Your state is not mine: a closer look at evading stateful internet censorship

Zhongjie Wang, Yue Cao, Zhiyun Qian, Chengyu Song, Srikanth V Krishnamurthy

University of California, Riverside

Background

Nov 13,

Times

LISTINGS & GUIDE

NEWS

Feb. 05

In

in

n Russia

World map showing the status of YouTube blocking

Internet censorship and surveillance by country (2018)[71][72][73][74][75]

US News CIVIC » Best Countries Best States Cities Healthiest Communities The Civic Report

HOME / CIVIC / OPINION / WORLD REPORT

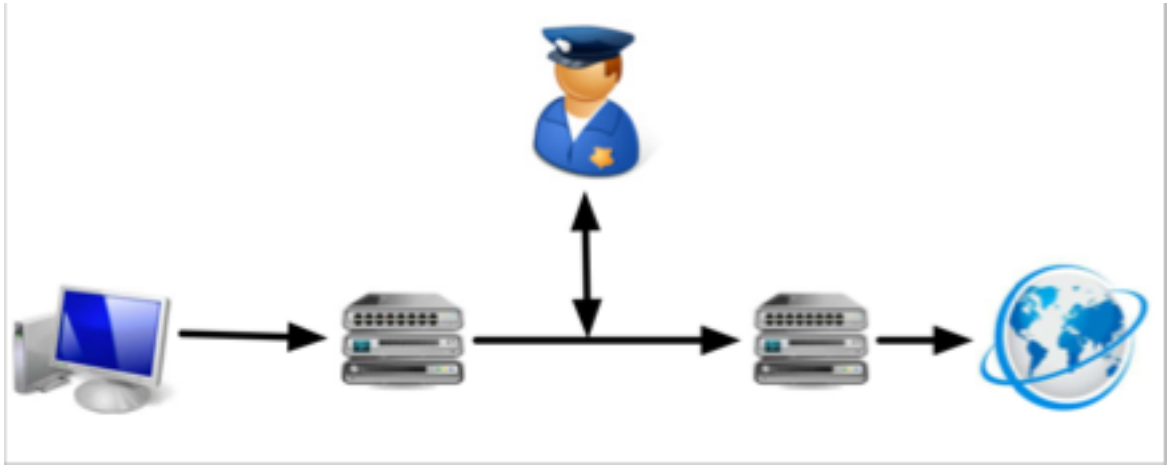
China's Newest Export: Internet Ce

China is teaching other countries how to control the Internet.

By **Mark C. Eades**, Contributor Jan. 30, 2014, at 4:03 p.m.

The Great Fire Wall (GFW)

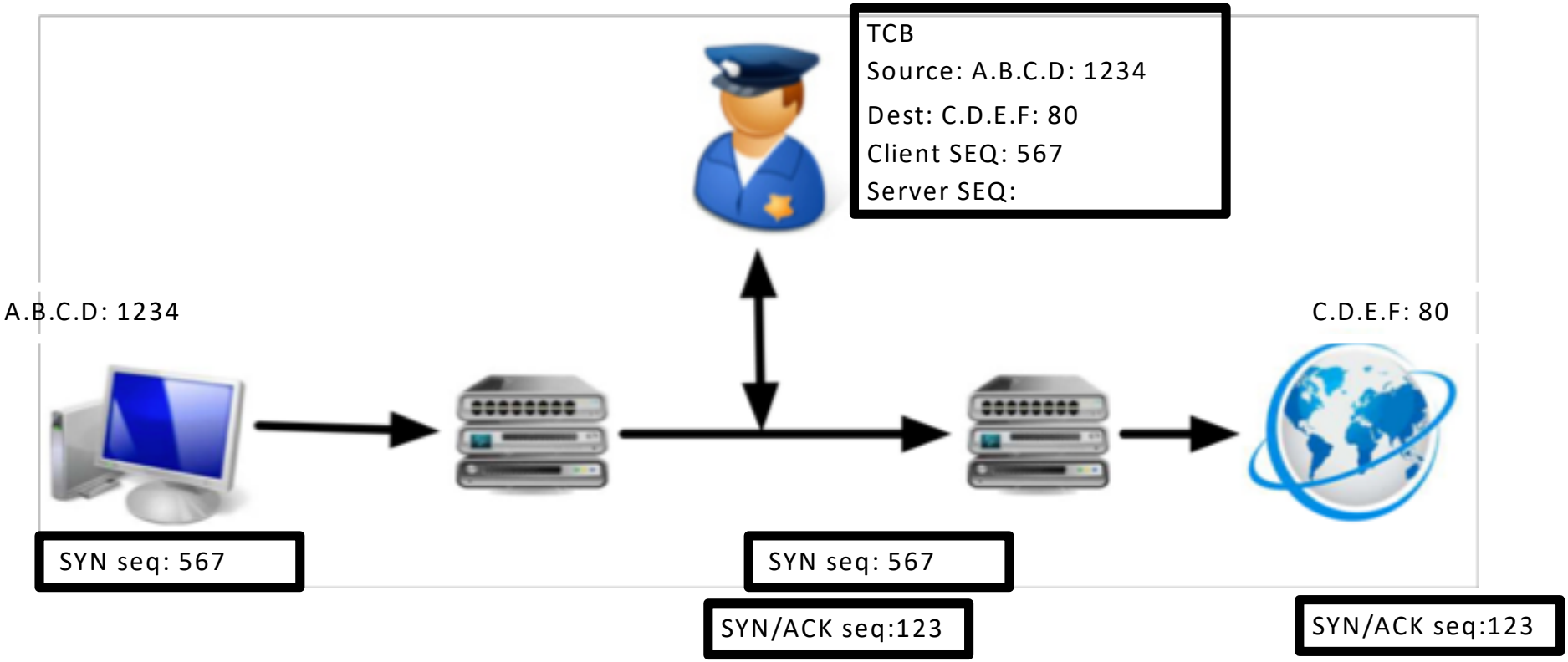
- A sophisticated censorship tool that performs:
  - Deep packet Inspection (DPI)
  - DNS pollution
  - IP blocking, etc

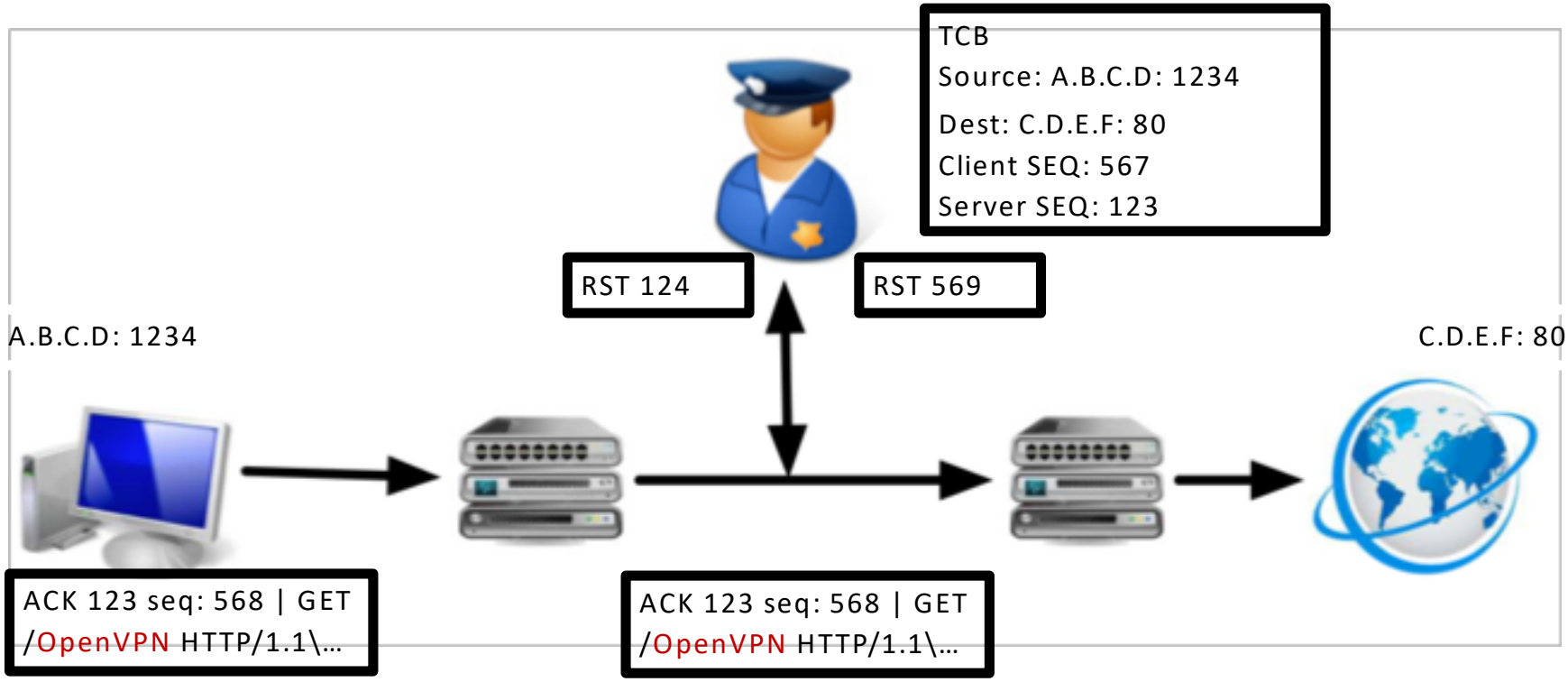


Deep Packet Inspection

- Reconstruct the TCP flow
- Examine contents of the flow for sensitive keyword ……GET /OpenVPN HTTP/1.1\…
- Inject RST and RST/ACK packets to both endpoints

- The censor need to maintain TCP Control Block (TCB) for each connection to track flow state


TCB
Source: A.B.C.D: 1234
Dest: C.D.E.F: 80
Client SEQ: 567
Server SEQ:
A.B.C.D: 1234
C.D.E.F: 80
SYN seq: 567
SYN seq: 567
SYN/ACK seq:123
SYN/ACK seq:123


TCB
Source: A.B.C.D: 1234
Dest: C.D.E.F: 80
Client SEQ: 567
Server SEQ: 123
RST 124
RST 569
A.B.C.D: 1234
C.D.E.F: 80
ACK 123 seq: 568 | GET /OpenVPN HTTP/1.1\...
ACK 123 seq: 568 | GET /OpenVPN HTTP/1.1\...

Challenges for DPI

- Diversity in host information -> Different TCP standards

- Diversity in network information -> No knowledge of packet losses

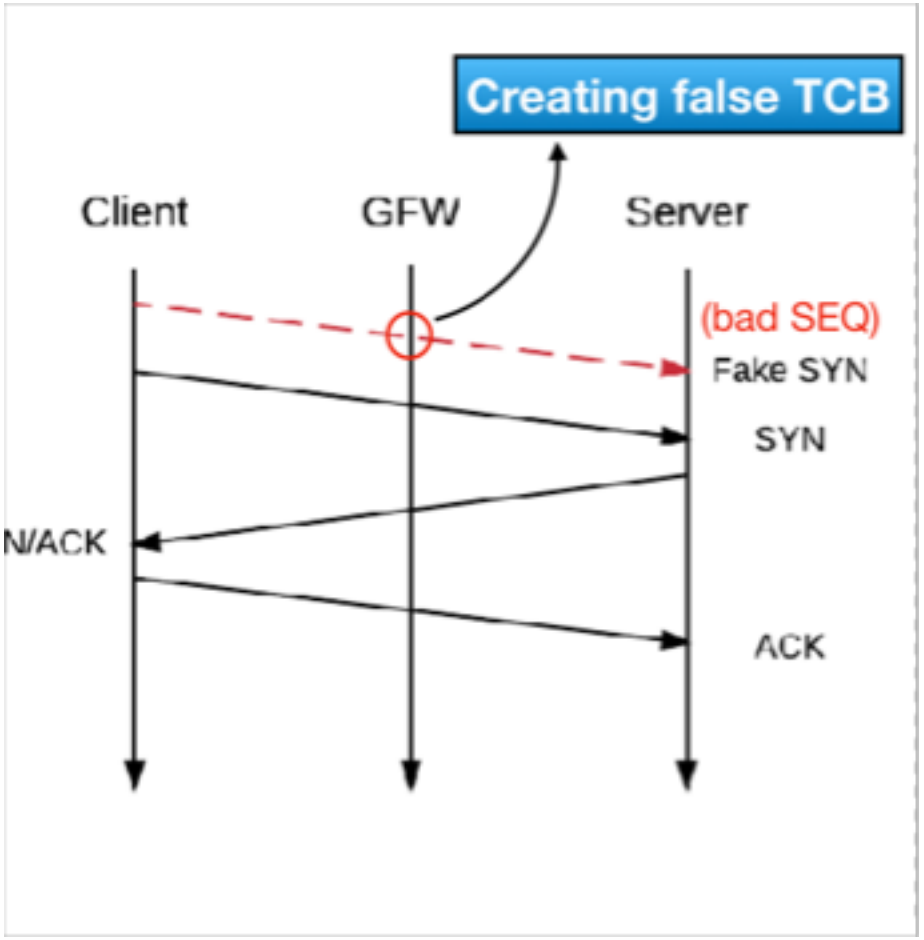
- Presence of middleboxes
  -> Packets might be altered/dropped by middleboxes after DPI process

=> Impossible to maintain an accurate state of a connection
Client can disrupt the state maintained by GFW

Existing Evading Strategies

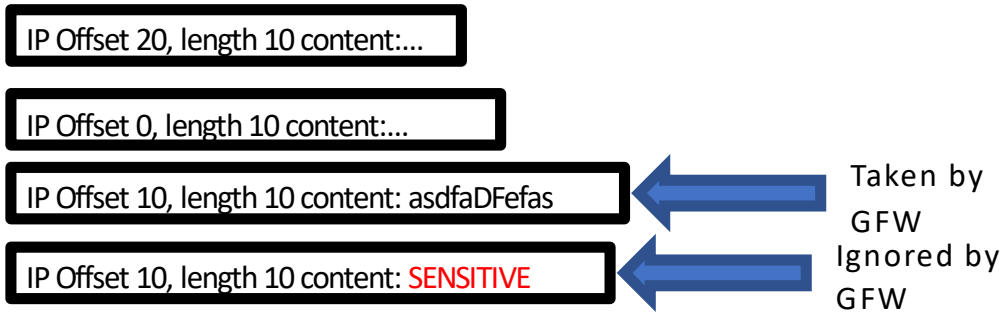
TCB Creation

- Assumption: GFW creates a TCB upon seeing a SYN packet.
- Strategy:
  - The client can send a SYN insertion packet with a fake SEQ to create a false TCB on the GFW
  - Then build the real connection.



Data Reassembly

- Out-of-order data overlapping
  - Assumption:
    - Two out-of-order IP fragments: the GFW prefers the former and discards the latter.
    - Two out-of-order TCP fragments: the GFW prefers the latter
  - Strategy:
    - Leave a gap in the data stream
    - Send 2 packets for that gap, one containing random data, the other containing real data

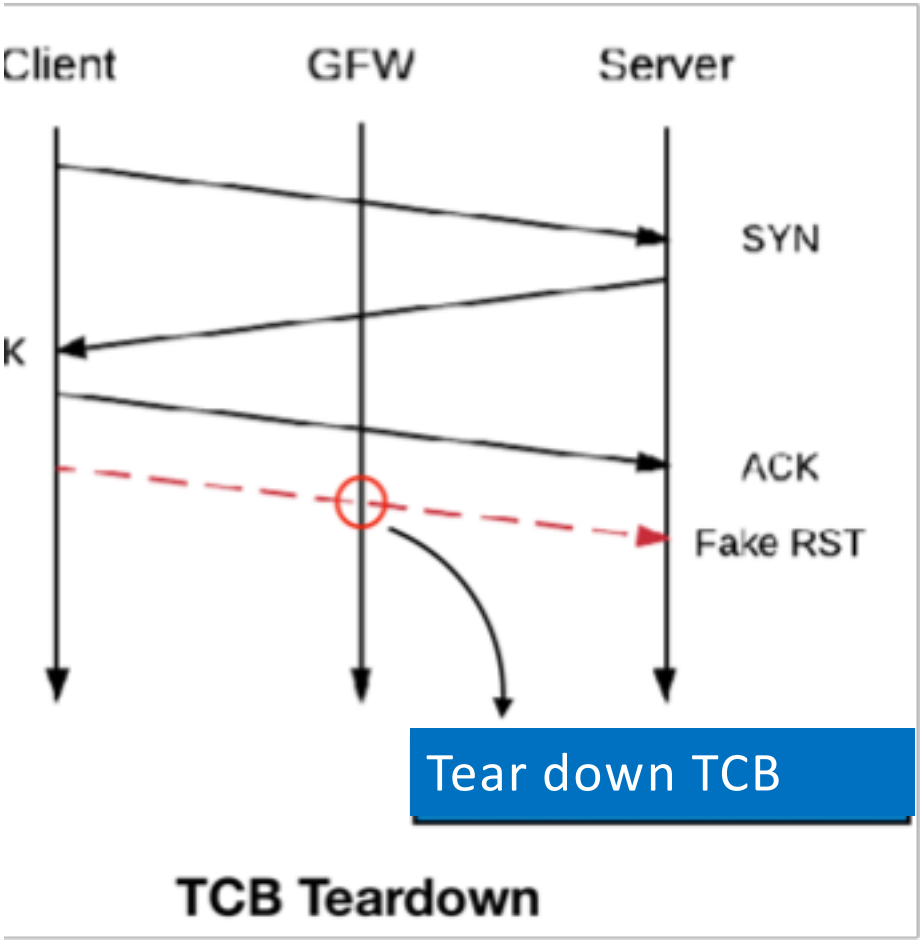


Data Reassembly

- In-order data overlapping
  - Assumption:
    - two in-order data packets: the GFW accepts the first one
- Strategy: Craft insertion packets that contain junk data to fill the GFW's receive buffer, while making them to be ignored by the server

TCB Teardown

- Assumptions:
  - GFW tear down TCB when seeing RST, RST/ACK, or FIN.
  - GFW only creates a TCB upon seeing a SYN packet
- Strategy: After handshake, send RST to tear down TCB while making it ignored by the server



Evaluation

- Set up
  - 11 Vintage points
  - 3 ISPs, 9 cities
  - 77 Alexa top global sites
  - HTTP requests
  - Sensitive keyword: ultrasurf
- Observation:
  - GFW has evolved
  - Heterogenous: Old model still exists

Failure1: no reps. from server

Failure2: RST from GFW

Strategy	Discrepancy	w/ sensitive keyword			w/o sensitive keyword	
		Success	Failure 1	Failure 2	Success	Failure 1
No Strategy	N/A	2.8%	0.4%	96.8%	98.9%	1.1%
TCB creation with SYN	TTL	6.9%	4.2%	88.9%	95.3%	4.7%
	Bad shecksum	6.2%	5.1%	88.7%	93.5%	6.5%
Reassembly out-of-order data	IP fragments	1.6%	54.8%	43.6%	45.1%	54.9%
	TCP segments	30.8%	6.5%	62.6%	92.8%	7.2%
Reassembly in-order data	TTL	90.6%	5.7%	3.7%	95.1%	4.9%
	Bad ACK number	83.1%	7.5%	9.5%	93.5%	6.5%
	Bad checksum	87.2%	1.9%	10.8%	98.4%	1.6%
	No TCP flag	48.3%	3.3%	48.4%	97.1%	2.9%
TCB teardown with RST	TTL	73.2%	3.2%	23.6%	94.7%	5.3%
	Bad checksum	63.1%	7.6%	29.3%	89.5%	10.5%
TCB teardown with RST/ACK	TTL	73.1%	3.2%	23.7%	97.1%	2.9%
	Bad checksum	68.9%	1.9%	29.2%	98.2%	1.8%
TCB teardown with FIN	TTL	11.1%	1.0%	87.9%	99.4%	0.6%
	Bad checksum	8.4%	0.8%	90.7%	99.0%	1.0%

Packets with real data are dropped by middleboxes,

sever side implementation, topology changes etc.

New GFW behaviors, inserted packets

dropped by middleboxes

New Behaviors

New TCB upon SYN/ACK

- Prior Assumption: GFW creates a TCB only upon seeing a SYN packet.
- New behavior: GFW creates a TCB not only upon receiving SYN packets, but also SYN/ACK packets.
- TCB creation won't work

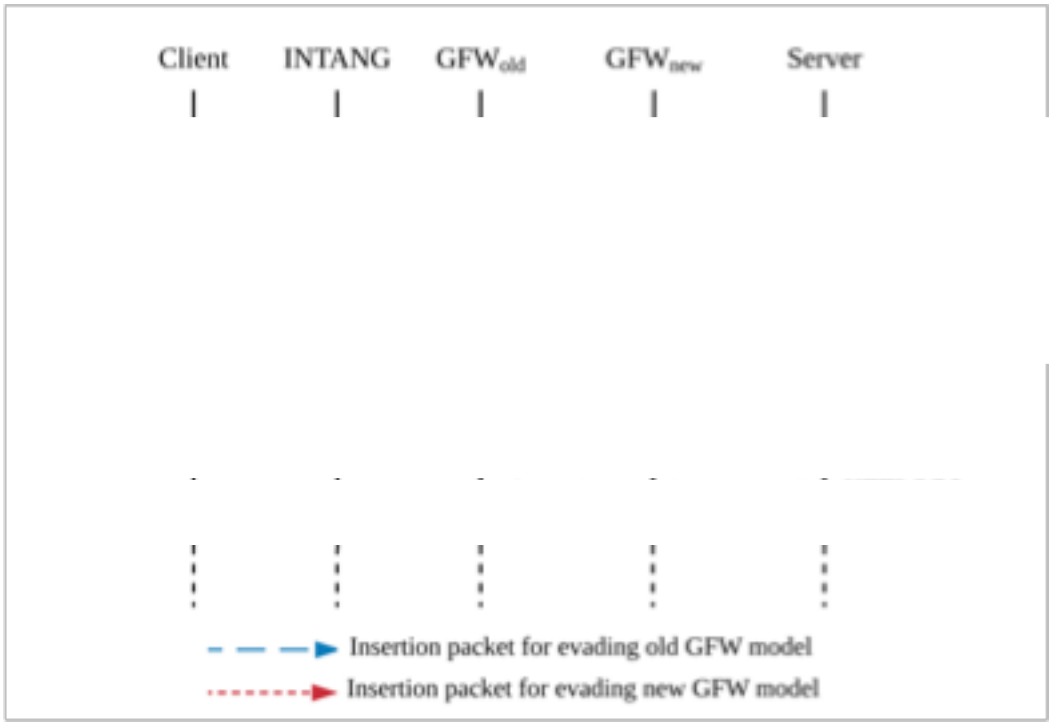
Re-synchronization State

- Prior Assumption: the GFW creates TCB with SEQ in the first SYN
- New Behavior: Enter re-synchronization state upon seeing:
  - Multiple SYN from client side or
  - Multiple SYN/ACK from server side or
  - SYN/ACK with incorrect ACK
  - A RST or RST/ACK packet (instead of tear down TCB)
- The GFW updates client SEQ using next:
  - SEQ in client to server packet or
  - ACK number in SYN/ACK from server to client
- TCB teardown won't work

New Evading Strategies

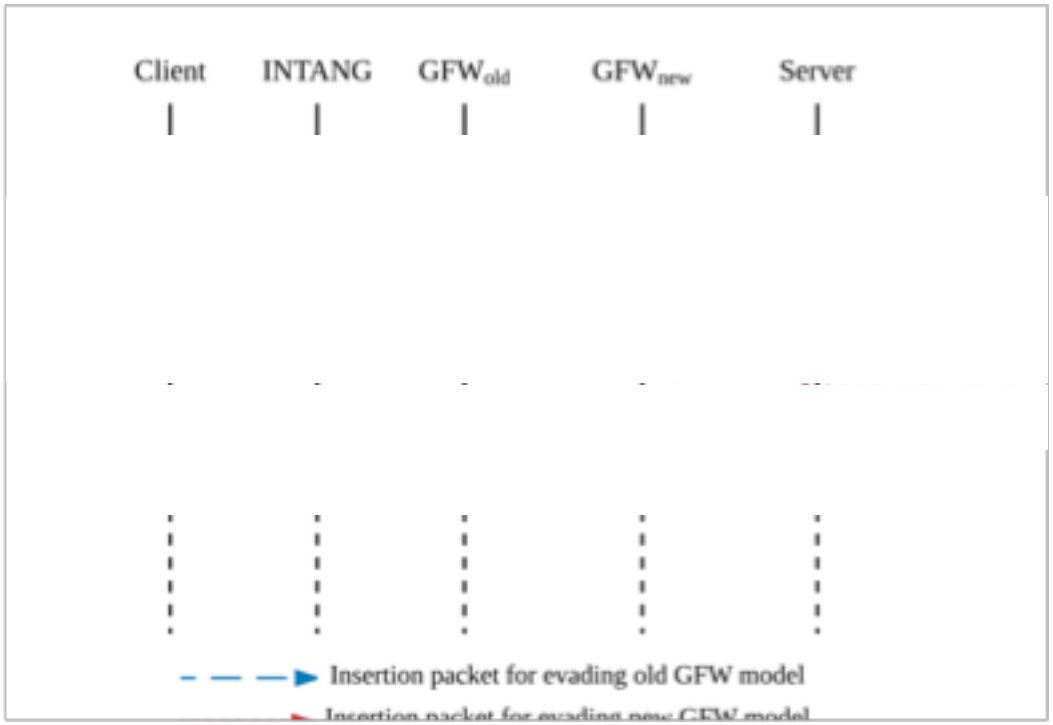
TCB Creation + Resync/Desync

- Resync/Desync
  1. Perform normal handshake
  2. Send a SYN insertion packet (Resync)
  3. Send a packet containing an out-of-window SEQ (Desync)
  4. Then send real request (Ignore by GFW because of its SEQ)
- Combined Strategy
  - First, perform TCB Creation to handle old GFW model
  - Then perform Resync/Desync



TCB Teardown + TCB Reversal

- TCB Reversal:
  - GFW doesn't censor server to client traffic
  - GFW assumes SYN/ACK is sent from server to client and creates TCB accordingly
  - Strategy: Craft a fake SYN/ACK from the client side
- Combined Strategy
  1. Perform TCB Reversal for new GFW model
  2. Then perform TCB teardown for old model



New Insertion Packets

- All evading methods requires injecting additional packets
  - Such packets should only be accepted by the GFW but not the server
- First find insertion packets that would be ignored by the server
  - Ignore path Analysis
    - Program paths that lead to the packet being discarded or “ignored” without any TCP state change. E.g. packet with an incorrect checksum
    - Could be done with static analysis
- Then use them to probe GFW

TCP State	GFW State	TCP Flags	Condition
Any	Any	Any	IP total length > actual length
Any	Any	Any	TCP Header Length < 20
Any	Any	Any	TCP checksum incorrect
SYN_RECV	ESTABLISHED/RESYNC	RST+ACK	Wrong acknowledgement number
SYN_RECV/ESTABLISHED	ESTABLISHED/RESYNC	ACK	Wrong acknowledgement number
SYN_RECV/ESTABLISHED	ESTABLISHED/RESYNC	Any	Has unsolicited MD5 Optional Header
SYN_RECV/ESTABLISHED	ESTABLISHED/RESYNC	No flag	TCP packet with no flag
SYN_RECV/ESTABLISHED	ESTABLISHED/RESYNC	FIN	TCP packet with only FIN flag
SYN_RECV/ESTABLISHED	ESTABLISHED/RESYNC	ACK	Timestamps too old

Not dropped by any middlebox

Table 3: Discrepancies between GFW and server on ignoring packets – candidate insertion packets

Packet Type	TTL	MD5	Bad ACK	Timestamp
SYN	✓			
RST	✓	✓		
Data	✓	✓	✓	✓

Table 5: Preferred construction of insertion packets

INTANG

- Measurement driven censorship evasion tool
- Chooses strategy based on historical measurement results
- Could work with any protocol as long as the IP is not blocked

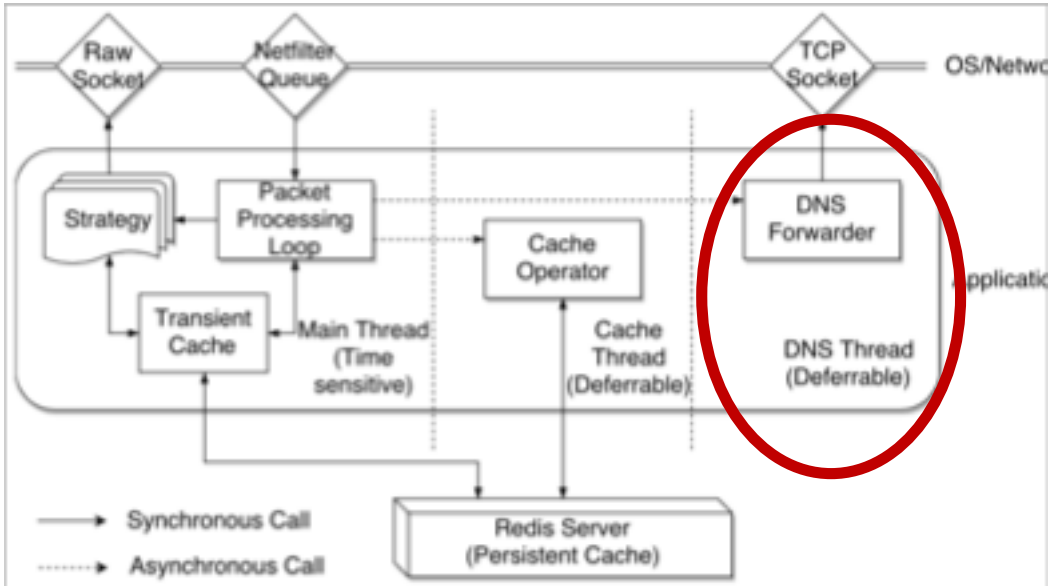


Figure 2: INTANG and its components

Evaluation

Vantage Points	Strategy	Success			Failure 1			Failure 2		
		Min	Max	Avg.	Min	Max	Avg.	Min	Max	Avg.
Inside China	Improved TCB Teardown	89.2%	98.2%	**95.8%**	1.7%	6.7%	**3.1%**	0.0%	5.4%	**1.1%**
	Improved In-order Data Overlapping	86.7%	97.1%	**94.5%**	2.9%	8.9%	**4.4%**	0.0%	5.2%	**1.1%**
	TCB Creation + Resync/Desync	88.5%	98.1%	**95.6%**	1.9%	7.0%	**3.3%**	0.0%	5.5%	**1.1%**
	TCB Teardown + TCB Reversal	90.2%	98.2%	**96.2%**	1.7%	5.6%	**2.6%**	0.0%	5.7%	**1.1%**
	INTANG Performance	93.7%	100.0%	**98.3%**	0.0%	3.0%	**0.9%**	0.0%	3.5%	**0.6%**
Outside China	Improved TCB Teardown	85.6%	92.9%	**89.8%**	4.6%	7.6%	**6.8%**	0.3%	6.8%	**3.5%**
	Improved In-order Data Overlapping	89.4%	96.0%	**92.7%**	1.3%	6.2%	**3.6%**	0.6%	7.0%	**3.7%**
	TCB Creation + Resync/Desync	78.1%	95.6%	**84.6%**	2.4%	18.6%	**12.9%**	0.9%	4.0%	**2.6%**
	TCB Teardown + TCB Reversal	84.6%	93.1%	**89.5%**	5.5%	8.7%	**7.1%**	0.1%	7.9%	**3.3%**

Table 4: Success rate of new strategies

- Better performance than previously existing strategies
- Reasons for failure 1: Misbehaved servers/middleboxes, inaccurate TTL

INTANG with DNS

DNS resolver	IP	except Tianjin	All
Dyn 1	216.146.35.35	98.6%	92.7%
Dyn 2	216.146.36.36	99.6%	93.1%

Table 6: Success rate of TCP DNS censorship evasion

INTANG with Tor

- Background: GFW performs passive traffic analysis and begins active probing after a Tor connection established from China

- Results:
  - W/o INTANG: Hidden bridge nodes triggers active probing and are immediately blocked
  - W/ INTANG: 100% success rate during a 9-hour-experiment-period

Conclusion

- Takeaway
  - GFW and censorship is evolving
  - GFW is heterogeneous with different co-existing versions
  - ITANG could be used to hide VPN/Tor nodes

- Limitation
  - Can’t help with IP level blocking
  - Discovering new strategies and insertion packets requires manual force
  - Can’t hide connection destination

Thank you!