03/15 - Anonymity

How does Tor fit into anonymity?

- Want sender anonymity (mail, no return address), receiver anonymity (don't know who received the mail)

Tor - fun historical points

- US Navy Invention
- Initially rejected as a paper, eventually won test of time award
- 3M million estimated users (!)

Mixed Nets - early proposal

- Batch up a bunch of messages
    - Random reorder -> re-encrypt
    - "Remailer"
    - Downside - latency

Tor

- No batching -> low latency!
- Distributed relays
    - Low latency in exchange for lack of protection against a global eavesdropper

Potential weaknesses:

- Traffic confirmation
    - Correlated messages coming in and out of the network
    - Alice sends at (T+1, T+3), bob receives at (T+4, T+6)
    - Could be mitigated
        - Constant bitrate (destroy timing information)
            - But really expensive
        - Selecting paths so they are not in the same country, heuristical, but makes it harder to spy on
    - Fun historical fact (CMU)
        - Silk road
            - Set up relays, insert invisible tags and attempt to be first and last replay in scheme
            - Satisfying defense is hard
    - NSA: attack endpoint using vulnerability in firefox
        - Find TOR connections (easy)
            - Mount man in the middle attack
            - Fox acid : imposter website
                - Client side security is important
            - Potentially fix (restrict TOR browser to https websites)
- Potentially hard to deal with national spy agencies

Censorship (a story of a cat and mouse)

- Imagine you have keywords in unencrypted search
    - Could ask the search engines to turn off encryption

- Users could switch search providers
    - Nations could block those search engines (e.g., china + google)
- VPN
    - Nations could find VPNS
        - Look at packets and look for signatures
- Users may uses Tor
    - Nations may attempt to find Tor connections
    - Find onion routers
        - Tor may try to generate non-listed bridge nodes
            - Not publicly listed
            - IP Scanning
                - Send Tor bridge type request to all IP addresses
- Domain fronting aka decoy routing
    - Quirk of TLS handshake
        - Web address is sent twice, one encrypted in TLS handshake
        - Once in the clear in handshake
    - Only in host header, use real website, in the clear, use some dummy address
    - April, 2018: Telegram (real world)
        - Was blocked in Russia
            - Telegram resorted to using Google/Amazon CDNs with domain fronting
                - Russia responds by blocking Google and Amazon CDNs (wow)
        - Collateral damage
            - Protect anonymity by making the only way to shut things down is to cause too much pain
        - One CDN exists at the moment that allows domain fronting (microsoft azure)
            - Could go down at any point, major blow to domain fronting defense

Discussion (Bock et al.)

Nation-state-level censors

- Powerful entities able to inspect, inject, and/or drop traffic throughout countries
    - Two broad methods: on-path or in-path
    - On Path: censor obtains copies of packets, inject packets that end-hosts accept, such as TCP RSTS to tear down connections
    - In Path: man-in-the-middle, can simply drop packets altogether or hijack connection

Current Evasion Methods

- Existing methods rely on packet-manipulation strategies: alter and/or inject insertion packets at one endpoint (processed by censor only):
    - to de-synchronize censor’s state (eg. thinks connection is down)
    - confuse censor into not recognizing censored keywords through segmentation

- All prior work rely on some amount of client side evasion

Goal: Have servers outside censoring regime to help clients evade censorship without clients having to install any extra software

Geneva:
- Use genetic algorithm Geneva to automatically discover packet-manipulation strategies that evade censorship
    - Composes of 5 building blocks: duplicate, fragment, tamper, drop, send
    - Trains against censors by being run from within censoring nation-state
    - Authors extend Geneva to be purely server-side and apply to other protocols beyond HTTP
- Evaluated in China, India, Iran, Kazakhstan across five protocols (DNS, FTP, HTTP, HTTPS, SMTP) by running Geneva server-side
- Interesting findings:
    - Some evasion strategies succeed some of the time by exploiting bugs in censor synchronization
    - Although the strategies operate at TCP level, the success rates vary depending on the higher-layer application -- Great Firewall handles different protocols differently