The use of TLS in Censorship Circumvention

R07944058
網媒碩一 陳鵬宇

1 Summary

此篇論文對 TLS connection 進行了測量與統計上的分析。他提出了一些評測方式來比較用了不同規避工具（circumventing tools）的情況，其中有些工具較為突出，他們更易於阻擋不希望的連線。

TLS 是審查規避工具的主流使用的協議，包括 Tor、Signal 等。本文收集並分析了 9 個月內超過 118 億個 TLS connection 的真實 TLS 流量，依此來確定實際在 Internet 使用的各種 TLS 是如何在客户端實現的。作者使用大量數據來分析幾種流行的審查規避工具是如何實現 TLS，包括 Lantern、Psiphon、Signal 等。大部分的工具都使用 TLS 設定，這些設定很容易與他們試圖模仿的實際流量區分開來。

為了解決這個問題，作者也針對 Golang 既有的 libray `crypto/tls` fork 後開發了自己的 `utls`，它使工具維護者能夠自動模仿其他流行的 TLS 實作方式。並宣稱他們的工具能夠降低許多人為手動的 work，更靈活地適應動態的 TLS 生態。

2 Strength(s)

本文很詳細地敘述了 TLS 的機制，像是當 client 端一開始要和 server 端建立 handshake 時的各個流程，ClientHello 的訊息必需包含：TLS Versions、Cipher Suites、Compression Methods、Extension IDs 等資訊，而這些資訊會形成 fingerprint。Censor 的目標是希望「只」阻擋下規避工具，而 fingerprinting TLS 連線希望能最小化附帶傷害（collateral damage），同時越不流行的 fingerprint 越容易被 block。

對於現今規避工具最大的挑戰為，工具必需妥善地模仿、使用流行的 TLS 實作方式、不時地追蹤最新的更動，並更新以確保自己不會過時，但模仿總是困難的。

文中對於 measurement 提出很好的分析方式，作者藉由大學收集了匿名的 TLS "fingerprints"，並比較了 fingerprints 和規避工具的 fingerprints，雖然數據不見得能代表什麼，但仍顯示出總體的趨勢。

3 Weakness

本文重點在於分析他們所蒐集到的資料，但其中卻在兩個時間有明顯的缺漏，這樣無法獲得連續的觀察。並且全篇大多側重在 fingerprint 的分佈，若能夠針對文中所提到的一些工具的實作分式做更進一部的説明會更好。

同時對於 TLS 是傳送明文這些事可能會導致 Man in the Middle attack 也可以考慮進去，一但受到 MitM 攻擊，這樣無法確定 ServerHello 是不是是因為收到 Eve 所偽造的 ClientHello，而誤傳了錯誤的 ServerHello 給 client。

4 Reflection

TLS 是基於 TCP/IP 的一種協定，其中最重要的部分為一開始建立 handshake 時，整個協定用到了許多前面課程所教授的知識，像是數位簽章認證，公私鑰交換等，而由於這些訊息都是以明文的方式傳遞，因此檢查 integrity 變得很重要。

Cipher Suite 可以將他想像成 key exchange、signature、encryption、hash function 的一大包資訊，其中有許多選項可供選擇。

本文花了很多時間在比較 Postgres 資料庫所蒐集到的資訊，並透過將分析工具、data 開源讓大家都能使用是滿不錯的。