1. **Data set used in paper for evaluation**
   1. GFW Evasion strategies and analysis, including forging of TCP, HTTP packets and TCB manipulation.
   2. Demonstrated methods in TCP DNS censorship evasion.
2. **Testbed/simulation/emulation experimental setup**
   1. Modeling GWF state with different TCP messages and flags in order to design evasion strategies.
   2. Proposed that traditional model of the GWF is incorrect/outdated and new testing was done to redefine how the GFW censors and intercepts connections.
      1. GFW now thought to enter a re-synchronous state when encountering TCP messages in order to correctly resynch its TCB with the client connection state. The authors propose that this can be exploited to bypass censorship black boxes and middleboxes
   3. New ways to avoid GFW
      1. Resync + Desync
         1. Sending an out of window sequence number to the GFW to cause it to resync after a 3-way handshake is performed. Then this is followed by the real request to the server
      2. TCB Reversal
         1. GFW only censors’ traffic from the client side to the server. The first SYN/ACK seen by the GFW is assumed to be from the server side of the connection to the client. IT creates a TCB for this case and then monitors the packets from the server to the client, mistakenly thinking it is monitoring packets from the client to the server.
            1. To exploit this the client sends a SYN/ACK insertion packet before sending a real request to the server

Care needs to be taken in the crafting of the packet in order to avoid the server inadvertently closing the connection.

1. **Algorithm proposed in paper.**
   1. Crafting insertion packets.
      1. The paper describes the types of insertion packets designed to evade the GFW. Each packet is not universally good as described in the paper, leading to multiple being attempted and designed.
      2. To find candidate insertion packets, the Linux TCP stack was analyzed as the server environment to determine which insertion packets will be ignored by common webservers.
   2. The authors describe a measurement driven censorship tool called INTANG to process packets for a connection and select the appropriate insertion packets to use for that network address. The program also contains logic to convert UDP DNS requests into TCP DNS requests, in an application transparent way, to better bypass DNS poisoning censorship strategies implemented by the GFW.
      1. <https://github.com/seclab-ucr/INTANG>
   3. New method achieves a near 90% success rate in GWF censorship bypassing.