Alice Burlats, Aurélien Buchet, Vany Ingenzi , Paul-Edouard Bertrand Van Ouytsel, Louis Navarre,
Olivier Bonaventure

Analyse d'applications réseaux

1 Contexte du projet

En tant qu'ingénieur·e réseau, il est important de pouvoir comprendre le fonctionnement des protocoles qui constituent l'Internet et également le comportement de ses applications. C'est en combinant ces deux savoirs qu'un·e ingénieur·e réseau peut à la fois améliorer les protocoles de l'Internet et concevoir des applications réseaux efficaces.

Dans ce projet, vous analyserez une application réseau fonctionnant sur un schéma *peer-to-peer* soit de partage d'écran, soit de partage de fichiers. Ce type d'application est très commun et comporte beaucoup de comportements différents, chacun alliant des exigences particulières se traduisant en des utilisations et combinaisons de protocoles différentes.

Par groupe de trois, vous étudierez comment les différentes fonctionnalités de l'application considérée sont implémentées en observant le trafic réseau produit par celle-ci. Vous produirez un ensemble de slides résumant vos observations. Vous trouverez dans ce document les lignes directrices nécessaires pour mener à bien cette étude et les délivrables attendus.

2 Méthode d'analyse

Commencez par découvrir toutes les fonctionnalités de l'application étudiée. Parcourez-la et énumérez ce qu'elle permet de faire. Sur base des fonctionnalités découvertes, créez un ou plusieurs scénarios de tests permettant de les parcourir de façon systématique. Cela vous permettra de répéter les mêmes actions dans l'application dans des conditions réseaux différentes (par ex. connecté en Wi-Fi, Ethernet ou partage 4/5G).

À l'aide de `tcpdump` ou Wireshark, lancez une capture de paquets. Pour observer un maximum de comportements, choisissez l'interface de capture `any` qui reprendra l'entièreté des paquets échangés, peu importe leur provenance (Ethernet, Wi-Fi, etc). Vérifiez à bien identifier les paquets qui proviennent de votre application. Afin de faciliter votre analyse du trafic, il est important de stopper *toutes* les autres applications susceptibles d'utiliser Internet pendant le test. Une fois la capture lancée, démarrez l'application et exécutez votre scénario de test et stoppez la capture une fois terminé [1]. Il peut vous être utile de capturer votre écran lorsque vous exécutez votre scénario. Cela vous permettra par après de mieux comprendre le lien entre votre utilisation de l'application et l'activité observée sur le réseau. Avec certaines applications, il est possible de stocker les secrets échangés en créant une variable d'environnement **SSLKEYLOGFILE** avec un chemin vers un fichier. Il est alors possible de déchiffer le contenu des paquets dans Wireshark en indiquant ce chemin dans *Edit->Preferences->Protocols->TLS->(Pre)-Master-Secret log filename.* **Attention !** Pour être sûr que cela fonctionne, veillez à initialiser cette variable d'environnement avant chaque lancement de l'application et à lancer l'application depuis ce même terminal.

2.1 Analyse des traces

Les questions reprisent dans cette sous-section sont un point de départ de votre réflexion. Elles vous permettent de vous donner des pistes d'analyses des différents protocoles. Il est attendu de vous que vous interprétiez et expliquiez l'utilité des fonctionnalités des protocoles utilisés lorsque vous le pouvez. Il est important d'être factuel et de vérifier vos affirmations en citant des faits ou des références. Il peut vous être utile de produire des graphes ou tableaux pour soutenir vos explications [2]. Donnez toujours le contexte dans lequel vous avez observé ces faits.

1. Il peut être intéressant de garder quelques secondes d'inactivité à la fin du scénario de test pour observer des comportements retardés.
2. Soyez attentif à la taille de ceux-ci, minimiser la en gardant une certaine lisibilité.

Lors de votre analyse des traces, il peut vous être utile de développer des scripts qui extrairont et calculeront des statistiques de votre choix sur base des fichiers de capture de paquets. Vous pouvez utiliser par exemple la librairie Python `pyshark` pour cela.

2.1.1 DNS

- Combien de noms de domaines sont résolus et quand ?
- Quels sont les serveurs autoritatifs pour ces noms de domaines ? Sont-ils gérés par des entreprises différentes ? en a-t-il d'autres que celle qui détient l'application ?
- Quels sont les types de requête DNS effectuées ?
- Lorsqu'une requête DNS souhaite obtenir une adresse IP, quelle est sa famille ? Il y a-t-il une version IP préférée par l'application ?
- Les requêtes contiennent elles des records additionnels ? Le cas échéant, à quoi servent-ils ?
- Observez-vous des comportements DNS inattendus ?
- Certaines applications pourraient ne pas utiliser (beaucoup) le DNS. Dans quels cas et pourquoi cela arrive-t-il ?

2.1.2 Couche réseau

- Lorsque IPv4 est utilisé, l'application utilise-t-elle des techniques pour traverser les NAT [3]. Retrouvez à quels noms de domaine elles correspondent, observez-vous une tendance particulière dans la famille d'adresse ? Pouvez-vous l'expliquer ?
- **Sachant que les applications à analyser sont plus orientées *peer-to-peer*** plutôt que dans le schéma classique client-serveur, quelles méthodes sont utilisées pour mettre en relation les deux clients (i.e., *peers*) ?
- En plus des deux clients, y a-t-il des intermédiaires participant à la communication ?

2.1.3 Couche transport

- Quels sont les protocoles de transports utilisés pour chaque fonctionnalité ?
- Il y a-t-il plusieurs connexions vers un même nom de domaine ? Si oui, pouvez-vous l'expliquer ?
- Lorsque vous observez du trafic UDP, identifiez-vous d'autres protocoles que QUIC et DNS ? Expliquez comment ils sont utilisés par l'application.

2.1.4 Chiffrement et sécurité

- L'utilisation du DNS est-elle sécurisée ? Si oui, comment ?
- La communication est-elle sécurisée ? Si oui, comment ? Utilise-t-elle TLS ou une autre méthode *ad hoc* ? Pourquoi l'application privilégierait-elle une autre méthode que du TLS ?
- (Spécifique à l'utilisation de TLS :) Quel est la durée de vie des certificats utilisés ? Par qui sont-ils certifiés ?
- (Spécifique non-TLS :) Comment les clés de chiffrement sont-elles dérivées dans ce cas ?
- Lorsque vous pouvez observer l'établissement du chiffrement, quels sont les algorithmes de chiffrement utilisés ?
- Si vous observez du trafic UDP, semble-t-il chiffré ? Comment est-il sécurisé ?

3. Les NAT étant propres à IPv4, le cours ne les aborde pas. Une bonne ressource est `https://www.cisco.com/c/dam/en_us/about/ac123/ac147/archived_issues/ipj_7-3/ipj_7-3.pdf`

2.1.5 Application

- Pour des transferts de fichiers, quels comportements observez-vous lors du transfert de nouveaux fichiers comparé à la modification de fichiers existant ? Si l'application supporte l'envoie de plusieurs fichiers en même temps, comment agence-t-elle l'envoie de ces fichiers ?
- Pour du partage d'écran, comment l'application se comporte-t-elle lorsque plusieurs utilisateurs regardent le même écran ?
- Pour du partage d'écran, l'application se comporte-t-elle différemment lorsque le flux vidéo devient plus important (par exemple lorsque l'écran subit beaucoup de modifications) ?
- Pour du partage d'écran, est-ce que la qualité vidéo s'adapte au débit de la connexion internet ?
- Quel est le volume de données échangées par l'application pour chacune de ces fonctionnalités ? Utilisez une base appropriée permettant la comparaison (par ex. par minute).
- Il y a-t-il des serveurs relais utilisés pour interagir avec un utilisateur ou les applications communiquent-elles directement ? Observez-vous autre chose lorsque deux utilisateurs sont sur le même réseau Wi-Fi [4] ?
- Est-ce qu'interagir avec un utilisateur se trouvant dans le même réseau Wi-Fi ou Ethernet a un impact sur la façon dont le trafic applicatif est transporté ? Il y a-t-il des serveurs relais ?
- Observez-vous des requêtes HTTP ou autres protocoles spécifiques à un navigateur ? Quelles sont les utilités de ces protocoles ?
- Si vous identifiez d'autres protocoles que DNS et HTTP par dessus les protocoles de transport, justifiez l'utilisation de ces protocoles de transport.

3 Evaluation et délivrables

Vous serez évalué·e·s à l'oral. Pour cela vous devez préparer une courte présentation de 3 minutes, appuyée par des slides, donnant un aperçu de vos résultats sur l'analyse de l'application reprenant les différents points énoncés ci-dessus (DNS, couche réseau, couche transport, chiffrement et application). Cette présentation sera suivie de 7 minutes de questions où nous rentrerons plus en détails dans votre analyse de l'application. Vous pouvez préparer des slides qui ne seront pas dans la présentation comme support à de potentielles questions, par exemple des schémas des différents *handshakes*.

Vous devrez soumettre vos slides finales sur Moodle pour le **14 mars 2025 à 20h**. Nous utiliserons ces slides pour les présentations orales. Les horaires de passage pour cette évaluation orale seront annoncés ultérieurement sur Moodle.

Critères d'évaluation. L'évaluation de votre présentation se fera majoritairement sur la pertinence et la profondeur de vos résultats d'analyse. L'objectif du projet est de vous faire analyser une application réelle sur différents aspects. Il est possible que votre application ne donne pas énormément de résultats sur un des aspects de la Section 2. Si c'est le cas, il est suffisant de pouvoir justifier son absence pour l'application en question. D'autres parties contiendront d'avantage de détails intéressants à relever, où nous vous demandons de rentrer plus en profondeur.

Vos éventuels (et conseillés) schémas de fonctionnement de l'application peuvent être proprement faits à la main et importés dans vos slides.

4. Le réseau `eduroam` isole les utilisateurs, observez plutôt ce qui se passe dans un réseau domestique.