LINFO1341 - Réseau Informatique

*Projet d'analyse du logiciel de vidéoconférence Google Meet

Hogge Jacques
SINF13BA
UCL - EPL
Louvain-la-neuve, Belgium
jacques.hogge@student.uclouvain.be

Barbason Romain
*S*INF13BA
UCL - EPL
Louvain-la-neuve, Belgium
romain.barbason@student.uclouvain.be

***Résumé*—Ce document est le rapport du projet 1 du cours LINFO1341 donnée à l'Université de Louvain-la-neuve à la faculté de l'EPL en Science informatique. Le but de ce projet est d'analysé les paquets de donnée qui sont échangés par une application de vidéoconférence, dans ce cas-ci, il s'agit de Google Meet.**

***Index Terms*—LINFO1341, Computer network, Google Meet, vidéoconférence, application, packages, échange, analyse**

I. Introduction

Ce rapport analyse les paquets de données échangés durant une vidéoconférence avec l'application Google Meet. Nous allons analyser ces paquets de données sous plusieurs aspects, en analysant les DNS, les couches réseau, les couches de transport, le chiffrement et la sécurité et pour finir sur les applications. Nos traces de paquets enregistrés sont disponibles sur notre répertoire GitHub [1].

II. DNS

A. Noms de domaines

Dans toutes les captures de paquets que nous avons prises, nous avons détecté 43 noms de domaines différents résolus, certains résolus plus de fois que d'autres bien sûr. Ils sont généralement résolus durant des phases d'interactions avec Google Meet, tel qu'appuyer sur un bouton, commencer un appel, actualiser le site, etc. Dans la liste des noms de domaines rencontrée, nous pouvons citer beaucoup de noms de domaine en lien avec l'entreprise Google tel que :

- accounts.google.com
- meet.google.com
- play.google.com
- scone-pa.clients6.google.com
- www.gstatic.com
- lh3.googleusercontent.com
- apis.google.com
- ...

et de nombreux autre que nous passons. Il s'agit de domaine permettant d'accéder ou d'envoyer des informations. Ceux-ci font partie de $77,5\%$ des noms de domaines observés, comme vous pouvez le voir sur le graphique 1.
Nous avons aussi observé d'autre domaine qui ne sont pas en lien avec Google, mais plus en lien avec le moteur de recherche et ses extensions ou bien de l'OS de l'ordinateur (Manjaro), nous pouvons citer :

- manjaro.news
- ping.manjaro.org
- avira-pwm-extensions.s3.eu-central-1.amazonaws.com
- ...

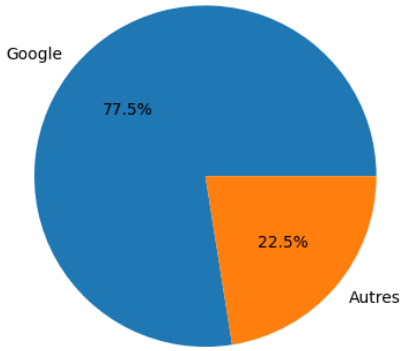

Fig. 1. Pourcentage d'appartenance des noms de domaine par des entreprises

B. Serveurs autoritatifs

Sur toutes les requêtes DNS capturé durant les tests de l'application de vidéoconférence Google Meet, nous n'avons capturé aucun serveur autoritatif pour les noms de domaines. Cela est tout à fait normal, car les serveurs autoritatifs pour les requêtes DNS vers des applications de Google sont gérées justement par l'entreprise Google et donc pas du tout géré par d'autres entreprises. Le fait que Google Meet utilise ses propres infrastructures a pour but de pouvoir contrôler la qualité, la sécurité, et d'avoir une structure personnalisée comme il le souhaite. [1]

C. Adresses IPs

Avec la commande *$ dig meet.google.com ANY* [2], nous pouvons voir que le site www.meet.google.com, en utilisant un wifi domestique ou le wifi Eduroam de l'UCL, possède

[1]https://github.com/RomainBarbason/LINFO1341-Projet-1

une adresse IPv4 (A), une adresse IPv6 (AAAA) mais pas de CNAME et pas de serveur sur lequel il repose. Comme dit précédemment, Google Meet est une application de Google et donc repose sur sa propre infrastructure.

- meet.google.com : 142.250.179.174 (A)
- meet.google.com : 2a00:1450:400e:802::200e (AAAA)

Par contre, si on utilise la fonction *dig* avec un resolver public tel que celui de Google (8.8.8.8), on peut observer que l'on reçoit en réponse, 6 adresses IPv4 et 4 adresses IPv6.

- meet.google.com : 142.250.145.113 (A)
- meet.google.com : 142.250.145.139 (A)
- meet.google.com : 142.250.145.101 (A)
- meet.google.com : 142.250.145.100 (A)
- meet.google.com : 142.250.145.102 (A)
- meet.google.com : 142.250.145.138 (A)
- meet.google.com : 2a00:1450:4013:c14::8a (AAAA)
- meet.google.com : 2a00:1450:4013:c14::64 (AAAA)
- meet.google.com : 2a00:1450:4013:c14::66 (AAAA)
- meet.google.com : 2a00:1450:4013:c14::71 (AAAA)

Toutes ces adresses ont un TTL (time to live) de 300 secondes avec le resolver DNS de Google et un TTL, 273 secondes et 43 secondes pour respectivement l'adresse IPv6 et IPv4 sans resolver publique avec le wifi domestique ou UCL.

Bien évidemment, Google, en raison de sa grande utilisation, possède un grand nombre d'adresses IP différente qui change en fonction de la surcharge des serveurs, de l'endroit où nous sommes, etc. Il est donc normal de ne pas avoir les mêmes adresses IP, mais celle-ci ramènent bel et bien au même domaine.

Comme nous pouvons le voir sur le graphique 2 ci-dessous, il n'y a pas de préférence pour l'IPV4 ou l'IPV6. Cette préférence peut simplement s'expliquer par le fait que l'IPV4 fut implémentée avant l'IPV6 et qu'une transition entre les deux types d'adresse prend du temps et que pour l'instant seulement la moitié des adresses IP sont sur 128-bits.

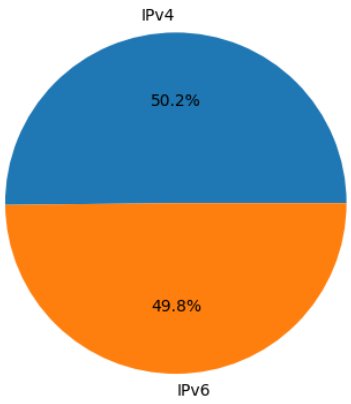

Fig. 2. Pourcentage des versions des adresses IP lors des requêtes DNS

Nous pouvons aussi remarquer qu'aucune des requêtes DNS ne possède de records additionnels. Les records additionnels sont facultatifs, à la demande de celui qui envoie le paquet, et permettent juste de donner des informations en plus si nécessaire. Dans notre cas, il n'y en a pas.

Il est parfois possible de rencontrer des comportements inattendus de la part des requêtes DNS, comme par exemple :

- Des réponses à des paquets non-existants
- Un grand nombre de paquets en un laps de temps court
- Des paquets DNS avec la valeur flag "Reply Code" différente de 0

Dans notre analyse de paquets, nous ne remarquons aucune requêtes DNS de ce type et donc aucun comportement inattendu.

III. Couche Réseau

A. NAT

Lors de l'utilisation de l'IPv4, Google Meet peut utiliser plusieurs techniques afin de pouvoir traverser le NAT (Network Address Translation). Le NAT est un processus qui permet de modifier les adresses privées d'un appareil pour le transformer par le router par une adresse publique qui pourra être utilisé hors du réseau.

Une des techniques les plus utilisés par Google Meet est le protocole STUN (Session Traversal Utilities for NAT). Il est utilisé dans les applications de communication en temps réel pour traverser les pars-feu et routeurs.

Dans un cas ou STUN n'est pas possible, Google Meet peut utiliser TURN (Traversal Using Relay NAT). TURN est un protocole qui permet à un utilisateur de relayer les informations à un serveur qui sert de relais entre les deux utilisateurs.

Parallèlement à ça, Google Meet utilise la technique ICE (Interactive Connectivity Establishment) afin de trouver le chemin le plus optimal pour la communication instantanée. [3]

IV. Couche Transport

A. Protocoles de transports

Dans les protocoles de transports analysés avec les paquets d'information, nous avons remarqué que le protocole QUIC (Quick UDP Internet Connections) [4] est plus utilisé durant un appel après que l'utilisateur, soit, rejoint une salle déjà crée avec un lien ou bien s'il en crée une directement.

UDP est plus utilisé dans l'appel en lui-même (Avec ou sans caméra) et dans la gestion du chat

TCP est plus présent quand l'utilisateur est sur l'écran d'accueil que partout ailleurs.

Avec l'analyse de nos paquets, nous pouvons remarquer sur le graphique 3 que tous les noms de domaines sont au moins connectés deux fois, celui ayant le plus de connexion étant *meet.google.com*, ce qui n'est pas étonnant étant donné que c'est le nom de domaine de Google Meet. Le fait de se connecter plusieurs fois au même nom de domaine est quelque chose de fréquent sur des sites hautement fréquentés, un nom de domaine peut avoir plusieurs serveurs, cela aide à garder le site Web disponible, rapide et moins susceptible de cesser de fonctionner à cause d'une panne.

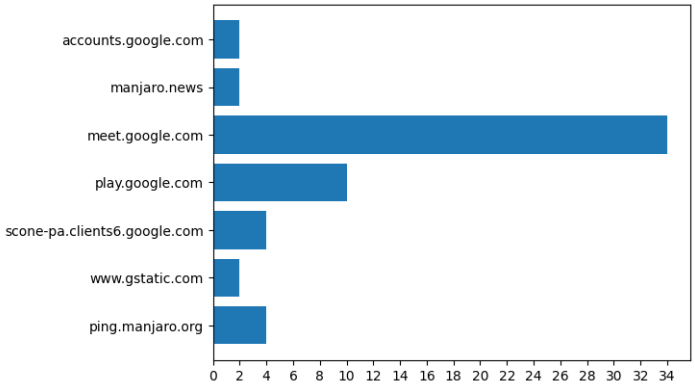

Fig. 3. Nombre de connexions vers un nom de domaine

B. UDP

Plusieurs protocoles se basent/utilisent UDP pour leur bon fonctionnement et nous retrouvons deux d'entre eux durant l'utilisation de Google Meet.
Premièrement, nous observons du trafic de paquets avec le protocole de transport QUIC, qui est un protocole de transport crée par l'entreprise Google dans l'objectif d'améliorer les performances des applications Web, notamment en termes de vitesse. [4]
Il est donc normal de le retrouver utilisé dans leurs applications. La version utilisée dans tous les paquets est la version 1.0 et les extensions négociées dans le *handshake* sont

- pre_shared_key : Une liste d'identités de clé symétriques connue de l'utilisateur
- key_share : contient les paramètres cryptographiques du point de terminaison [5]
- supported_versions : Indique les groupes que le client supporte pour l'échange de clés, trié du plus préféré au moins préféré [6]

Nous pouvons aussi observer un autre protocole, le SRTCP, qui permet l'authentification et l'intégrité de message, du cryptage et une protection contre les *replay attacks*. Une *replay attack* est une attaque informatique où une personne intercepte une communication entre deux hôtes, et s'en sert pour rejouer la même communication plus tard.

V. Chiffrement et sécurité

A. Sécurité des protocoles

Avec le transfert massif de données constamment, plusieurs protocoles utilisent d'autres protocoles pour se sécuriser, comme par exemple, le trafic UDP qui est chiffré et sécurisé en utilisant le protocole DTLS, qui permet de sécuriser contre l'écoute clandestine (eavesdropping), l'attaque de l'homme du milieu (tampering) et la falsification de messages (message forgery) [7] Le DTLS utilise le protocole TLS, utilisé pour créer une sécurité lors de communications sur un réseau. TLS n'est cependant pas seulement utilisé pour DTLS mais est aussi utilisé sur Google Meet, séparément ou avec d'autres protocoles. La version utilisée sur Google Meet est la 1.2, à l'exception des paquets "Client Hello" et "Server Hello" qui utilise le TLS 1.0. Le TLS utilise un ensemble d'algorithmes de chiffrements pour sécuriser une connexion réseau, appelé une *Cipher suite*. Il en existe beaucoup mais nous utilisons uniquement les suivants [8] :

- TLS_AES_128_GCM_SHA256
- TLS_AES_256_GCM_SHA384
- ECDHE_RSA_WITH_AES_128_GCM_SHA256

Les protocoles DNS ne sont pas sécurisés en utilisant d'autres protocoles sur Google Meet, mais si c'était le cas, ils auraient pu être protégés avec :

- DoH (DNS over HTTPS)
- DoT (DNS over TLS)
- DNSSEC (DNS Security Extensions)

VI. Application

A. Conversation

Nous avons voulu tester s'il y avait une différence entre un appel en vidéoconférence (avec ou sans caméra et micro) mais, Google Meet ne permet pas de faire une conversation sans lancer un appel-vidéo, du coup, il n'y a aucun changement dans le transfert de paquet, excepté le protocole utilisé pour envoyer le message durant l'appel.

B. Volume des données

Avec l'application *Wireshark [9]* avec laquelle nous avons capturé nos paquets, il est possible d'avoir certaines statistiques sur les paquets enregistrés, nous pouvons remarquer que :

- Pour les transferts de paquets QUIC, ils ont tous en moyenne une taille de 1401 Bytes avec 68% des paquets et le reste étant en moyenne de 100 Bytes, d'autre paquet ont des tailles différentes, mais ceux-ci sont minoritaires. Du point de vue du nombre de paquets par minutes, nous sommes à environs 7692 paquets/minutes. Avec donc en moyenne un transfert de données d'environs 8020.53 KBytes/minute ou bien 8.02 MBytes/minute.
- 57% des transferts DNS font en moyenne 165 Bytes et 43% font en moyenne 102 Bytes, Cela peut s'expliquer, car les requêtes DNS sont divisé en deux formats, les *Query*, qui demande les noms de domaine, et les *Answer* qui ont la réponse à la demande et qui donc ajoute la réponse au paquet, ce qui explique qu'une moitié est plus volumineuse que d'autre. Du point de vue du nombre de paquets par minute, nous sommes à environs 358 paquets/min, donc en soi 144 paquets *Query*/minutes, et 198 paquets *Answer*/minutes. Cela représente en moyenne 32889.46 Bytes/minutes, soit $32,89$ KBytes/minutes pour les requêtes DNS, avec donc en moyenne, 11093.76 Bytes/minutes, soit 11.09 KBytes/minutes pour les requêtes DNS *Query* et en moyenne 20166.30 Bytes/minutes, soit $220,17$ KBytes/minutes pour les requêtes DNS *Answer*

C. Transfert de données

La manière dont les paquets sont transportés dépend d'énormément de facteur, de la charge des serveurs, du nombre de personnes présente, du réseau dans le qu'elle se trouve les personnes de l'appel, etc.

Prenons un premier cas, si toutes les personnes en vidéoconférence sont tous connectés au même réseau local/Wifi, alors l'application ne va pas passer par des serveurs relais (routeurs) pour retransmettre les paquets d'informations aux autres personnes de l'appel. L'application va profiter du réseau local pour envoyer en local les paquets de donnée. Cela est possible, car l'IP de chaque machine est connue au sein du réseau local. Cela permet de gagner sur la performance et sur le temps d'attente et ainsi ne pas saturer les serveurs inutilement.

Dans la situation où les personnes en vidéoconférences ne sont pas sur le même réseau, Google Meet passera par un serveur relais pour transmettre les paquets d'informations de manière efficace, sans perte et le plus rapidement possible aux autres participants n'étant pas sur le même réseau. Il existe des cas où des serveurs relais sont utilisés même si toutes les participants de l'appel sont sur le même réseau, notamment quand les paquets nécessitent une optimisation, une aide, etc de la part d'un serveur relais, par exemple lors d'un partage d'écran, de l'utilisation des sous-titres automatique, etc. Ou bien même lors d'un dysfonctionnement du réseau local. C'est l'application qui décide alors de passer par un serveur relais ou non pour améliorer la qualité de l'appel vidéo. [10]

Même si Google possède un grand nombre de Serveurs dans le monde, ce n'est pas obligatoire de passer que par ceux-ci. Il est tous a fait possible de passer par des serveurs appartenant à d'autre entreprise et se dans le but d'optimiser la transmission des paquets de donnée afin que cela se fasse le plus rapidement et efficacement possible.

REFERENCES

[1] Google Centres de Données nbsp;, Google. Google. Available at: https://www.google.com/intl/fr/about/datacenters/ (Accessed: April 1, 2023).

[2] Dig(1) - linux man page (no date) dig(1): DNS lookup utility - Linux man page. Available at: https://linux.die.net/man/1/dig (Accessed: March 29, 2023).

[3] Nguyen, C. (2020) WebRTC-the technology that Powers Google Meet/Hangout, Facebook Messenger and discord, Medium. Available at: https://medium.com/swlh/webrtc-the-technology-that-powers-google-meet-hangout-facebook-messenger-and-discord-cb926973d786 (Accessed: April 3, 2023).

[4] What is QUIC? Everything You Need to Know, Available at: https://www.auvik.com/franklyit/blog/what-is-quic-protocol/

[5] RFC 8446, IANA - Internet Assigned Numbers Authority. Available at: https://www.rfc-editor.org/rfc/rfc8446.html#section-4.2.8

[6] RFC 8446, IANA - Internet Assigned Numbers Authority. Available at: https://www.rfc-editor.org/rfc/rfc8446.html#section-4.2.7

[7] RFC 9147, IETF Datatracker. Available at: https://datatracker.ietf.org/doc/html/rfc9147

[8] Transport Layer Security (TLS) Extensions, IANA - Internet Assigned Numbers Authority. Available at: https://www.iana.org/assignments/tls-extensiontype-values/tls-extensiontype-values.xhtml

[9] Wireshark · go deep, Wireshark. Available at: https://www.wireshark.org/

[10] Marc (2021) Améliorer l'accès aux Visioconférences Jitsi, Aduneo. Available at: https://www.aduneo.com/ameliorer-lacces-aux-visioconference-jitsi/ (Accessed: April 3, 2023).