Déboguer les applications réseau lorsque tout est chiffré (1/13)

Stéphane Bortzmeyer
stephane+cdl@bortzmeyer.org

Capitole du Libre, 19 novembre 2022

Avant

Avant

- Rien n'était chiffré,

Avant

- Rien n'était chiffré,
- Les bisounours et les licornes gambadaient ensemble dans des lacs de confiture de groseille,

Avant

- Rien n'était chiffré,
- Les bisounours et les licornes gambadaient ensemble dans des lacs de confiture de groseille,
- On faisait confiance au réseau,

Avant

- Rien n'était chiffré,
- Les bisounours et les licornes gambadaient ensemble dans des lacs de confiture de groseille,
- On faisait confiance au réseau,
- Et on déboguait avec tcpdump / Wireshark,

Avant

- Rien n'était chiffré,
- Les bisounours et les licornes gambadaient ensemble dans des lacs de confiture de groseille,
- On faisait confiance au réseau,
- Et on déboguait avec tcpdump / Wireshark,
- Remarquables outils pour apprendre le réseau, d'ailleurs.

Depuis

Depuis

- Edward nous a prévenu que des méchants espionnaient,

Depuis

- Edward nous a prévenu que des méchants espionnaient,
- Même les jenairiencacher ont compris,

Depuis

- Edward nous a prévenu que des méchants espionnaient,
- Même les jenairiencacher ont compris,
- On chiffre tout, et à juste titre,

Depuis

- Edward nous a prévenu que des méchants espionnaient,
- Même les jenairiencacher ont compris,
- On chiffre tout, et à juste titre,
- Les derniers bastions du non-chiffrement cèdent :
  - DNS avec DoT, DoH et DoQ,
  - La couche transport, avec QUIC.

Depuis

- Edward nous a prévenu que des méchants espionnaient,
- Même les jenairiencacher ont compris,
- On chiffre tout, et à juste titre,
- Les derniers bastions du non-chiffrement cèdent :
  - DNS avec DoT, DoH et DoQ,
  - La couche transport, avec QUIC.
- Tout ce qui n'est pas chiffré pourra être utilisé contre vous (RFC 8546 sur la *wire image*).

Alors, comment on fait ?

On peut lutter contre le chiffrement

On peut lutter contre le chiffrement

- Les gens qui réclament de la « visibilité »,

On peut lutter contre le chiffrement

- Les gens qui réclament de la « visibilité »,
- Les affaiblissements délibérés (comme le ETLS de l'ETSI),

On peut lutter contre le chiffrement

- Les gens qui réclament de la « visibilité »,
- Les affaiblissements délibérés,
- La diabolisation de DoH,

On peut lutter contre le chiffrement

- Les gens qui réclament de la « visibilité »,
- Les affaiblissements délibérés,
- La diabolisation de DoH,
- « Nous sommes pour le chiffrement, mais » (discours de l'Internet Watch Foundation),

On peut lutter contre le chiffrement

- Les gens qui réclament de la « visibilité »,
- Les affaiblissements délibérés,
- La diabolisation de DoH,
- « Nous sommes pour le chiffrement, mais »,
- Combat d'arrière-garde ?

On peut demander à l'application de donner ses clés

On peut demander à l'application de donner ses clés

- Rappel de cryptographie « hybride » : la cryptographie asymétrique ne sert qu'à l'échange de clés, on fait de la cryptographie symétrique après, avec cette clé de session,

On peut demander à l'application de donner ses clés

- Rappel de cryptographie « hybride » : la cryptographie asymétrique ne sert qu'à l'échange de clés, on fait de la cryptographie symétrique après, avec cette clé de session,
- L'application peut exporter cette clé, un logiciel d'analyse du trafic peut ensuite l'utiliser pour déchiffrer,

On peut demander à l'application de donner ses clés

- Rappel de cryptographie « hybride » : la cryptographie asymétrique ne sert qu'à l'échange de clés, on fait de la cryptographie symétrique après, avec cette clé de session,
- L'application peut exporter cette clé, un logiciel d'analyse du trafic peut ensuite l'utiliser pour déchiffrer,
- `SSL_CTX_set_keylog_callback` avec OpenSSL,

On peut demander à l'application de donner ses clés

- Rappel de cryptographie « hybride » : la cryptographie asymétrique ne sert qu'à l'échange de clés, on fait de la cryptographie symétrique après, avec cette clé de session,
- L'application peut exporter cette clé, un logiciel d'analyse du trafic peut ensuite l'utiliser pour déchiffrer,
- `SSL_CTX_set_keylog_callback` avec OpenSSL,
- Wireshark sait relire ces clés (`tls.keylog_file: /tmp/mykeys` dans la configuration),

On peut demander à l'application de donner ses clés

- Rappel de cryptographie « hybride » : la cryptographie asymétrique ne sert qu'à l'échange de clés, on fait de la cryptographie symétrique après, avec cette clé de session,
- L'application peut exporter cette clé, un logiciel d'analyse du trafic peut ensuite l'utiliser pour déchiffrer,
- `SSL_CTX_set_keylog_callback` avec OpenSSL,
- Wireshark sait relire ces clés,
- Attention, c'est à l'application de faire ce qu'il faut, la bibliothèque TLS ne le fait pas (curl le fait mais pas openssl),

On peut demander à l'application de donner ses clés

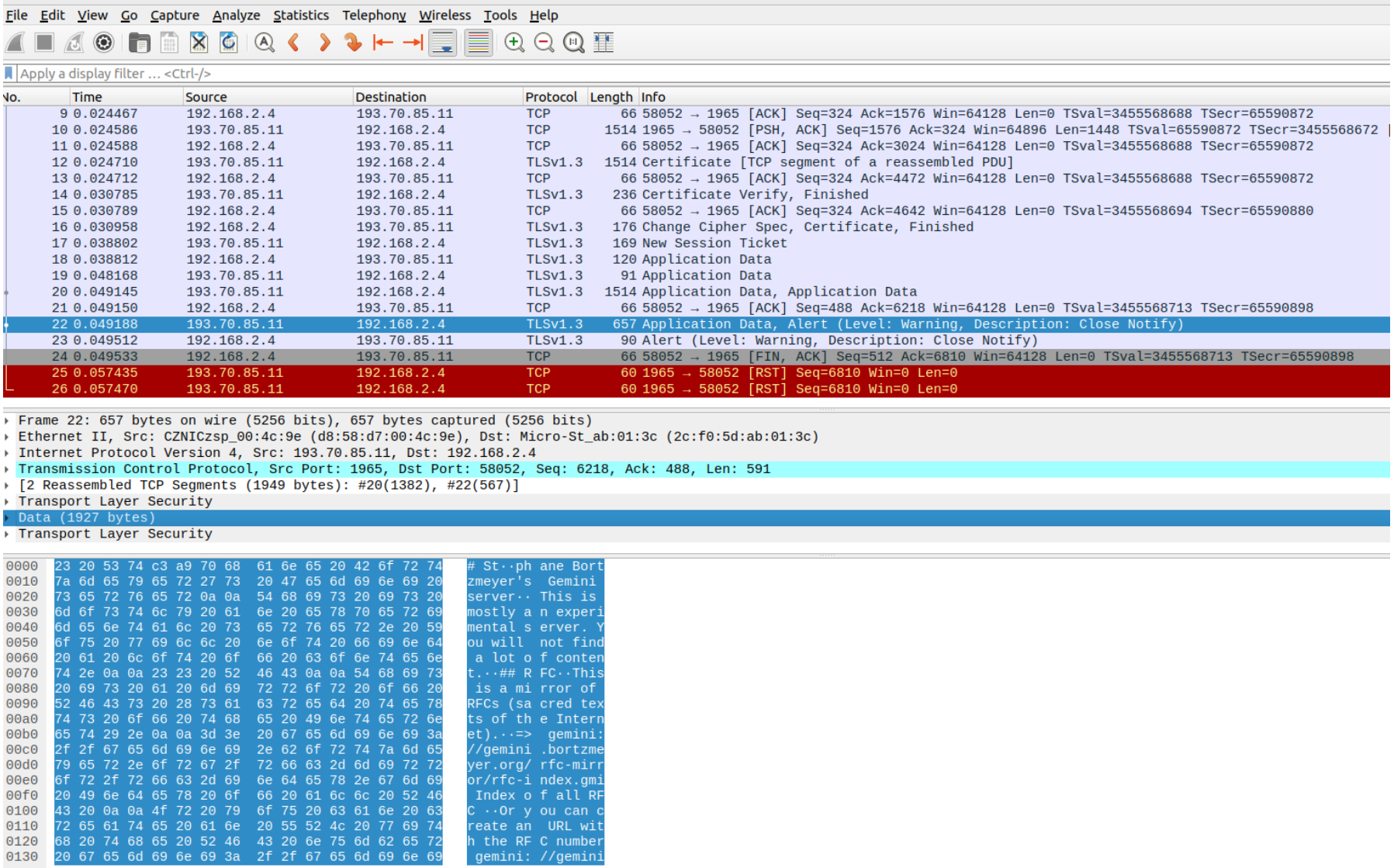
- Rappel de cryptographie « hybride » : la cryptographie asymétrique ne sert qu'à l'échange de clés, on fait de la cryptographie symétrique après, avec cette clé de session,
- L'application peut exporter cette clé, un logiciel d'analyse du trafic peut ensuite l'utiliser pour déchiffrer,
- `SSL_CTX_set_keylog_callback` avec OpenSSL,
- Wireshark sait relire ces clés,
- Attention, c'est à l'application de faire ce qu'il faut, la bibliothèque TLS ne le fait pas (curl le fait mais pas openssl),
- Moyen le plus courant de le demander : la variable d'environnement `SSLKEYLOGFILE`.

Exemple de code Python

```
def write_keys(conn, keys):
    keylogfile.write(keys.decode() + "\n")

if os.environ["SSLKEYLOGFILE"]:
    keylogfile = open(os.environ["SSLKEYLOGFILE"], "a")
    context.set_keylog_callback(write_keys)
```

Dans Wireshark



On peut demander à l'application de dire ce qu'elle fait

On peut demander à l'application de dire ce qu'elle fait

- L'application peut aussi aider, en étant bavarde,

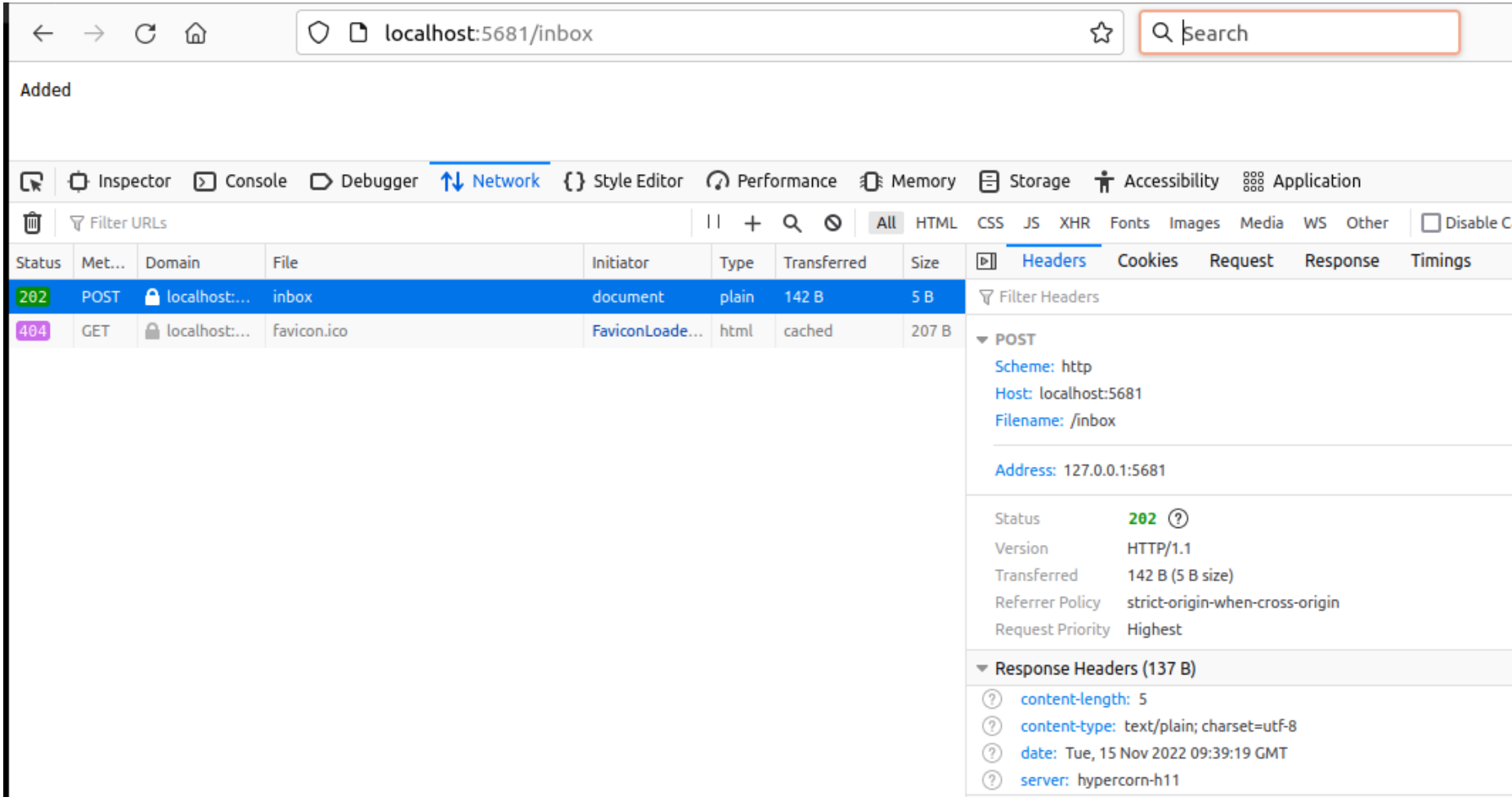
On peut demander à l'application de dire ce qu'elle fait

- L'application peut aussi aider, en étant bavarde,
- `curl --verbose` est un modèle, `curl --trace` (ou `--trace-ascii`) est encore plus détaillé,

On peut demander à l'application de dire ce qu'elle fait

- L'application peut aussi aider, en étant bavarde,
- `curl --verbose` est un modèle, `curl --trace` est encore plus détaillé,
- Firefox et ses *Web Developer Tools* aussi.

Firefox nous aide



Encore mieux si c'est normalisé

Encore mieux si c'est normalisé

- Les solutions précédentes sont pour examen par un humain, et sont spécifiques à une application,

Encore mieux si c'est normalisé

- Les solutions précédentes sont pour examen par un humain, et sont spécifiques à une application,
- Il y a un format normalisé pour exporter dans un fichier le dialogue, qlog (créé à l'origine pour QUIC).

Encore mieux si c'est normalisé

- Les solutions précédentes sont pour examen par un humain, et sont spécifiques à une application,
- Il y a un format normalisé pour exporter dans un fichier le dialogue, qlog.
- qlog est un format abstrait, qui peut se retrouver instancié en, par exemple, JSON.

Encore mieux si c'est normalisé

- Les solutions précédentes sont pour examen par un humain, et sont spécifiques à une application,
- Il y a un format normalisé pour exporter dans un fichier le dialogue, qlog.
- qlog est un format abstrait, qui peut se retrouver instancié en, par exemple, JSON.
- qlog est générique et peut s'appliquer à plusieurs protocoles.

Et avec une brosse à dents connectée ?

Et avec une brosse à dents connectée ?

- Comment savoir ce que votre brosse à dents raconte à son maitre ?

Et avec une brosse à dents connectée ?

- Comment savoir ce que votre brosse à dents raconte à son maitre ?
- (Ça a été utilisé comme argument anti-DoH.)

Et avec une brosse à dents connectée ?

- Comment savoir ce que votre brosse à dents raconte à son maitre ?
- (Ça a été utilisé comme argument anti-DoH.)
- Le fond du problème est plutôt le logiciel privateur.

En résumé

En résumé

- Wireshark reste super, mais il faut aussi enseigner / pratiquer le déboguage par la coopération de l'application,

En résumé

- Wireshark reste super, mais il faut aussi enseigner / pratiquer le déboguage par la coopération de l'application,
- Toute application réseau devrait fournir des outils de déboguage du trafic,

En résumé

- Wireshark reste super, mais il faut aussi enseigner / pratiquer le déboguage par la coopération de l'application,
- Toute application réseau devrait fournir des outils de déboguage du trafic,
- Il faut évidemment exiger du logiciel libre,

En résumé

- Wireshark reste super, mais il faut aussi enseigner / pratiquer le déboguage par la coopération de l'application,
- Toute application réseau devrait fournir des outils de déboguage du trafic,
- Il faut évidemment exiger du logiciel libre,
- Car sinon, on ne peut pas avoir confiance, même si le logiciel privateur produit une trace.