliés au routeur derrière sa propre adresse IPv6 publique. Le routeur, servant alors d'intermédiaire entre le client et le serveur, peut contrôler les paquets reçus et filtrer le trafic indésirable. Cependant, cela rend également l'adresse publique d'un appareil inconnue de son propre point de vue, alors qu'il est parfois nécessaire de la communiquer au serveur contacté, afin d'ouvrir une connexion TCP, par exemple. Afin de la connaître, le client peut contacter un serveur STUN (pour ”Simple Traversal of UDP through NAT”) avec qui il peut effectuer des tests simples afin de déterminer le type de NAT en vigueur et récupérer son adresse publique. En analysant les traces d'appels audio/vidéo, nous avons pu constater que Microsoft Teams utilise de telles techniques pour contourner les NATs et établir un connexion ”Peer-to-peer” entre les deux clients, ce qui permet une communication plus fluide et donc un appel de meilleure qualité. Des techniques plus complexes telles que TURN sont parfois utilisées, mais plus rares.

B. Analyse des adresses IP

Les adresses (IPv4, donc privées) contactées présentent un pattern général qui peut être identifié grâce à notre analyse DNS. On trouve majoritairement trois types d'adresses participant à la communication :

- Une adresse ayant pour préfixe `192.168.0.0/16`, qui est l'adresse du client.
- Des adresses de serveurs ”cloud” de Microsoft. Ces adresses peuvent avoir de nombreux préfixes différents, les plus récurrents étant `52.64.0.0/10`, `40.64.0.0/10`, `20.0.0.0/10` et `103.0.0.0/10`. Les domaines concernés sont *teams.microsoft.com*, *azure.com* et *cloudapp.net*.
- Dans le cas d'un appel : l'adresse d'un serveur relais appartenant à un fournisseur de services numériques. Ces serveurs font parties de Réseaux de Diffusion de Contenu (”CDN”), spécialisés dans le transfert relayé de grands volumes de données ou de hauts débits. Le préfixe le plus courant est `80.0.0.0/3`, qui appartient au domaine d'*akamai.net*. Notons que dans le cas où l'appel se fait entre deux appareils sur le même réseau Wi-Fi, un serveur relais n'est pas nécessaire et qu'une autre adresse `192.168.0.0/16` apparaît alors à la place : celle du second client.

IV. Couche transport

Pour chaque fonctionnalité de l'application, les protocoles principalement utilisés sont les suivants (dans l'ordre d'importance) :

- Connexion au démarrage : TCP, TLSv1.2, TLSv1.3, UDP
- Conversation textuelle : TCP, TLSv1.2
- Envoi/Téléchargement d'un fichier : TCP, TLSv1.2
- Conversation audio : UDP, TCP, TLSv1.2
- Conversation vidéo : UDP, TCP, TLSv1.2
- Partage d'écran : UDP, TCP, TLSv1.2

Le protocole majoritairement utilisé est donc TCP car il supporte des connexions sécurisées et garantit la bonne réception des paquets envoyés, ce qui est primordial pour l'envoi de messages textes, les échanges avec le cloud ou l'établissement d'une connexion avec un serveur relais. Cependant, UDP est préféré pour les appels audios et vidéos, ainsi que pour le partage d'écran car il est plus rapide au prix d'une fiabilité non garantie. En effet, si on veut avoir une conversation audiovisuelle fluide en temps réel, transmettre les paquets rapidement au fur et à mesure qu'ils sont crées est alors plus important que garantir leur arrivée.

Nous pouvons par ailleurs observer à plusieurs reprises deux ou trois connexions TCP s'établir entre les mêmes adresses. Cela a pour avantage d'agrandir le débit de données entre les deux utilisateurs ou réduire le délai de transmission, bien qu'une seule connexion suffise la plupart du temps.

On peut également observer l'utilisation d'autres protocoles tels que :

- QUIC (Version 1), qui permet d'établir des connexions sécurisées et fiables en UDP, donc plus rapidement qu'en TCP [2]
- RTCP, utilisé pour évaluer la performance d'un appel en faisant des retours en temps réel au serveur, utile pour l'encodage adaptatif et la détection de défauts de transmission [3]
- STUN et TURN, des techniques de contournement de NATs, dont nous avons déjà parlé
- SSDP, qui fait partie de la base du protocole de découverte de ”Universal Plug and Play”, un ensemble de protocoles de mise en réseaux poste à poste [4]

V. Chiffrement et sécurité

A. DNS

L'utilisation DNS ne semblent pas sécurisée par des mécanismes de type DNSSEC (DNS Security Extensions), DBS-over-TLS (DoT) et DNS-over-HTTPS (DoH). Cependant, le flag ”response” des réponses DNS est bien à 1 pour chacune des réponses ce qui indique que la réponse est authentifiée et donc sécurisée.

B. TLS

Les versions TLS utilisées dépendent des fonctionnalités :

	message	fichier	audio	vidéo
Versions	1.2	1.2	1.0 et 1.2	1.0 et 1.2

Les protocoles de transport sécurisés par la version 1.2 sont les suivants (pour un paquet TLS particulier, ici `ClientHello`) :

- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
- TLS_DHE_RSA_WITH_AES_256_GCM_SHA384
- TLS_DHE_RSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
- TLS_RSA_WITH_AES_256_GCM_SHA384
- TLS_RSA_WITH_AES_128_GCM_SHA256
- TLS_RSA_WITH_AES_256_CBC_SHA256
- TLS_RSA_WITH_AES_128_CBC_SHA256
- TLS_RSA_WITH_AES_256_CBC_SHA
- TLS_RSA_WITH_AES_128_CBC_SHA
- TLS_RSA_WITH_3DES_EDE_CBC_SHA

Ce nombre varie en fonction des différents `ClientsHello`. Le protocole sécurisé par la version 1.0 est le suivant :

- TLS_DH_anon_WITH_RC4_128_MD5

C. Certificats

Lors de l'envoi d'un message, on a une connexion client-serveur qui s'établit à l'aide de protocoles TLS. On a remarqué que deux certificats sont échangés durant cette connexion : l'un certifié par une société nommée DigiCert Inc. et l'autre par Microsoft Corporation. DigiCert est une société qui délivrent des certificats de sécurité pour protocole TLS. Le premier certificat est valide environ 1 an tandis que le second est valide 4 ans.

D. Chiffrement

Lors de l'établissement du chiffrement, on remarque que le client propose une vingtaine d'algorithmes de déchiffrement au serveur (algorithmes supportés par le client). Le serveur sélectionne ensuite un des algorithmes parmi ceux proposés par le client, à savoir, dans notre cas, celui-ci :

- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384

Cet algorithme est en réalité une combinaison de plusieurs algorithmes de chiffrement à savoir :

- L'utilisation d'un échange de clé ECDHE
- Un algorithme RSA (Rivest-Shamir-Adleman) qui est un algorithme de chiffrement asymétrique utilisé pour la négociation de la clé et la signature numérique
- Le chiffrement symétrique AES-256 en mode GCM (Galois/Counter mode) et l'algorithme de hachage SHA-384

Il y a également un algorithme utilisé pour l'échange des clés. L'algorithme utilisé pour déchiffrer la clé du serveur et du client se trouve être celui-ci : `rsa_pkcs1_sha256`. Cet algorithme utilise le hachage SHA-256 pour assurer l'intégrité des données échangées entre le client et le serveur. Il utilise également le protocole `PKCS\#1 v1.5` de RSA pour chiffrer la clé de session.

VI. Application

Grâce à l'analyse faite tout au long de ce rapport, on peut émettre l'hypothèse que Teams est en réalité une application majoritairement gérée sur le cloud, dans le sens que, dans la plupart des cas, le programme qui tourne sur l'ordinateur du client ne fait que relayer les informations qui sont ensuite traitées sur le cloud.

A. Comparaison des fonctionnalités

Si on met en vis-à-vis les différents paquets échangés par les différentes fonctionnalités de l'arbre, on peut mettre en place un tableau comparatif :

	Message	Appel audio/vidéo
Protocoles utilisés	DNS, TCP	DNS, TCP, UDP
Volumes de protocoles échangés	Peu de DNS, échange continu par TCP	10x plus de DNS, Connexion par TCP, Echange continu par UDP
Récepteur des paquets	Serveurs Microsoft	Connexion : Serveurs Microsoft, Echange : Application du client

L'utilisation de la webcam augmente considérablement le nombre de paquets échangés (3x plus de paquets TCP échangés lors de l'établissement de la connexion, 5x plus d'échanges de protocoles UDP)

B. Comparaisons des débits

On remarque, via le graphique ci-dessous, que le nombre de paquets échangés par seconde est beaucoup plus élevé pour un appel vidéo qu'un appel audio. Cela vient du fait qu'un appel vidéo demande une meilleure connexion et une nombre plus important de protocoles UDP (transmission de l'image en temps réel, etc...). Le nombre de paquets échangés lors d'un échange de messages est insignifiant par rapport aux deux autres situations, puisque la quantité d'information à transmettre est beaucoup moins grande.

C. Audio vs vidéo

En analysant les paquets échangés lors d'un appel audio et vidéo, on remarque que les courbes ont pratiquement la même forme. Lors de l'établissement de la connexion entre les clients (qui correspond à la période où on attend que le récepteur décroche, d'une durée d'environ 5 à 10 secondes), on remarque

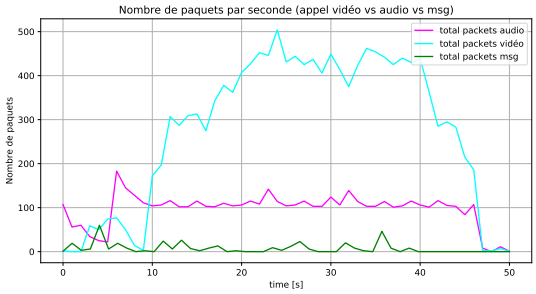

Fig. 3. Nombre de paquets échangés : appel audio vs appel vidéo vs messages textuels

un important échanges de STUN durant cette période, car il faut établir une connexion à haut débit, ce qui nécessite de contourner les NATs. Ensuite, lors de la connexion, un pic de TCP va apparaître. Cela vient du fait que des protocoles RTP et SIP sont encapsulés dans les paquets TCP qui permettent une livraison plus fiables que les UDP.Les protocoles RTP et SIP sont utilisés pour mettre en connexion deux utilisateurs lors d'un appel.

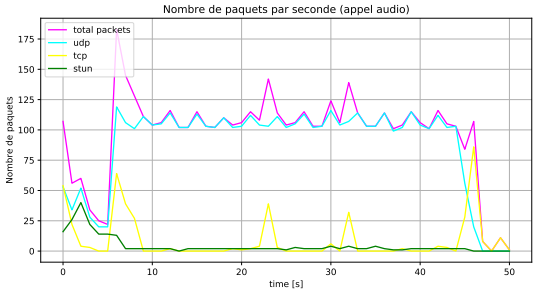

Fig. 4. Différents paquets échangés lors d'un appel audio

Dans le graphe ci dessous, on remarque un "pic" de paquets TCP en milieu d'appel. Celui-ci est apparu lors du lancement du partage d'écran d'un des utilisateurs. On peut donc émettre l'hypothèse que l'application prévient alors l'arrivée imminente de paquets concernant ce partage et demande potentiellement un agrandissement de la bande passante disponible.

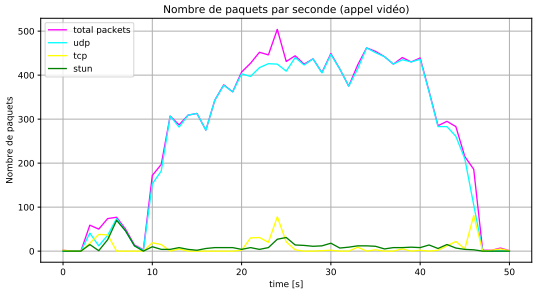

Fig. 5. Différents paquets échangés lros d'un appel vidéo

D. Même réseau vs différents réseaux

Lors d'échanges par message, l'application communique essentiellement avec des serveurs Microsoft. Lors d'un appel audio ou vidéo, le client communique avec un serveur relais (qui n'appartient pas forcément à Microsoft). Dans notre cas, ce dernier a comme adresse 87.66.147.99, ce qui correspond par Reverse DNS au serveur 99.147-66-87.adsl-dyn.isp.belgacom.be. Cependant, lorsque les utilisateurs se trouvent sur le même réseau Wi-Fi, la connexion lors d'un appel se fait directement entre les deux applications clients.

E. Conversation privée vs équipes

Durant ce rapport, nous avons omis la comparaison entre les conversations (textuelles ou audio-visuelles) privées et en équipe (ce qui est pourtant la fonctionnalité principale de Microsoft Teams). En effet, nos analyses n'ont pas montré de différences significatives entre ces deux situations, car elles sont en réalité très similaires, ce qui signifie que les opérations nécessaires (et donc les traces récupérées) sont globalement identiques.

References

[1] G. Huston, "Anatomy : A Look Inside Network Adress Translators" in "The Internet Protocol Journal", Vol. 7, n°3, pp.2–33, September 2004, https://www.cisco.com/c/dam/en_us/about/ac123/ac147/archived_issues/ipj_7-3/ipj_7-3.pdf, consulté le 03/04/2023.

[2] "QUIC", Wikipedia.org, https://fr.wikipedia.org/wiki/QUIC, consulté le 03/04/2023.

[3] "Real-time Transport Control Protocol", Wikipedia.org, https://fr.wikipedia.org/wiki/Real-time_Transport_Control_Protocol, consulté le 03/04/2023.

[4] "Simple Service Discovery Protocol", Wikipedia.org, https://fr.wikipedia.org/wiki/Simple_Service_Discovery_Protocol, consulté le 03/04/2023.

[5] "Universal Plug and Play", Wikipedia.org, https://fr.wikipedia.org/wiki/Universal_Plug_and_Play, consulté le 03/04/2023.

[6] "Domain Names & Identity for Everyone", whois.com, https://www.whois.com/, consulté le 05/04/2023.