# Blocking-resistant communication through domain fronting

## ✧ Summary

本篇旨在利用 Domain Fronting 技術，令真正訪問的域名與所看到的域名不同，進而隱藏真實位址。關鍵的地方便在於不同的通訊層中，使用不同的域名。在 http(s)請求中，目標域名通常會標示於三個地方：DNS query、TLS Server Name Indication(SNI)、HTTP Host header。在一般情況此三處皆會是欲訪問的域名，然而在 Domain fronting 中，前兩者做為 front domain 會攜帶公開且允許的域名，而在 HTTP Host header 則會攜帶另一個應被禁止訪問的域名。

通常 censor 無法 block DNS 與 SNI request 內容，而 Host Header 因為加密的緣故，sensor 無法得知，但在 frontend server 是可以得知 request 的真正目的地，再行內部轉送，因此**透過 domain fronting 便可對 censor 隱藏真正的目的地**。

## ✧ Strengths

Domain Fronting 通常適用於 Content delivery network(**CDN**)，最大優勢在於能有效解決 censorship circumvention 會遭遇的挑戰，包含 Address blocking、Content blocking (deep packet inspection, DPI)以及 Active probing。

傳統的 address 或 domain blocking 可能會將一些有價值或善意的也 block 掉，或是 censor 可能會需要極大資源才能做到完善。而在 content-based 部分，因為由 HTTPS 加密，因此是無法透過 keywords 來 block 的，除非直接激進地禁止所有 HTTPS 流量。Active probing 方面，雖 censor 可能得以發現 web service 被用去進行 circumvention，但無法直接將該 service block 住，否則很可能會招致相當大的 collateral damage。

## ✧ Weaknesses

在 CDN 方面，未必皆會透過 Host header domain 來 forward request，很可能還會需要輔以其他封包資訊(如:簡易認證、來源、時間等資訊)，而或者所連結的 domain 或 IP 會隨時間改變。此外，censor 也有可能會禁用 HTTPS，或甚至能 control certificate authority，從而進行 man-in0the-middle HTTPS connections，移除加密並偵測 host header。

在另一方面，雖一般 blocking 方法無法有效杜絕 Domain Fronting，但若透過流量分析應是另一種可能性。不僅是 packet length 分布或是 connection lifetime，其他如 latency、burst、patter 等等，皆可做為分析資訊，甚至輔以 AI、深度學習等方式，將能有效判別出一般流量與 circumvention traffic，縱使欲偽裝流量或訊息仍會有相當難度。

## ✧ Reflection

本文提出一個相當簡單但卻有效的 idea，甚至也證實在諸多 CDN 服務如 Google app engine、amazon CloudFront、Azure、CloudFlare、Akamai、Fastly 等皆能成功。

然而 censor 實際上會依據不同時間、供應商而會有不同的 blocking policy，當時能成功也未必以後皆有效。又或者 censor 與 operator 合作，便可更容易禁用 domain fronting。

當然，太強太嚴格的 censor，也會更容易 block 到 valuable service，且所需要的 resource 也會更多，端看 policy 該如何權衡輕重。