Geneva: Evolving Censorship Evasion

Kevin Bock
In-network censorship by nation-states

Client → Police → Server
In-network censorship by nation-states
In-network censorship by nation-states
In-network censorship by nation-states
In-network censorship by nation-states
In-network censorship by nation-states
In-network censorship by nation-states
In-network censorship by nation-states
In-network censorship by nation-states

Spoofed tear-down packets
In-network censorship by nation-states

Client  Spoofed tear-down packets  Server
In-network censorship by nation-states

Spoofed tear-down packets
In-network censorship by nation-states

Spoofed tear-down packets
In-network censorship by nation-states

Client

The server terminated

Server

The client terminated

Spoofed tear-down packets
In-network censorship by nation-states

Client

Terminated

Server

Terminated

Spoofed tear-down packets

Requires *per-flow state*
In-network censorship by nation-states

Spoofed tear-down packets

Requires *per-flow state*

Censors necessarily *take shortcuts*
In-network censorship by nation-states

Requires *per-flow state*

Censors necessarily *take shortcuts*

Evasion can take advantage of these shortcuts
In-network censorship by nation-states

Client

TTL=2

Requires *per-flow state*

Censors necessarily *take shortcuts*

Evasion can take advantage of these shortcuts
In-network censorship by nation-states

Client -> [TTL=1] -> [Censor] -> Server

Requires *per-flow state*

Censors necessarily *take shortcuts*

Evasion can take advantage of these shortcuts
In-network censorship by nation-states

Client

TTL=1

Requires *per-flow state*

Censors necessarily *take shortcuts*

Evasion can take advantage of these shortcuts
In-network censorship by nation-states

Requires *per-flow state*

Censors necessarily *take shortcuts*

Evasion can take advantage of these shortcuts
In-network censorship by nation-states

Requires *per-flow state*

Censors necessarily *take shortcuts*

Evasion can take advantage of these shortcuts
In-network censorship by nation-states

The client terminated

Requires *per-flow state*

Censors necessarily *take shortcuts*

Evasion can take advantage of these shortcuts
Censorship evasion research

Hypothesize → Measure → Evade

Hypothesize
Measure
Evade
Censorship evasion research

1. Understand how censors operate

- Hypothesize
- Measure
- Evade
Censorship evasion research

1. Understand how censors operate

Hypothesize \rightarrow Measure \rightarrow Evade

2. Apply insight to create evasion strategies
Censorship evasion research

1. Understand how censors operate

Hypothesize – Measure – Evade

2. Apply insight to create evasion strategies

Largely manual efforts give censors the advantage
Censorship evasion research

1. Understand how censors operate

2. Apply insight to create evasion strategies

Largely manual efforts give censors the advantage

Our work gives evasion the advantage
AI-assisted censorship evasion research

Evade → Hypothesize → Measure

Evade
Hypothesize
Measure
AI-assisted censorship evasion research

1. Use AI to **automatically learn** new evasion strategies

- Evade
- Hypothesize
- Measure
AI-assisted censorship evasion research

1. Use AI to automatically learn new evasion strategies

2. Use the strategies the AI finds to understand how the censor works
Geneva Genetic Evasion

1. Use AI to **automatically learn** new evasion strategies.

2. Use the strategies the AI finds to **understand** how the censor works.
Geneva
Genetic Evasion

Building Blocks

Composition

Mutation

Fitness
Geneva runs strictly at one side and manipulates packets to and from the client.
Geneva
Genetic Evasion

Building Blocks

Manipulates packets to and from the client

- Duplicate
- Tamper
- Fragment
- Drop
Geneva
Genetic Evasion

Building Blocks

Manipulates packets to and from the client

- Duplicate
- Tamper
- Fragment
- Drop

Alter or corrupt any TCP/IP header field

No semantic understanding of what the fields mean
Geneva
Genetic Evasion

Building Blocks

Manipulates packets to and from the client

Duplicate
Tamper
Fragment
Drop

Fragment (IP) or Segment (TCP)

Alter or corrupt any TCP/IP header field

No semantic understanding of what the fields mean
Geneva
Genetic Evasion

Building Blocks
Actions manipulate individual packets
- Duplicate
- Tamper
- Fragment
- Drop

Composition

Mutation

Fitness
Geneva
Genetic Evasion

**Building Blocks**
- Actions manipulate individual packets
  - Duplicate
  - Tamper
  - Fragment
  - Drop

**Composition**

**Mutation**

**Fitness**

Actions manipulate individual packets through building blocks such as duplicate, tamper, fragment, and drop. The composition shows the structure of DNA, and mutation involves changes in DNA sequences. Fitness is represented by the evolutionary process from ape to human.
Geneva
Genetic Evasion

Composition

out:tcp.flags=A

Duplicate

Tamper
tcp.flags = R

Tamper
ip.ttl = 2
Running a Strategy

Composition

Client

Duplicate

Tamper
tcp.flags = R

Tamper
ip.ttl = 2

out:tcp.flags=A

Server
Running a Strategy

Composition

Client

Duplicate

Tamper
tcp.flags = R

Tamper
ip.ttl = 2

out:tcp.flags=A

Server
Running a Strategy

Composition

Client

Duplicate

Tamper
tcp.flags = R
Tamper
ip.ttl = 2

out:tcp.flags=A

Server
Running a Strategy
Running a Strategy

Composition

Client -> Duplicate -> Tamper (tcp.flags = A) -> Server

Tamper (tcp.flags = R)
Tamper (ip.ttl = 2)
Running a Strategy

Composition

Client

Duplicate

Tamper
tcp.flags = R

Tamper
ip_ttl = 2

TTL=8

TTL=2

Server
Running a Strategy

Composition

Client

Duplicate

Tamper
tcp.flags = R

Tamper
ip.ttl = 2

TTL=2

Server
Running a Strategy

Composition

Client to Server Composition

Client

Duplicate

Tamper
	tcp.flags = R

Tamper
	ip.ttl = 2

TTL=2

Server
Running a Strategy

Composition

Client

Duplicate

Tamper
tcp.flags = R

Tamper
ip.ttl = 2

Server
Running a Strategy

Composition

Client

Duplicate

Tamper
  tcp.flags = R

Tamper
  ip.ttl = 2

out:tcp.flags=A

Server
Geneva
Genetic Evasion

Building Blocks
Actions manipulate individual packets
- Duplicate
- Tamper
- Fragment
- Drop

Composition
Actions compose to form trees
- out:tcp.flags = A
- Duplicate
- Tamper tcp.flags = R
- Tamper ip.ttl = 2

Mutation

Fitness

Geneva
Genetic Evasion

Building Blocks
Actions manipulate individual packets
- Duplicate
- Tamper
- Fragment
- Drop

Composition
Actions compose to form trees
out:tcp.flags=A
- Duplicate
- Tamper tcp.flags = R
- Tamper ip.ttl = 2

Mutation

Fitness

Geneva

Genetic Evasion

Building Blocks
Actions manipulate individual packets
- Duplicate
- Tamper
- Fragment
- Drop

Composition
Actions compose to form trees
- Duplicate
- Tamper tcp.flags = R
- Tamper ip.ttl = 2

Mutation
Randomly alter types, values, and trees

Fitness
Geneva
Genetic Evasion

**Building Blocks**
Actions manipulate individual packets

- Duplicate
- Tamper
- Fragment
- Drop

**Composition**
Actions compose to form trees

- out:tcp.flags=A
- Duplicate
- Tamper tcp.flags = R
- Tamper ip.ttl = 2

**Mutation**
Randomly alter types, values, and trees

**Fitness**
Geneva Genetic Evasion

Fitness

Which *individuals* should survive to the next *generation*?
Which **individuals** should survive to the next **generation**?
Which *individuals* should survive to the next *generation*?
Which *individuals* should survive to the next *generation*?

- Not triggering on any packets
- Breaking the TCP connection
- Successfully obtaining forbidden content
- Conciseness
Geneva
Genetic Evasion

**Building Blocks**
Actions manipulate individual packets
- Duplicate
- Tamper
- Fragment
- Drop

**Composition**
Actions compose to form trees
- out:tcp.flags=A
- Duplicate
- Tamper tcp.flags = R
- Tamper ip.ttl = 2

**Mutation**
Randomly alter types, values, and trees

**Fitness**
Goal: Fewest actions needed to succeed
- No trigger
- Break TCP
- Successful
- Concise
Geneva’s results – Real censor experiments

- **Injects TCP RSTs**
  - China

- **Injects & blackholes**
  - Iran

- **Injects & blackholes**
  - Kazakhstan

- **Injects a block page**
  - India

<table>
<thead>
<tr>
<th>Protocol</th>
<th>HTTP</th>
<th>HTTPS</th>
<th>DNS</th>
<th>FTP</th>
<th>SMTP</th>
</tr>
</thead>
<tbody>
<tr>
<td>China</td>
<td>🇨🇳</td>
<td>🇨🇳</td>
<td>🇨🇳</td>
<td>🇨🇳</td>
<td>🇨🇳</td>
</tr>
<tr>
<td>Iran</td>
<td>🇮🇷</td>
<td>🇮🇷</td>
<td>🇮🇷</td>
<td>🇮🇷</td>
<td>🇮🇷</td>
</tr>
<tr>
<td>Kazakhstan</td>
<td>🇰🇿</td>
<td>🇰🇿</td>
<td>🇰🇿</td>
<td>🇰🇿</td>
<td>🇰🇿</td>
</tr>
<tr>
<td>India</td>
<td>🇮🇳</td>
<td>🇮🇳</td>
<td>🇮🇳</td>
<td>🇮🇳</td>
<td>🇮🇳</td>
</tr>
</tbody>
</table>

*Note: The DNS column in India has an asterisk to indicate a specific censor response.*
## Geneva’s results – Real censor experiments

### Diversity of censors

<table>
<thead>
<tr>
<th>Country</th>
<th>Injection Method</th>
<th>HTTP</th>
<th>HTTPS</th>
<th>DNS</th>
<th>FTP</th>
<th>SMTP</th>
</tr>
</thead>
<tbody>
<tr>
<td>China</td>
<td>Injects TCP RSTs</td>
<td><img src="image" alt="China Flag" /></td>
<td><img src="image" alt="China Flag" /></td>
<td><img src="image" alt="China Flag" /></td>
<td><img src="image" alt="China Flag" /></td>
<td><img src="image" alt="China Flag" /></td>
</tr>
<tr>
<td>Iran</td>
<td>Injects &amp; blackholes</td>
<td><img src="image" alt="Iran Flag" /></td>
<td><img src="image" alt="Iran Flag" /></td>
<td><img src="image" alt="Iran Flag" /></td>
<td><img src="image" alt="Iran Flag" /></td>
<td><img src="image" alt="Iran Flag" /></td>
</tr>
<tr>
<td>Kazakhstan</td>
<td>Injects &amp; blackholes</td>
<td><img src="image" alt="Kazakhstan Flag" /></td>
<td><img src="image" alt="Kazakhstan Flag" /></td>
<td><img src="image" alt="Kazakhstan Flag" /></td>
<td><img src="image" alt="Kazakhstan Flag" /></td>
<td><img src="image" alt="Kazakhstan Flag" /></td>
</tr>
<tr>
<td>India</td>
<td>Injects a block page</td>
<td><img src="image" alt="India Flag" /></td>
<td><img src="image" alt="India Flag" /></td>
<td><img src="image" alt="India Flag" /></td>
<td><img src="image" alt="India Flag" /></td>
<td><img src="image" alt="India Flag" /></td>
</tr>
</tbody>
</table>
### Diversity of censors

<table>
<thead>
<tr>
<th>Country</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>China</td>
<td>Injects TCP RSTs</td>
</tr>
<tr>
<td>Iran</td>
<td>Injects &amp; blackholes</td>
</tr>
<tr>
<td>Kazakhstan</td>
<td>Injects &amp; blackholes</td>
</tr>
<tr>
<td>India</td>
<td>Injects a block page</td>
</tr>
</tbody>
</table>

### Diversity of protocols

<table>
<thead>
<tr>
<th>Protocols</th>
<th>China</th>
<th>Iran</th>
<th>Kazakhstan</th>
<th>India</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP</td>
<td><img src="image.png" alt="Police" /></td>
<td><img src="image.png" alt="Police" /></td>
<td><img src="image.png" alt="Police" /></td>
<td><img src="image.png" alt="Police" /></td>
</tr>
<tr>
<td>HTTPS</td>
<td><img src="image.png" alt="Police" /></td>
<td><img src="image.png" alt="Police" /></td>
<td><img src="image.png" alt="Police" /></td>
<td><img src="image.png" alt="Police" /></td>
</tr>
<tr>
<td>DNS</td>
<td><img src="image.png" alt="Police" /></td>
<td><img src="image.png" alt="Police" /></td>
<td><img src="image.png" alt="Police" /></td>
<td><img src="image.png" alt="Police" /></td>
</tr>
<tr>
<td>FTP</td>
<td><img src="image.png" alt="Police" /></td>
<td><img src="image.png" alt="Police" /></td>
<td><img src="image.png" alt="Police" /></td>
<td><img src="image.png" alt="Police" /></td>
</tr>
<tr>
<td>SMTP</td>
<td><img src="image.png" alt="Police" /></td>
<td><img src="image.png" alt="Police" /></td>
<td><img src="image.png" alt="Police" /></td>
<td><img src="image.png" alt="Police" /></td>
</tr>
</tbody>
</table>

*DNS in Kazakhstan is not censored.*
Geneva’s results – Real censor experiments

- China
- India
- Iran
- Kazakhstan
Geneva’s results – Real censor experiments

- 6 Species
- 13 Sub-species
- 36 Variants

Flags:
- China
- India
- Iran
- Kazakhstan
Geneva’s results – Real censor experiments

- **6 Species**
- **13 Sub-species**
- **36 Variants**

The underlying bug
How Geneva exploits it
Functionally distinct

---

- China
- India
- Iran
- Kazakhstan
Geneva’s results – Real censor experiments

6 Species
13 Sub-species
36 Variants

The underlying bug
How Geneva exploits it
Functionally distinct

China
India
Iran
Kazakhstan
Geneva’s results – Real censor experiments

6 Species

13 Sub-species

36 Variants

The underlying bug

How Geneva exploits it

Functionally distinct

31
6
9
13

China
India
Iran
Kazakhstan
Trick the censor into thinking the client is the server.
Segmentation species

out:tcp.flags=PA

Fragment
tcp:8:inorder

Fragment
tcp:4:inorder

Turnaround species

out:tcp.flags=S

Duplicate

Tamper
tcp.flags = SA

Trick the censor into thinking the client is the server

Segment the request
Trick the censor into thinking the client is the server

Segment the request

Turnaround species

Segmentation species

out:tcp.flags=S

Duplicate

Tamper
tcp.flags = SA

GET /?search=ultrasurf

out:tcp.flags=PA

Fragment
tcp:8:inorder

Fragment
tcp:4:inorder
Trick the censor into thinking the client is the server

Segment the request

Turnaround species

out:tcp.flags=S

Duplicate

Tamper
tcp.flags = SA

Segmentation species

out:tcp.flags=PA

Fragment
tcp:8:inorder

Fragment
tcp:4:inorder

GET /?search=ultrasurf
Trick the censor into thinking the client is the server

Segment the request, but not the keyword

Turnaround species

out:tcp.flags=S

Duplicate

Tamper
tcp.flags = SA

Segmentation species

out:tcp.flags=PA

Fragment
tcp:8:inorder

Fragment
tcp:4:inorder

GET /?se
arch
=ultrasurf

8
4
Remainder
Segment the request, but *not the keyword*

**Turnaround species**

- out:tcp.flags=S
- Duplicate
- Tamper tcp.flags = SA

**Segmentation species**

- out:tcp.flags=PA
- Fragment tcp:8:inorder
- Fragment tcp:4:inorder
- GET /?se=ultrasurf
- arch
- =ultrasurf

Trick the censor into thinking the client is the server
Trick the censor into thinking the client is the server

Segment the request, but *not* the keyword
Censoring regime

Client
Geneva

Server
Server-side evasion

Censoring regime

Client

Server

Geneva
Server-side evasion

Censoring regime

Clients

Potentially broadens reachability without *any* client-side deployment
Server-side evasion “shouldn’t” work
Server-side evasion “shouldn’t” work

Censored keyword

Client

SYN
SYN/ACK
ACK
PSH/ACK
response

Server

SYN
SYN/ACK
ACK

PSH/ACK
query

(response)

"shouldn’t" work
Server-side evasion “shouldn’t” work

All a server does before client is censored

Censored keyword
Server-side evasion “shouldn’t” work

All a server does before client is censored

Censored keyword

Fortunately, the AI doesn’t know it “shouldn’t” work
Server-side evasion “shouldn’t” work
Server-side results
Server-side evasion “shouldn’t” work
Server-side results

China
8 strategies
Server-side evasion “shouldn’t” work
Server-side results

China
8 strategies

Iran/India
1 strategy
Server-side evasion “shouldn’t” work

Server-side results

China
8 strategies

Iran/India
1 strategy

Kazakhstan
3 strategies
Server-side evasion “shouldn’t” work

Server-side results

China
8 strategies

Iran/India
1 strategy

Kazakhstan
3 strategies

None of these require any client-side deployment
Server-side evasion “shouldn’t” work

Client

SYN
SYN/ACK
ACK

Server

PSH/ACK (query)

PSH/ACK (response)

All a server does before client is censored

Client-side evasion

Censored keyword

"shouldn't" work
Server-side evasion results

Double benign-GETs

Client  Server

SYN
SYN/ACK (benign GET)
SYN/ACK (benign GET)
ACK
ACK
ACK
PSH/ACK (query)
ACK
PSH/ACK (response)
Server-side evasion results

Double benign-GETs

Server sends uncensored GETs inside two SYN/ACKs
Server-side evasion results

**Double benign-GETs**

Censor confuses connection direction

Server sends uncensored GETs inside two SYN/ACKs
Server-side evasion results

Simultaneous-open-based desynchronization
Server-side evasion results

Simultaneous-open-based desynchronization

TCP simultaneous open
Server-side evasion results

Simultaneous-open-based desynchronization

TCP simultaneous open

Client sends a SYN/ACK

Client

SYN
SYN
SYN (corrupted)
SYN/ACK
ACK
ACK
ACK
PSH/ACK (query)
ACK
PSH/ACK (response)

Server
Server-side evasion results

Simultaneous-open-based desynchronization

TCP simultaneous open

Client sends a SYN/ACK

Censor de-synchronizes
New Model for Chinese Censorship

All of the server-side strategies operate strictly during the TCP 3-way handshake

<table>
<thead>
<tr>
<th>Strategy #</th>
<th>Description</th>
<th>DNS</th>
<th>FTP</th>
<th>HTTP</th>
<th>HTTPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>China</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>No evasion</td>
<td>3%</td>
<td>3%</td>
<td>3%</td>
<td>3%</td>
</tr>
<tr>
<td>1</td>
<td>Simultaneous Open, Injected RST</td>
<td>89%</td>
<td>52%</td>
<td>54%</td>
<td>14%</td>
</tr>
<tr>
<td>2</td>
<td>Simultaneous Open, Injected Load</td>
<td>83%</td>
<td>38%</td>
<td>54%</td>
<td>55%</td>
</tr>
<tr>
<td>3</td>
<td>Corrupt ACK, Simultaneous Open</td>
<td>26%</td>
<td>65%</td>
<td>4%</td>
<td>4%</td>
</tr>
<tr>
<td>4</td>
<td>TCP window reduction</td>
<td>3%</td>
<td>47%</td>
<td>2%</td>
<td>3%</td>
</tr>
<tr>
<td>5</td>
<td>Corrupt ACK Alone</td>
<td>7%</td>
<td>33%</td>
<td>5%</td>
<td>5%</td>
</tr>
<tr>
<td>6</td>
<td>Corrupt ACK, Injected Load</td>
<td>15%</td>
<td>97%</td>
<td>4%</td>
<td>3%</td>
</tr>
<tr>
<td>7</td>
<td>Injected Load, Induced RST</td>
<td>82%</td>
<td>55%</td>
<td>52%</td>
<td>54%</td>
</tr>
<tr>
<td>8</td>
<td>Injected RST, Induced RST</td>
<td>83%</td>
<td>85%</td>
<td>54%</td>
<td>4%</td>
</tr>
</tbody>
</table>

India

<table>
<thead>
<tr>
<th>Strategy #</th>
<th>Description</th>
<th>DNS</th>
<th>FTP</th>
<th>HTTP</th>
<th>HTTPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>TCP window reduction</td>
<td>100%</td>
<td>100%</td>
<td>2%</td>
<td>100%</td>
</tr>
</tbody>
</table>

Kazakhstan

<table>
<thead>
<tr>
<th>Strategy #</th>
<th>Description</th>
<th>DNS</th>
<th>FTP</th>
<th>HTTP</th>
<th>HTTPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>TCP window reduction</td>
<td>100%</td>
<td>100%</td>
<td>0%</td>
<td>100%</td>
</tr>
</tbody>
</table>

---

---
New Model for Chinese Censorship

All of the server-side strategies operate **strictly** during the TCP 3-way handshake.

<table>
<thead>
<tr>
<th>Strategy #</th>
<th>Description</th>
<th>DNS</th>
<th>FTP</th>
<th>HTTP</th>
<th>HTTPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>China</td>
<td>No evasion</td>
<td>3%</td>
<td>3%</td>
<td>3%</td>
<td>3%</td>
</tr>
<tr>
<td></td>
<td>1 Simultaneous Open, Injected RST</td>
<td>89%</td>
<td>52%</td>
<td>54%</td>
<td>14%</td>
</tr>
<tr>
<td></td>
<td>2 Simultaneous Open, Injected Load</td>
<td>83%</td>
<td>36%</td>
<td>54%</td>
<td>55%</td>
</tr>
<tr>
<td></td>
<td>3 Corrupt ACK, Simultaneous Open</td>
<td>26%</td>
<td>65%</td>
<td>4%</td>
<td>4%</td>
</tr>
<tr>
<td></td>
<td>4 TCP window reduction</td>
<td>3%</td>
<td>47%</td>
<td>2%</td>
<td>3%</td>
</tr>
<tr>
<td></td>
<td>5 Corrupt ACK Alone</td>
<td>7%</td>
<td>33%</td>
<td>5%</td>
<td>5%</td>
</tr>
<tr>
<td></td>
<td>6 Corrupt ACK, Injected Load</td>
<td>15%</td>
<td>97%</td>
<td>4%</td>
<td>3%</td>
</tr>
<tr>
<td></td>
<td>7 Injected Load, Induced RST</td>
<td>82%</td>
<td>55%</td>
<td>52%</td>
<td>54%</td>
</tr>
<tr>
<td></td>
<td>8 Injected RST, Induced RST</td>
<td>83%</td>
<td>85%</td>
<td>54%</td>
<td>4%</td>
</tr>
<tr>
<td>India</td>
<td>No evasion</td>
<td>100%</td>
<td>100%</td>
<td>2%</td>
<td>100%</td>
</tr>
<tr>
<td></td>
<td>4 TCP window reduction</td>
<td>–</td>
<td>–</td>
<td>100%</td>
<td>–</td>
</tr>
<tr>
<td>Kazakhstan</td>
<td>No evasion</td>
<td>100%</td>
<td>100%</td>
<td>0%</td>
<td>100%</td>
</tr>
<tr>
<td></td>
<td>4 TCP window reduction</td>
<td>–</td>
<td>–</td>
<td>100%</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>9 Triple Load</td>
<td>–</td>
<td>–</td>
<td>100%</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>10 Double GET</td>
<td>–</td>
<td>–</td>
<td>100%</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>11 Null Flags</td>
<td>–</td>
<td>–</td>
<td>100%</td>
<td>–</td>
</tr>
</tbody>
</table>

So why are different applications affected differently in China?
New Model for Chinese Censorship

Sane

DNS  HTTP  FTP
TCP  IP
New Model for Chinese Censorship

They appear to be running multiple censoring middleboxes in parallel.

Apparently what’s happening:

Sane:
- DNS
- HTTP
- FTP
- TCP
- IP

Apparently what’s happening:
- DNS
- HTTP
- FTP
- TCP
- IP
New Model for Chinese Censorship

Sane

<table>
<thead>
<tr>
<th>DNS</th>
<th>HTTP</th>
<th>FTP</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IP</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Apparently what’s happening

<table>
<thead>
<tr>
<th>DNS</th>
<th>HTTP</th>
<th>FTP</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>TCP</td>
<td>TCP</td>
</tr>
<tr>
<td>IP</td>
<td>IP</td>
<td>IP</td>
</tr>
</tbody>
</table>

They appear to be running multiple censoring middleboxes in parallel.
New Model for Chinese Censorship

How does the censor know which one to apply to a connection?

Not port number

They appear to apply protocol fingerprinting

Basic protocol confusion could be highly effective

Apparently what’s happening

They appear to be running multiple censoring middleboxes in parallel
Geneva defeats censorship-in-depth

February 2020: Iran launched a new system: a protocol filter
Geneva defeats censorship-in-depth

February 2020: Iran launched a new system: a protocol filter

Censors connections that do not match protocol fingerprints
Geneva defeats censorship-in-depth

February 2020: Iran launched a new system: a protocol filter

Censors connections that do not match protocol fingerprints

Those that do match are then subjected to standard censorship
Geneva defeats censorship-in-depth

February 2020: Iran launched a new system: a protocol filter

Censors connections that do not match protocol fingerprints

Those that do match are then subjected to standard censorship

Geneva discovered 3 strategies to evade Iran’s filter
Automating the arms race

AI has the potential to fast-forward the arms race for both sides
Automating the arms race

AI has the potential to fast-forward the arms race for both sides.

- Bugs in implementation: Easy for censors to fix the low-hanging fruit.
- Gaps in logic: Harder for censors to fix systemic issues.
Automating the arms race

AI has the potential to **fast-forward** the arms race *for both sides*

- **Bugs in implementation**
  - Easy for censors to fix the low-hanging fruit

- **Gaps in logic**
  - Harder for censors to fix systemic issues

**What is the *logical conclusion* of the arms race?**
Geneva Team

Kevin Bock
Louis-Henri Merino
Tania Arya
Daniel Liscinsky
Regina Pogosian
Yair Fax

George Hughey
Kyle Reese
Jasraj Singh
Kyle Hurley
Michael Harrity

Dave Levin
Xiao Qiang
Evolving censorship evasion

Geneva Genetic Evasion

Client-side & Server-side

Has found dozens of strategies

Quickly discovers new strategies

Gives the advantage to evaders