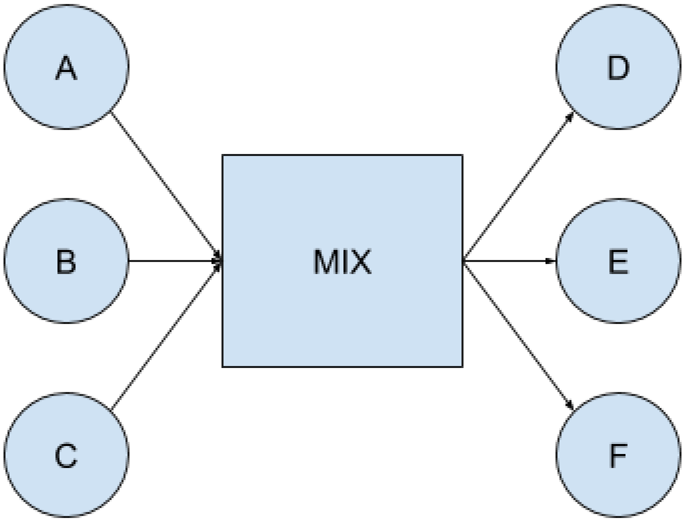
Networks & Mobile Systems - Network censorship

Guest lecture by Joe Bonneau

A precursor to Tor

- In the 1990s, MIX networks were used as an attempt to anonymize traffic
- In the diagram below, senders A, B, and C deliver their packets to a MIX
- The MIX stores these packets, or cells, in a buffer, and sends them out in a random order to the destination D, E, and F
- The same cells enter and leave the MIX, but we do not know which pairs are communicating



- Some issues with this system include:
    - High latency: the MIX doesn’t send out cells until there are enough on its queue. This causes a delay in cells being sent out
    - Not fully private: the MIX always knows communicating pairs, so if a malicious third party were to become a MIX, that would defeat the whole purpose of the system

Tor

What is it?

- Tor is a network designed to protect its users' privacy
- It allows for the anonymous transport of TCP streams

How does it work?

- Network is a set of nodes acting as relays for communication streams
- Tor nodes ensures that the correspondence between incoming data streams and outgoing data streams is obscured from the attacker
- An attacker cannot be sure about which of the originating user streams corresponds to an observed output of the network
- Connection establishment procedure:
    - Each node has:
        - A routing table
        - A queue for each outgoing path
        - You can get a weak form of mixing if multiple queues exist on a particular node (however Tor tries to avoid this for the sake of latency)
    - In order to form a "circuit," we must pick a node at random from the network, and establish a secure connection with it
        - Circuits last on the order of minutes to hours
    - Through this circuit, we can communicate with all other nodes in the network
    - Each additional node provides an additional layer of encryption
    - Each layer is decoded by a Tor node and the data is forwarded to the next Onion router using standard route labelling techniques
    - The initiator can ask the last Tor node on the path to connect to a particular TCP port at a remote IP address or domain name
- Tor does not do any explicit mixing, and therefore forwards packets immediately, leading to improved latency

What are some other special features of Tor?

- Any Tor node is involved in thousands of circuits at any given moment
- Tor implements a token bucket strategy to make sure that long-term traffic volumes are kept below a specified limit set by each Tor node operator
- Each stream has two windows associated with it, the first describes how many cells are to be received by the initiator, while the other describes how many are allowed to be sent out to the network. If too many cells are in transit through the network – and have not already been accepted by the final destination – the Tor node stops accepting any further cells until the congestion is eased
- Hidden servers:
    - Tor browsers resolve .onion domains in a special way
    - Only certain entry nodes store where hidden servers are
    - Global public lists maintain a list of all nodes in the network

How can we attack a Tor network?

- **Objective 1**: link the initiators of connections with their respective communication partners
- **Objective 2**: link each transaction with a particular initiator
- Conventional anonymous systems protect against a global passive adversary, but Tor protects against a non-global adversary
- Tor also does not attempt to protect against traffic confirmation attacks (adversary observes two parties that he suspects to be communicating with each other, to either confirm or reject this suspicion)
- Traditional traffic analysis
    - Goal is to infer information from network metadata, including the volumes and timing of network packets, as well as the visible network addresses they are originating from and destined for
    - One method for this is by looking only at times when users are initiating connections, and connections are being relayed to network services outside the Tor network
    - Another method is inspecting the traffic within the anonymous communication network, and further, the actual shape (load) of the traffic on each network link
- Traffic analysis of Tor

  - By routing a connection through specific Tor nodes, and measuring the latency of the messages, the adversary can get an estimate of the traffic load on the Tor node (shown in diagram below)
  - The initiator S sends traffic through Tor relays (TR) 1-3 to the corrupt destination server T. The attacker A can measure traffic at TR2

  - To reiterate, the goal of an attacker is, based on timing data from all nodes on the network, to identify which nodes are carrying the traffic with the pattern injected by the corrupt server
  - Two tests are performed:
    - stream from the corrupt server went through the target node
    - stream did not go through the target node
  - If the test was successful, the correlation between the traffic modulation and probe latency in the case where the victim stream did go through the target node should be higher than the case where it did not

The great firewall

Background information

- China has been using active probing to detect and block privacy tools
- The paper tries to explain these probing mechanisms and design methods for circumventing them
- Active probing:
  - passively monitoring the network for suspicious traffic, actively probing the corresponding servers, and blocking those determined to run circumvention services such as Tor

Some circumvention protocols

- Tor (discussed earlier)
- Obfs2
  - Provides obfuscation layer around Tor's TLS
  - entire communication looks like a uniformly random byte stream in both directions
  - first send a key, then send ciphertext encrypted with that key
  - **Major issue**: censor can simply read the first few bytes of every TCP connection, treat them as a key, and speculatively decrypt the first few bytes that follow
- Obfs3:
  - Builds on Obfs2
  - Main contribution: Diffie-Hellman negotiation that determines the keys to be used to encrypt the rest of the stream
  - Now, censor must either intercede in the key exchange (using a man-in-the-middle attack to learn the secret encryption keys), or settle for heuristic detection of random-looking streams

Inspection setup

- Shadow infrastructure:
  - Controlled Tor network not used by anyone
  - 6 servers in china: 2 w/ vanilla Tor, 2 with Obfs2, 2 with Obfs3
  - 6 Tor bridges outside China in EC2
  - 9 bridges as control (none of the chinese servers attempt to connect to these)
- Server log analysis:
  - server runs various common network services, including: HTTP, HTTPS, and SSH
  - application logs reveal that the server has been receiving active probes from China for over 2.5 years
- Counterprobing:
  - Active probers seem to share their IP address pools with normal Internet users

Analysis

- Is active probing successful at discovering Tor bridges?
  - Despite the Great Firewall having the ability to probe for obfs2 and obfs3, only vanilla Tor was blocked
- What is the delay between a connection attempt to a Tor bridge and subsequent active probing of the bridge?

  - censors maintained a “probing queue” that was processed every 15 minutes
- Where in the network are the probes coming from?
  - Nearly every probe is from a Chinese AS
  - However, a probe IP address is highly unlikely to be seen again
- What types of probers do we see?
  - TLS, Tor, Obfs2, Obfs3, AppSpot
- Do active probers have “fingerprints” that distinguish them from normal clients?
  - IP layer: no
  - TCP layer: yes (based on window scaling, permission of selective ACKs, TCP timestamp option)