Projet LINFO1123
Analyse de NextCloud

Giovanni Karra
45032100

Pierre Leboutte
43302100

I. Introduction

Dans ce rapport, nous allons analyser le fonctionnement de NextCloud, un logiciel de partage de fichiers. Nous allons entre-autres discuter des divers protocoles de transfert utilisés dans les différentes fonctionnalités, ainsi que la configuration du réseau des hébergeurs de NextCloud. Cette analyse n'est pas exhaustive, en effet NextCloud offre pléthore de fonctionnalités, et il faudrait une étude approfondie de plusieurs dizaines de pages pour toutes les couvrir, mais nous voulions présenter une base assez solide pour permettre au lecteur d'approfondir lui-même cette étude.

II. Vue d'ensemble

Nous présentons ci-dessous un overview des paquets impliqués dans l'opération de téléchargement, un choix arbitraire qui nous semble cependant pertinent car représentatif de l'ensemble des scénarios.

Protocole	Pourcent Paquets	Paquets	Pourcent Octets	Octets	Bits/s	Paquets de Fin	Octets de Fin	Octets/s de Fin	PDUs
Frame	100.0	70031	100.0	78105912	13 M	0	0	0	70031
Ethernet	100.0	70031	1.3	980863	171 k	0	0	0	70031
Internet Protocol Version 6	0.4	258	0.0	10320	1808	0	0	0	258
User Datagram Protocol	0.1	41	0.0	328	57	0	0	0	41
QUIC IETF	0.0	22	0.0	11919	2088	22	10037	1758	25
Multicast Domain Name System	0.0	19	0.0	6583	1153	19	6583	1153	19
Transmission Control Protocol	0.2	161	0.0	25082	4394	125	8984	1574	161
Transport Layer Security	0.0	16	0.0	10708	1876	16	10708	1876	16
Hypertext Transfer Protocol	0.0	20	0.0	10145	1777	4	864	151	20
Online Certificate Status Protocol	0.0	14	0.0	4048	709	14	4048	709	14
Line-based text data	0.0	2	0.0	98	17	2	98	17	2
Internet Control Message Protocol v6	0.1	56	0.0	5600	981	56	5600	981	56
Internet Protocol Version 4	99.6	69771	1.8	1395436	244 k	0	0	0	69771
User Datagram Protocol	0.3	185	0.0	1480	259	0	0	0	185
QUIC IETF	0.1	56	0.1	42863	7509	56	23662	4145	76
Multicast Domain Name System	0.0	19	0.0	6583	1153	19	6583	1153	19
Domain Name System	0.2	110	0.0	12083	2117	110	12083	2117	110
Transmission Control Protocol	99.4	69582	96.8	75617449	13 M	45009	34665798	6073 k	69582
Transport Layer Security	35.1	24565	108.5	84728007	14 M	24565	26348341	4616 k	38929
Hypertext Transfer Protocol	0.0	8	0.0	3089	541	3	567	99	8
Online Certificate Status Protocol	0.0	4	0.0	1108	194	4	1108	194	4
Line-based text data	0.0	1	0.0	8	1	1	8	1	1
Internet Group Management Protocol	0.0	4	0.0	64	11	4	64	11	4
Address Resolution Protocol	0.0	2	0.0	74	12	2	74	12	2

Fig. 1. Hiérarchie des protocoles, téléchargement d'un grand fichier

Nous pouvons observer - si l'on exclut le bruit - que notre application n'utilise que des adresses IPv4, se justifiant probablement par la répartition encore très vaste de ce choix d'adressage. Constatons l'absence de QUIC. Nous avons par contre beaucoup de TCP (avec TLS). La taille des paquets est principalement comprise entre 1280 et 2559 bytes, ce qui est probablement expliqué par le fait que notre carte réseau est capable de découper certains paquets en plus petits fragments.

III. NextCloud

NextCloud est un logiciel libre permettant l'hébergement et le partage de fichiers, ainsi que bien d'autres fonctionnalités. Afin d'utiliser cet outil, deux choix se présentent à nous: soit nous hébergeons nous-même un serveur qui tourne NextCloud, soit nous faisons appel à un fournisseur qui utilise le logiciel dans son service. N'ayant pas de moyens d'hébergement, nous avons décidé d'utiliser un serveur du fournisseur TheGoodCloud, qui offre un espace de stockage de 2GB avec l'abonnement gratuit. NextCloud peut être utilisé comme Google Drive, c'est-à-dire pour stocker et partager des fichiers à distance, mais aussi comme Dropbox si nous le souhaitons, c'est-à-dire en mettant en place une synchronisation des fichiers localement par rapport au serveur. Dans la suite nous auront l'occasion d'analyser ces deux cas de figure. En plus du partage de fichiers, il y a toute une suite d'applications bureau semblables à la suite Office, ainsi que des applications de discussions, de prise de notes, un calendrier, et bien d'autres encore. NextCloud est donc un réel compétiteur au Drive de Google et OneDrive de Microsoft en terme de fonctionnalités proposées, et se veut aussi libre et respectueux de la vie privée des utilisateurs.

IV. DNS

En s'inscrivant sur TheGoodCloud, une URL parmis plusieurs peut nous être attribuée, et il se trouve que le site nous a attribué `use09.thegood.cloud`. En envoyant une requête DNS pour ce nom, nous ne trouvons qu'un record A avec l'adresse `82.94.183.165`, qui est l'adresse IPv4 d'un serveur qui se trouve à Amsterdam. Le serveur autoritatif pour ce nom de domaine est `auth01.dns.trueserver.nl`, ce qui signifie probablement que les serveurs de TheGoodCloud sont hébergés par les services de TrueServer.

V. Couche réseau

Cette application utilise exclusivement des adresses de la famille IPv4. Il existe différents noms de domaine pour accéder aux services de TheGoodCloud, nommés `use0<num>.thegood.cloud`. À part use01, ils correspondent tous à la même adresse IPv4, ce qui signifie que TheGoodCloud emploie un serveur proxy inversé, qui reçoit les paquets et les distribue aux bons serveurs selon le nom d'hôte dans les requêtes HTTP par exemple. Évidemment, à moins d'avoir accès à des informations en interne de TheGoodCloud, nous ne sommes sûrs de rien par rapport au fonctionnement de leur réseau.

VI. Couche transport

L'entièreté des interactions avec le serveur de TheGoodCloud utilisent TCP. En effet, les seuls séquences

UDP qu'on peut observer dans les traces proviennent soit d'une interaction avec un autre service, soit de requêtes DNS. Lorsque l'adresse IP du serveur a été saisie grâce au DNS, une connexion TCP est directement établie avec celui-ci, et cette même connexion est conservée jusqu'à la fermeture de l'application.
Au dessus de TCP, le protocole le plus largement utilisé pour transférer des données est HTTP/2. Il est bien-sûr utilisé pour récupérer les objets web du site, comme par exemple des documents HTML, CSS et javascript, mais aussi pour les interactions avec l'application et pour les transferts de fichiers. Pour permettre une utilisation aussi large du protocole HTTP, l'application utilise des extensions permettant d'avoir des nouvelles fonctionnalités très utiles. L'extension principale utilisée est WebDAV, qui a été conçue pour permettre la modifcation collaborative et concurrente de documents sur le web. WebDAV introduit des méthodes très utiles tel que COPY, MOVE et DELETE, que nous aborderons plus en détail dans la suite de l'analyse.
La grande majorité des échanges sont chiffrés grâce au protocole TLSv1.3, dont le handshake est réalisé juste après l'établissement d'une connexion TCP. Nous pouvons donc dire que le protocole le plus utilisé est HTTPS, c'est-à-dire HTTP au-dessus de TLS.

VII. Chiffrement et sécurité

Comme nous l'avons évoqué précédemment, le protocole de sécurité TLS est utilisé dans la très grande majorité des paquets échangés, ce qui n'est pas étonnant vu la popularité actuelle de HTTPS et le besoin de sécuriser les transferts de fichiers. La version utilisée est TLSv1.3, qui est la plus récente.
Le handshake TLS commence juste après l'établissement d'une connexion `TCP`. Le navigateur commence par envoyer un segment Client Hello contenant, entre autres, un nonce et les suites de cryptographie supportés. Le serveur répond avec un Server Hello contenant des informations similaires, dont une unique suite de cryptographie sélectionnée par le serveur: `TLS_AES_256_GCM_SHA384`
Ensuite, le serveur envoie son certificat numérique, qui est une preuve de son identité vérifiable par le navigateur. Le navigateur peut ensuite vérifier ce certificat en le comparant avec sa liste de certificats de confiance, et s'assurer que le serveur est bien celui qu'il prétend être. Le certificat en question a été fourni au serveur par Let's Encrypt, le plus grand fournisseur de certificats X.509 pour TLS au monde, sans demander de paiement en retour. Le certificat est signé avec l'algorithme `sha256WithRSAEncryption`, et il a un temps de validité de trois mois. Il est intéressant d'observer que le client ne fournit aucun certificat au serveur.
Après la vérification du certificat à l'aide de l'Online Certificate Status Protocol (OCSP), le navigateur génère une clé de session secrète, qu'il chiffre avec la clé publique du serveur avant de l'envoyer. Le serveur utilise sa clé privée pour déchiffrer cette clé de session.
Une fois que les clés de session sont établies des deux côtés, le handshake TLS est terminé et la communication sécurisée peut commencer. Les données échangées entre le navigateur et le serveur sont désormais chiffrées et protégées grâce aux algorithmes cryptographiques négociés durant le handshake.

VIII. Application

Nous avons analysé des captures de paquets en utilisant différentes fonctionnalités de l'application, et nous avons fait plusieurs observations que nous détaillons ci-dessous.

A. Téléchargement de fichiers

Les fichiers sur le serveur ont généralement comme URI `use09.thegood.cloud/remote.php/dav/files/<adresse_mail>/<nom_du_fichier>`. Durant le reste de cette section, par soucis de concision, nous désigneront '`use09.thegood.cloud/remote.php/`' par '`./`'.
Pour tranférer un fichier du serveur vers la machine locale, le navigateur effectue une requête HTTP/2 avec la méthode GET et l'URI du fichier, le serveur répond en envoyant des séquences HTTP/2 de type DATA contenant les données du fichier, et lorsque le transfert de celui-ci est terminé, le navigateur le sauvegarde sur la machine.
Pour transférer un fichier dans le sens inverse, c'est-à-dire de la machine vers le serveur, la méthode HTTP/2 utilisée est PUT, qui permet d'envoyer un fichier à un URI donné, suivi de séquences DATA pour envoyer les données. Lorsque le fichier que nous voulons envoyer est grand, nous commençons par envoyer une requête MKCOL, une méthode de l'extension WebDAV qui sert à créer une 'collection', c'est-à-dire l'équivalent d'un dossier, ayant comme URI `./dav/uploads/<mail>/web-file-upload-<id>`. Ensuite plusieurs séquences usant de la méthode PUT sont envoyés, ainsi le fichier est transféré par le biais de différents fichiers représentant différentes parties de ce premier, et se trouvant à l'URI `<uri_collection>/i`, $i = 1...nombre\_de\_transferts$, et pour finir une requête MOVE est utilisée pour envoyer tous les sous-fichiers vers un seul, ayant le même nom que celui de départ et se trouvant à l'URI `./dav/files/<mail>/`.
En ce qui concerne la quantité de paquets envoyés, nous pouvons observer un pic à chaque fois qu'une opération de transfert est effectuée.
Nous pouvons observer que lorsque nous uploadons un grand fichier, il est découpé et envoyé dans plusieurs paquets différents, d'où les pics prolongés. Nous pouvons aussi voir que le nombre d'erreurs de transmission TCP (barre rouge) atteint aussi des pics lors des transferts de fichiers, ce qui est logique vu l'augmentation générale du nombre de paquets transférés. Enfin, nous remarquons que le débit de paquets transférés est deux à trois fois plus élevé lors des download que lors des upload (il est important de préciser que ces captures ont été réalisées dans les même conditions de réseau). Ceci est lié au fait que le download est bien plus demandé sur le réseau que l'upload, et donc le réseau est mieux optimisé pour cette opération.
Pour ce qui est de la taille des paquets, nous remarquons

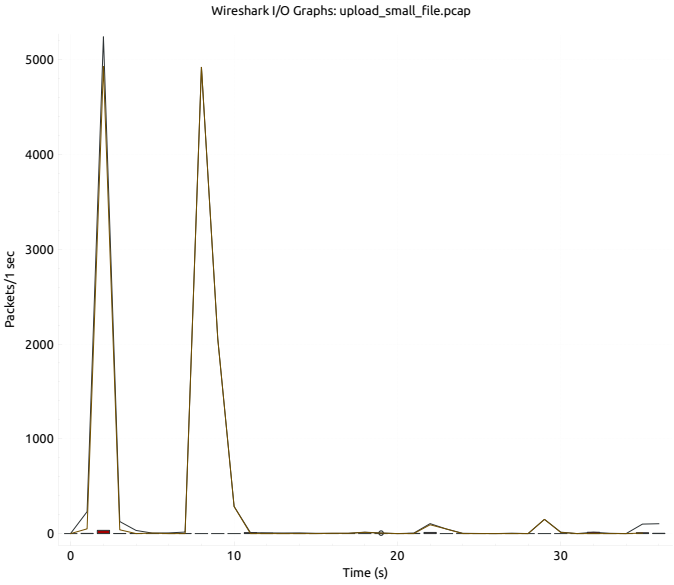

Fig. 2. Débit des paquets - Upload de deux petits fichiers

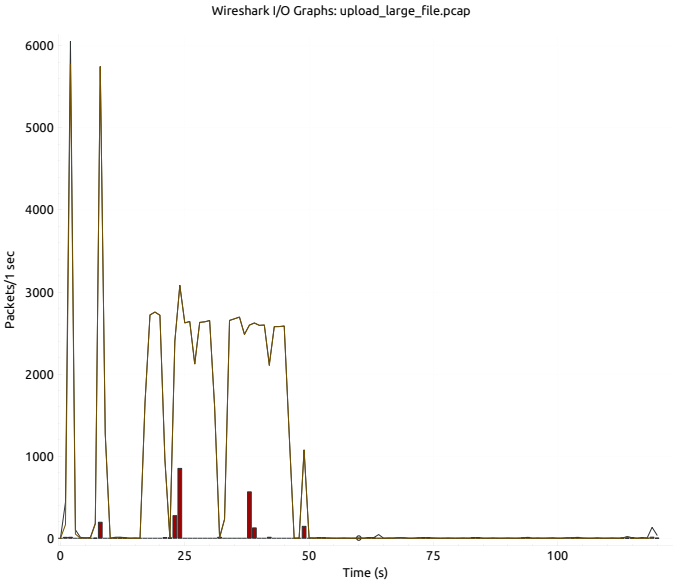

Fig. 3. Débit des paquets - Upload d'un grand fichier

que les pics ne correspondent pas exactement aux pics du nombre de paquets envoyés par seconde, mais en général nous pouvons dire que lors de transferts de fichiers, il y a un grand débit de paquets qui en plus ont une grande taille.

B. Manipulation de fichiers

La manipulation des fichiers sur le site web use également d'HTTP/2, et en particulier des méthodes de l'extension WebDAV.

Pour créer un fichier, la méthode POST est utilisée pour signaler au serveur que nous voulons créer un fichier, et avec quel template (fichier texte, docs, etc.).

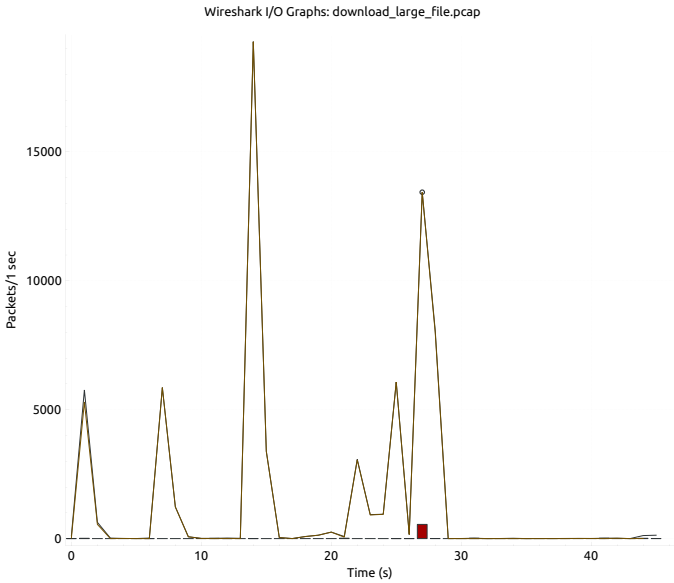

Fig. 4. Débit des paquets - Téléchargement d'un grand fichier

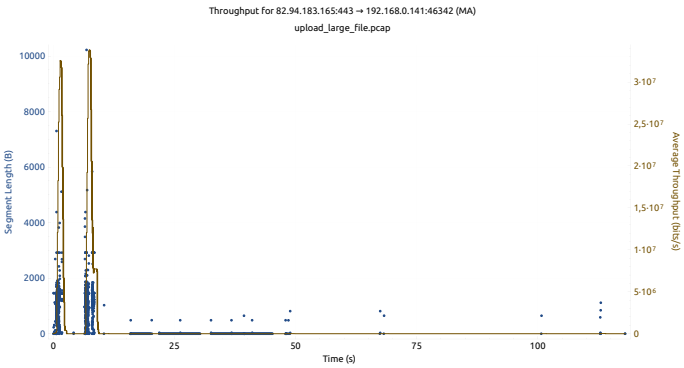

Fig. 5. Taille des paquets - Upload d'un grand fichier

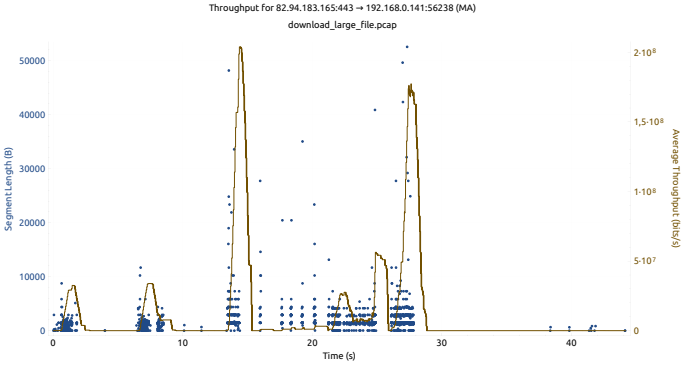

Fig. 6. Taille des paquets - Téléchargement d'un grand fichier

Pour créer un dossier, la méthode MKCOL abordée précédemment est utilisée, signalant au serveur la volonté de créer une 'collection', c'est-à-dire un dossier, ayant un certain nom.
Pour supprimer un fichier, la méthode DELETE est utilisée, demandant au serveur de supprimer un fichier à un URI donné.
Pour modifier l'emplacement d'un fichier, la méthode MOVE est utilisée, demandant la modification de l'URI d'une ressource.
Enfin, pour copier une ressource d'un URI à un autre, la méthode COPY est utilisée, qui fonctionne de la même manière que MOVE mais en conservant la ressource de base.

C. Écriture dans un document

Il est possible de modifier les documents directement sur le site grâce à la suite bureau de NextCloud. Afin d'assurer une modification rapide et concurrente, c'est le protocole WebSocket, que nous n'avons pas abordé auparavant, qui est utilisé.
Le protocole WebSocket est une technologie de communication bidirectionnelle, full-duplex, conçue spécifiquement pour les applications web qui nécessitent des échanges de données en temps réel entre un navigateur web et un serveur. Contrairement aux méthodes classiques qui impliquent de multiples requêtes HTTP, WebSocket offre une connexion persistante qui permet aux clients et aux serveurs d'envoyer et de recevoir des données simultanément, ce qui rend ce protocole parfait pour une application comme la modification concurrente et en temps réel de documents.
Le fonctionnement de WebSocket commence par un handshake initial qui ressemble à une requête HTTP standard: Le client envoie une requête HTTP avec la méthode GET pour l'URI `/chat`, avec le paramètre "upgrade" mis à "websocket", ensuite le serveur répond avec le code 101 "Switching Protocols", signalant que la connexion WebSocket a été établie. Une fois la connexion établie, les données peuvent être échangées sous forme de messages textuels ou binaires.
L'un des avantages majeurs de WebSocket est sa capacité à réduire la surcharge du trafic réseau en évitant les requêtes HTTP fréquentes.

D. Synchronisation locale

Comme mentionné précédemment, NextCloud fourni également une application bureau permettant de synchroniser les fichiers localement avec le serveur qui stocke les fichiers en fournissant son URL. Malheureusement, même en déchiffrant les paquets grâce aux secrets contenus dans le SSLKEYLOGFILE, nous n'avons aucune information sur le fonctionnement de la synchronisation au-delà du handshake TLS, et du fait que c'est encore une fois le protocole HTTP qui est utilisé au-dessus de TLS, mais toute autre information reste chiffrée.

IX. Conclusion

En conclusion, il est utile de mettre en lumière l'importance des protocoles TLS, TCP et HTTP/2, l'incorporation de WebDAV pour la manipulation de fichiers, ainsi que l'absence de protocoles QUIC. L'analyse réseau nous a convaincu que NextCloud était une option crédible, illustrée par son chiffrement avancé et son option de self-host. Au vu des nombreuses fonctionnalités offertes par la plateforme (contacts, calendriers, etc.) notre travail constitue une base pouvant être étendue à d'autres analyses réseaux.

X. Références

Sources d'informations:

- https://nextcloud.com/
- https://fr.wikipedia.org/wiki/
- https://www.iplocation.net/
- https://docs.nextcloud.com/server/19/user_manual/

Répertoire GitHub:

- https://github.com/PierreLeboutte/Projet-1-LINFO1341