Kommunikations-Verschlüsselung

Die Verschlüsselung jeglicher Kommunikation spielt in unseren Anwendungsfällen eine essentielle Rolle. In diesem Artikel wollen wir erklären, was mit Kommunikationsverschlüsselung gemeint ist, welche Arten es davon gibt und welche Vor- bzw. Nachteile sie haben.

Wir unterscheiden hier zwischen Transportverschlüsselung und Ende-zu-Ende-Verschlüsselung (E2EE: End-To-End-Encryption). Vorwegnehmend lässt sich schonmal sagen, dass Transportverschlüsselung nice-to-have ist, für uns aber in keiner Weise ausreicht und wir deshalb immer E2EE wollen.

Transportverschlüsselung

Transportverschlüsselung wird allgemein mit SSL bzw. TLS realisiert. Das kennt ihr zum Beispiel aus eurem Browser, wenn neben der URL ein Vorhängeschloss erscheint und vor der URI `https` steht. Kommt das nicht zum Einsatz, steht dor nur `http` (und meistens erscheint eine Warnung, dass die Verbindung nicht gesichert ist).

Um die Transportverschlüsselung zu erklären, nutzen wir unten stehende Grafik. Anna will Arthur eine Nachricht übermitteln, bspw. per Email. Das Beispiel funktioniert auch mit anderen Diensten ohne E2EE, wie: Telegram, Discord, oder Chats in Spielen. Dabei gäbe es aber nur einen anstatt zwei Servern.

Hier also das Beispiel mit Email.

Anna hat eine Email-Adresse bei dem gelben Server, hier `systemli.org`. Ihre Mail lautet also `anna@systemli.org`

Arthur hat eine Email-Adresse beim roten Server, hier `riseup.net`. Seine Mail lautet also `arthur@riseup.net`

Weil wir ja von Transportverschlüsselung reden, benutzen beide keine E2EE. Das heißt, weder hat Anna einen PGP-Key von Arthur, noch anders herum!

Die Schlüssel und Schlösser symbolisieren sogenannte **Zertifikate** (Vorhängeschlösser). Jeder Server hat sein eigenes Zertifikat, mit dem die Kommunikation mit ihm verschlüsselt (also eingeschlossen) werden kann. Nur der Server im Besitz des Zertifikats, hat auch den zugehörigen

Schlüssel.

Wenn Anna jetzt eine Mail schreiben will, holt sie sich das Zertifikat von Systemli (gelbes Schloss) und verschlüsselt damit ihre Mail. Völlig unabhängig davon, an wen die Mail am Ende gehen wird! Arthurs Empfangsadresse steht dann draußen auf dem Umschlag, wie bei normaler Post auch. Diese Mail (gelber, verschlossener Umschlag mit Schloss) geht dann zum Systemli-Mailserver (gelber Kasten).

Der Systemli-Mailserver schließt nun, die mit seinem eigenen Zertifikat verschlüsselte Mail auf und scant sie z.B. nach Spam. Vor allem schaut er sich die Empfangsadresse auf dem Umschlag an: `arthur@riseup.net`. An der Stelle hinter dem `@` erkennt der Server, an welchen Mailserver er diese Mail nun weiterleiten muss: `riseup.net` (roter Kasten). Also geht er kurz rüber zu Riseup, schnappt sich eine Kopie des ihres Zertifikats und verschlüsselt damit Annas Email wieder und schickt sie so (roter, verschlossener Umschlag mit Schloss) an den Riseup-Mailserver.

Ab hier wiederholt sich dieser Vorgang so lange, bis die Mail Arthur erreicht. Der Riseup-Server packt die Mail aus und wieder ein und schickt sie schließlich an Arthur.

Grafik Ende-zu-Ende Verschlüsselung

Problem

Hier der Verweis auf die Bedrohung Verkehrsdatenüberwachung/TKÜ.

Das Problem hierbei ist offensichtlich. Jede*r Teilnehmer*in in der Kommunikationskette kann die Mail einfach öffnen und lesen. Zusätzlich bleiben bei vielen Anwendungen (wie oben aufgezählt) die Nachrichten auf den (Mail)-Severn als Kopie liegen.

Ende-zu-Ende Verschlüsselung

Wenn ihr die Bedrohung durch Transportverschlüsselung verstanden habt, ergibt sich die Ende-zu-Ende-Verschlüsselung schon fast von selbst.

1. & 2.) Anna besorgt sich das Schloss (public-key) von Arthur. Dieser Punkt ist sehr wichtig, beachtet dazu den Absatz [TOFU]!

3.) Anna verschlüsselt ihre Nachricht mit Arthurs public-key

4.) Die Nachricht bleibt in allen Teilschritten von 4 (a-e) verschlüsselt. Lediglich die Metadaten (bspw. Absende-/Empfangsadresse) darauf sind (an allen möglichen Stellen, also auch beim Transport!) sichtbar und werden von den Servern gelesen, um die Mail weiter zu leiten.

5.) Arthur empfängt seine Nachricht. Weil die Nachricht mit seinem Vorhängeschloss verschlüsselt wurde und er gut auf seinen Schlüssel (private-key) aufgepasst hat, kann nur er die Nachricht mit seinem Schlüssel wieder entschlüsseln.

Grafik Ende-zu-Ende Verschlüsselung

TOFU ist böse

Trust On First Use

Schlüssel muss "out of band" verifiziert werden. Eine unverschlüsselte (also transportverschlüsselte) Mail ermöglicht das Austauschen der öffentlichen Keys.

Grafik machine-in-the-middle attack

Zu den Gefahren von Transportverschlüsselung siehe Verkehrsdatenüberwachung.

Version #3
Erstellt: 14 Januar 2025 18:53:35 von ESC-IT Migration Bot
Zuletzt aktualisiert: 15 Januar 2025 12:40:41 von ESC-IT Migration Bot