liberate, (n):
A library for exposing (traffic-classification) rules and avoiding them efficiently

Fangfan Li, Abbas Razaghpanah, Arash Molavi Kakhki, Arian Akhavan Niaki, David Choffnes, Phillipa Gill, Alan Mislove
Traffic management
Traffic management

Internet Service Provider

Throttling

YouTube
Traffic management

Internet Service Provider

Throttling

Blocking

YouTube
Traffic management

Internet Service Provider

Throttling

Blocking
Traffic management

Internet Service Provider

Throttling

Blocking

Zero rating
Traffic management

Internet Service Provider

Throttling

Blocking

Zero rating
Example policy

Now you can stream all you want for FREE without using your data.

With Binge On, Simple Choice users on a qualifying plan are FREE to stream unlimited video on your favorite services like YouTube, Netflix, HBO NOW, and many more without using a drop of your high-speed data. Nothing to configure – all automatically applied to your qualifying plan. Streamers, go ahead and Binge On.

Request a video streaming service to Binge On ➪

Detectable video typically streams at DVD quality (480p+) with Binge On unless video provider opts-out; on opt-out, high-speed data consumption will continue as if Binge On was disabled. Click below for opted-out providers (subject to change). On all T-Mobile plans, during congestion, the small fraction of customers using >50GB/mo. may notice reduced speeds until next bill cycle due to data prioritization. For best performance, leave any video streaming applications at their default automatic resolution setting. You may disable Binge On at any time, but will lose Binge On benefits. Sling not available in Puerto Rico. The trademarks shown are owned and registered by their respective owners.

See provider opt-out list ➪
Example policy

Now you can stream all you want for FREE without using your data.

With Binge On, Simple Choice users on a qualifying plan are FREE to stream unlimited video on your favorite services like YouTube, Netflix, HBO NOW, and many more without using a drop of your high-speed data. Nothing to configure – all automatically applied to your qualifying plan. Streamers, go ahead and Binge On.

Request a video streaming service to Binge On ➤

Detectable video typically streams at DVD quality (480p+) with Binge On using >50GB/mo. may notice reduced speeds until next bill cycle due to data prioritization. For best performance, leave any video streaming applications at their default automatic resolution setting. You may disable Binge On at any time, but will lose Binge On benefits. Sling not available in Puerto Rico. The trademarks shown are owned and registered by their respective owners.

See provider opt-out list ➤
Example policy

Now you can stream all you want for FREE without using your data.

With Binge On, Simple Choice users on a qualifying plan are FREE to stream unlimited video on your favorite services like YouTube, Netflix, HBO NOW, and many more without using a drop of your high-speed data. Nothing to configure – all automatically applied to your qualifying plan. Streamers, go ahead and Binge On.

Request a video streaming service to Binge On ➤

Detectable video typically streams at DVD quality (480p+) with Binge On using >50GB/mo. may notice reduced speeds until next bill cycle due to data prioritization. For best performance, leave any video streaming applications at their default automatic resolution setting. You may disable Binge On at any time, but will lose Binge On benefits. Sling not available in Puerto Rico. The trademarks shown are owned and registered by their respective owners.

See provider opt-out list ➤
Example policy

Now you can stream all you want for FREE without using your data.

With Binge On, Simple Choice users on a qualifying plan are FREE to stream unlimited video on your favorite services like YouTube, Netflix, HBO NOW, and many more without using a drop of your high-speed data. Nothing to configure – all automatically applied to your qualifying plan. Streamers, go ahead and Binge On.

Request a video streaming service to Binge On ➤

Detectable video typically streams at DVD quality (480p+) with Binge On using >50GB/mo. may notice reduced speeds until next bill cycle due to data prioritization. For best performance, leave any video streaming applications at their default automatic resolution setting. You may disable Binge On at any time, but will lose Binge On benefits. Sling not available in Puerto Rico. The trademarks shown are owned and registered by their respective owners.

See provider opt-out list ➤
Lack of user control
Lack of user control

- Policies are implemented by DPI (Deep Packet Inspection) devices [IMC 16]
Lack of user control

- Policies are implemented by DPI (Deep Packet Inspection) devices [IMC 16]
Lack of user control

- Policies are implemented by DPI (Deep Packet Inspection) devices [IMC 16]
- Differentiation policy can be harmful or unwanted to users/content providers
Lack of user control

- Policies are implemented by DPI (Deep Packet Inspection) devices [IMC 16]
- Differentiation policy can be harmful or unwanted to users/content providers
- Users/content providers have no control over these policies
Previous work
Previous work

- Approaches:
  - VPNs and proxies
  - Covert channels
  - Obfuscating traffic
  - Domain fronting
Previous work

• Approaches:
  • VPNs and proxies
  • Covert channels
  • Obfuscating traffic
  • Domain fronting

• Limitations:
Previous work

- **Approaches:**
  - VPNs and proxies
  - Covert channels
  - Obfuscating traffic
  - Domain fronting

- **Limitations:**
  - Brittle
Previous work

• **Approaches:**
  • VPNs and proxies
  • Covert channels
  • Obfuscating traffic
  • Domain fronting

• **Limitations:**
  • Brittle
  • Development effort
Previous work

- **Approaches:**
  - VPNs and proxies
  - Covert channels
  - Obfuscating traffic
  - Domain fronting

- **Limitations:**
  - Brittle
  - Development effort
  - Performance
Previous work

- Approaches:
  - VPNs and proxies
  - Covert channels
  - Obfuscating traffic
  - Domain fronting

- Limitations:
  - Brittle
  - Development effort
  - Performance
  - Manual inspection
Goals of liberate
Goals of liberate

- A technical solution for detecting and evading unwanted policies
Goals of liberate

- A technical solution for detecting and evading unwanted policies
- Enables unmodified applications to evade
Goals of liberate

- A technical solution for **detecting** and **evading** unwanted policies
- Enables unmodified applications to evade
  - Automatically
Goals of liberate

• A technical solution for detecting and evading unwanted policies

• Enables unmodified applications to evade
  • Automatically
  • Adaptively
Goals of liberate

- A technical solution for **detecting** and **evading** unwanted policies
- Enables unmodified applications to evade
  - Automatically
  - Adaptively
  - Unilaterally
Goals of liberate

- A technical solution for **detecting** and **evading** unwanted policies
- Enables unmodified applications to evade
  - Automatically
  - Adaptively
  - Unilaterally
  - With low overhead
Goals of liberate

- A technical solution for **detecting** and **evading** unwanted policies
- Enables unmodified applications to evade
  - Automatically
  - Adaptively
  - Unilaterally
  - With low overhead
Outline

• Design and implementation
  • Traffic-classification rules detection
  • Evasion techniques
  • Implementation

• Evaluation
  • Effectiveness across multiple networks
Overview of liberate
Overview of liberate

Differentiation Detection

Record traffic

Replay traffic

Replay inverted traffic

Differentiation?

YES

Content based differentiation

NO

Stop

Differentiation Evaluation

Characterization

Recorded traffic

Analysis/pending

Replay traffic

Complete?

YES

Differentiation rules

NO

Evasion Evaluation

Recorded traffic

Insert packet insertion/Splitting payload/Reordering payload

Effective techniques

Evasion Deployment

Application traffic

Selected evasion technique
Overview of liberate
Overview of liberate
Overview of liberate
Overview of liberate

Differentiation Detection
- Record traffic
- Replay traffic
- Replay inverted traffic
  - Differentiation?
    - YES
    - Content based differentiation
    - NO

Characterization
- Recorded traffic
- Binary Analysis/ Prepending
- Replay traffic
- Complete?
  - YES
  - Classification rules
  - NO

Evasion Evaluation
- Recorded traffic
- Insert packet insertion/ Splitting payload/ Reordering payload
- Effective techniques

Evasion Deployment
- Application traffic
- Selected evasion technique
Outline

• Design and implementation
  • Traffic-classification rules detection
  • Evasion techniques
  • Implementation
• Evaluation
  • Effectiveness across multiple networks
Design
Traffic-classification rules detection
Design
Traffic-classification rules detection

- How to detect differentiation?
  - Record and Replay [IMC 15]
Design

Traffic-classification rules detection

• How to detect differentiation?
  • Record and Replay [IMC 15]
How to detect differentiation?

- Record and Replay [IMC 15]
Design

Traffic-classification rules detection

• How to detect differentiation?
  • Record and Replay [IMC 15]

• How to evade differentiation efficiently?
Design
Traffic-classification rules detection

- How to detect differentiation?
  - Record and Replay [IMC 15]
- How to evade differentiation efficiently?
  - Understand classification rules [IMC 16]
Design

Traffic-classification rules detection

• How to detect differentiation?
  • Record and Replay [IMC 15]

• How to evade differentiation efficiently?
  • Understand classification rules [IMC 16]
Design
Traffic-classification rules detection

- How to detect differentiation?
- Record and Replay [IMC 15]
- How to evade differentiation efficiently?
- Understand classification rules [IMC 16]

<table>
<thead>
<tr>
<th>Header</th>
<th>Example matching content</th>
</tr>
</thead>
<tbody>
<tr>
<td>URI</td>
<td>site.js{…}-nbcspor-sports-com</td>
</tr>
<tr>
<td>Host</td>
<td>Host: <a href="http://www.spotifi.com">www.spotifi.com</a></td>
</tr>
<tr>
<td>User-Agent</td>
<td>User-Agent: Pandora 5.0{…}</td>
</tr>
<tr>
<td>Content-Type</td>
<td>Content-Type: video</td>
</tr>
<tr>
<td>SNI</td>
<td>googlevideo.com</td>
</tr>
</tbody>
</table>
Outline

- Design and implementation
  - Traffic-classification rules detection
  - Evasion techniques
  - Implementation
- Evaluation
  - Effectiveness across multiple networks
Design

Example classification

How does classifier classify application B?
How does classifier classify application B?
Design

Example classification

How does classifier classify application B?
Design
Example classification

How does classifier classify application B?
Design

Example classification

How does classifier classify application B?
Design
Example classification

How does classifier classify application B?
How does classifier classify application B?

Matching contents: ‘GET /B’
Design

Evasion techniques

Design
Evasion techniques

• Observation:
  • ‘Match and forget’ behavior

Design
Evasion techniques

• Observation:
  • ‘Match and forget’ behavior
  • Incomplete views of the connection

Using a small TTL value
Design
Evasion techniques

• Observation:
  • ‘Match and forget’ behavior
  • Incomplete views of the connection

• **Inert packet insertion***: Traffic processed only by a classifier but not endpoint

---

Design

Evasion techniques

• Observation:
  
  • ‘Match and forget’ behavior
  
  • Incomplete views of the connection

• **Inert packet insertion**
  * Traffic processed only by a classifier but not endpoint

Design
Evasion techniques

• Observation:
  • ‘Match and forget’ behavior
  • Incomplete views of the connection

• Inert packet insertion*: Traffic processed only by a classifier but not endpoint

---

Design
Evasion techniques

• Observation:
  • ‘Match and forget’ behavior
  • Incomplete views of the connection

• Inert packet insertion*: Traffic processed only by a classifier but not endpoint

Design
Evasion techniques

Client
SYN
SYN, ACK
ACK

Classifier
TCP 80
TCP 80
TCP 80

Server
Design
Evasion techniques

• Observation:
  • Each packet is searched independently for matching contents
Design

Evasion techniques

• Observation:
  • Each packet is searched independently for matching contents

• **Splitting/Reordering**: splitting the matching contents across multiple packets
Design

Evasion techniques

• Observation:

  • Each packet is searched independently for matching contents

• **Splitting/Reordering**: splitting the matching contents across multiple packets

![Diagram of network traffic and IP packet fragmentation]

Fragmenting the IP packet
Design

Evasion techniques

- Observation:
  - Each packet is searched independently for matching contents

- **Splitting/Reordering**: splitting the matching contents across multiple packets

![Diagram showing packet fragmentation and reordering](image-url)
Design
Evasion techniques

Client

SYN
SYN, ACK
ACK

Classifier

TCP 80
TCP 80
TCP 80

Server
Design

Evasion techniques

- Observation:
  - Classifiers do not retain classification results indefinitely
Design

Evasion techniques

- **Observation:**
  - Classifiers do not retain classification results indefinitely

- **Flush:** causing the classifier to remove the classification state for the flow

```
Client
  SYN  
  SYN, ACK
  ACK

Classifier
  TCP 80
  TCP 80

Server
  TCP 80
  TCP 80
```

Inserting large delays
Design

Evasion techniques

- Observation:
  - Classifiers do not retain classification results indefinitely
  - **Flushing**: causing the classifier to remove the classification state for the flow

```
Client
SYN
SYN, ACK
ACK
SEQ 1 | GET /B

Classifier
TCP 80
TCP 80
TCP 80

Server
TCP 80
TCP 80
TCP 80

\text{t seconds}
```

Inserting large delays
Design
Evasion techniques

- Observation:
  - Classifiers do not retain classification results indefinitely
  - **Flushing**: causing the classifier to remove the classification state for the flow

```
Client       Classifier       Server
SYN          TCP 80         TCP 80
SYN, ACK     TCP 80         TCP 80
ACK          TCP 80         TCP 80
SEQ 1 GET /B  TCP 80         TCP 80
```

Inserting large delays

App B is unclassified

$t$ seconds
Outline

• Design and implementation
  • Traffic-classification rules detection
  • Evasion techniques

• Implementation

• Evaluation
  • Effectiveness across multiple networks
Implementation
Implementation

• Phase 1: liberate does the analysis using a replay server
Implementation

**Phase 1**

**Differentiation Detection**
- Record traffic
- Replay traffic
- Replay inverted traffic

**Characterization**
- Recorded traffic
- Binary Analysis/Inserting
- Replay traffic
- Complete?
- YES
- Classification rules
- NO

**Evasion Evaluation**
- Recorded traffic
- Effective techniques

**Evasion Deployment**
- Application traffic
- Selected evasion technique
Implementation

- Phase 1: liberate does the analysis using a replay server
- Phase 2: liberate applies evasion technique to traffic in-flight
Implementation

Phase 1

Differentiation Detection
- Record traffic
- Replay traffic
- Replay inverted traffic
- Differentiation?
  - YES
    - Content based differentiation
  - NO
    - Stop

Characterization
- Recorded traffic
- Binary Analysis/Prepending
  - Replay traffic
  - Complete?
    - YES
      - Classification rules
    - NO

Evasion Evaluation
- Recorded traffic
  - Inert packet insertion/Splicing payload/Reordering payload
  - Effective techniques

Phase 2

Evasion Deployment
- Application traffic
  - Selected evasion technique

App > Phase 2 > liberate > Proxy

Phase 1

Replay Server

17
Outline

• Design and implementation
  • Traffic-classification rules detection
  • Evasion techniques
  • Implementation

• Evaluation
  • Effectiveness across multiple networks
Evaluation
Testbed and in the wild
Evaluation
Testbed and in the wild

- Testbed evaluation
Evaluation

Testbed and in the wild

• Testbed evaluation

• Evaluation “in the wild”
Evaluation
Testbed and in the wild

• Testbed evaluation

• Evaluation “in the wild”
Evaluation
Testbed and in the wild

• Testbed evaluation

• Evaluation “in the wild”
# Evaluation Results

<table>
<thead>
<tr>
<th>Prot.</th>
<th>Technique</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th>Server Response</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>✔</td>
<td>✔</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>IP</td>
<td>Inert packet insertion</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IP</td>
<td>Lower TTL to only reach classifier</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>IP</td>
<td>Invalid Version</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>IP</td>
<td>Invalid Header Length</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>IP</td>
<td>Total Length longer than payload</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>IP</td>
<td>Total Length shorter than payload</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>IP</td>
<td>Wrong Protocol</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>IP</td>
<td>Wrong Checksum</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>IP</td>
<td>Invalid Options</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>IP</td>
<td>Deprecated Options</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>TCP</td>
<td>Wrong Sequence Number</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>TCP</td>
<td>Wrong Checksum</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>TCP</td>
<td>ACK flag not set</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>TCP</td>
<td>Invalid Data Offset</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>TCP</td>
<td>Invalid flag combination</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>UDP</td>
<td>Inert Checksum</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>UDP</td>
<td>Length longer than payload</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>UDP</td>
<td>Length shorter than payload</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>TCP</td>
<td>Payload splitting</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>IP</td>
<td>Break packet into fragments</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>TCP</td>
<td>Break packet into segments</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td></td>
<td>Payload reordering</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>IP</td>
<td>Fragmented packet, out-of-order</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>TCP</td>
<td>Segmented packet, out-of-order</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>UDP</td>
<td>UDP packets out-of-order</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td></td>
<td>Classification flushing</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>IP</td>
<td>Pause for t sec. (after match)</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>IP</td>
<td>Pause for t sec. (before match)</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>TCP</td>
<td>TTL-limited RST packet (a)</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>TCP</td>
<td>TTL-limited RST packet (b)</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
</tbody>
</table>
## Evaluation

### Example result table

<table>
<thead>
<tr>
<th>Technique</th>
<th>Test case 1</th>
<th>Example technique</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inert packet insertion</td>
<td>IP</td>
<td><img src="#" alt="Green checkmark" /> Lower TTL to only reach classifier</td>
</tr>
<tr>
<td></td>
<td>TCP</td>
<td><img src="#" alt="Green checkmark" /> Wrong sequence number</td>
</tr>
<tr>
<td></td>
<td>UDP</td>
<td><img src="#" alt="Green checkmark" /> Wrong checksum</td>
</tr>
<tr>
<td>Payload Splitting</td>
<td><img src="#" alt="Red cross" /></td>
<td></td>
</tr>
<tr>
<td>Payload Reordering</td>
<td><img src="#" alt="Green checkmark" /></td>
<td>Reverse the transmission of first two fragments</td>
</tr>
<tr>
<td>Classification flushing</td>
<td><img src="#" alt="Red cross" /></td>
<td></td>
</tr>
</tbody>
</table>
## Evaluation

### Example result table

<table>
<thead>
<tr>
<th>Technique</th>
<th>Test case 1</th>
<th>Example technique</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inert packet insertion</td>
<td>IP</td>
<td>Lower TTL to only reach classifier</td>
</tr>
<tr>
<td></td>
<td>TCP</td>
<td>Wrong sequence number</td>
</tr>
<tr>
<td></td>
<td>UDP</td>
<td>Wrong checksum</td>
</tr>
<tr>
<td>Payload Splitting</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Payload Reordering</td>
<td></td>
<td>Reverse the transmission of first two fragments</td>
</tr>
<tr>
<td>Classification flushing</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Evaluation

#### Example result table

<table>
<thead>
<tr>
<th>Technique</th>
<th>Test case 1</th>
<th>Example technique</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inert packet insertion</td>
<td>IP</td>
<td>![Checkmark] Lower TTL to only reach classifier</td>
</tr>
<tr>
<td></td>
<td>TCP</td>
<td>![Checkmark] Wrong sequence number</td>
</tr>
<tr>
<td></td>
<td>UDP</td>
<td>![Checkmark] Wrong checksum</td>
</tr>
<tr>
<td>Payload Splitting</td>
<td>![X]</td>
<td></td>
</tr>
<tr>
<td>Payload Reordering</td>
<td>![Checkmark]</td>
<td>Reverse the transmission of first two fragments</td>
</tr>
<tr>
<td>Classification flushing</td>
<td>![X]</td>
<td></td>
</tr>
</tbody>
</table>
## Evaluation

### Example result table

<table>
<thead>
<tr>
<th>Technique</th>
<th>Test case 1</th>
<th>Example technique</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inert packet insertion</td>
<td>IP</td>
<td>Lower TTL to only reach classifier</td>
</tr>
<tr>
<td></td>
<td>TCP</td>
<td>Wrong sequence number</td>
</tr>
<tr>
<td></td>
<td>UDP</td>
<td>Wrong checksum</td>
</tr>
<tr>
<td>Payload Splitting</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Payload Reordering</td>
<td></td>
<td>Reverse the transmission of first two fragments</td>
</tr>
<tr>
<td>Classification flushing</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Evaluation

### Testbed results

<table>
<thead>
<tr>
<th>Technique</th>
<th>Testbed</th>
<th>Example technique</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inert packet insertion</td>
<td>IP</td>
<td>Lower TTL to only reach classifier</td>
</tr>
<tr>
<td></td>
<td>TCP</td>
<td>Wrong sequence number</td>
</tr>
<tr>
<td></td>
<td>UDP</td>
<td>Wrong checksum</td>
</tr>
<tr>
<td>Payload Splitting</td>
<td></td>
<td>Break packet into two IP fragments</td>
</tr>
<tr>
<td>Payload Reordering</td>
<td></td>
<td>Reverse the transmission of first two fragments</td>
</tr>
<tr>
<td>Classification flushing</td>
<td></td>
<td>TTL-limited RST packet before classification</td>
</tr>
</tbody>
</table>
## Evaluation

### Testbed results

<table>
<thead>
<tr>
<th>Technique</th>
<th>Testbed</th>
<th>Example technique</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inert packet insertion</td>
<td>IP</td>
<td>Lower TTL to only reach classifier</td>
</tr>
<tr>
<td></td>
<td>TCP</td>
<td>Wrong sequence number</td>
</tr>
<tr>
<td></td>
<td>UDP</td>
<td>Wrong checksum</td>
</tr>
<tr>
<td>Payload Splitting</td>
<td></td>
<td>Break packet into two IP fragments</td>
</tr>
<tr>
<td>Payload Reordering</td>
<td></td>
<td>Reverse the transmission of first two fragments</td>
</tr>
<tr>
<td>Classification flushing</td>
<td></td>
<td>TTL-limited RST packet before classification</td>
</tr>
</tbody>
</table>

- Efficiency:
- **One-time overhead** (phase 1): 13 minutes
Evaluation
Testbed results

- Efficiency:
  - **One-time overhead** (phase 1): 13 minutes
Evaluation
Testbed results

- Efficiency:
  - **One-time overhead** (phase 1): 13 minutes
  - Run-time overhead (phase 2): tens of bytes per flow
Evaluation

Testbed results

<table>
<thead>
<tr>
<th>Technique</th>
<th>Testbed</th>
<th>Example technique</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inert packet insertion</td>
<td></td>
<td>Lower TTL to only reach classifier</td>
</tr>
<tr>
<td>IP</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>TCP</td>
<td>✔️</td>
<td>Wrong sequence number</td>
</tr>
<tr>
<td>UDP</td>
<td>✔️</td>
<td>Wrong checksum</td>
</tr>
<tr>
<td>Payload Splitting</td>
<td>✔️</td>
<td>Break packet into two IP fragments</td>
</tr>
<tr>
<td>Payload Reordering</td>
<td>✔️</td>
<td>Reverse the transmission of first two fragments</td>
</tr>
<tr>
<td>Classification flushing</td>
<td>✔️</td>
<td>TTL-limited RST packet before classification</td>
</tr>
</tbody>
</table>

- Efficiency:
  - **One-time overhead** (phase 1): 13 minutes
  - Run-time overhead (phase 2): tens of bytes per flow

- Effectiveness:
  - All types of techniques were effective in testbed
## Evaluation

**T mobile ‘Binge On’**

<table>
<thead>
<tr>
<th>Technique</th>
<th>Testbed</th>
<th>T mobile</th>
<th>Example technique</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inert packet insertion</td>
<td>![Checkmark]</td>
<td>![Checkmark]</td>
<td>Lower TTL to only reach classifier</td>
</tr>
<tr>
<td>IP</td>
<td>![Checkmark]</td>
<td>![Checkmark]</td>
<td></td>
</tr>
<tr>
<td>TCP</td>
<td>![Checkmark]</td>
<td>![X]</td>
<td></td>
</tr>
<tr>
<td>UDP</td>
<td>![Checkmark]</td>
<td>![Checkmark]</td>
<td></td>
</tr>
<tr>
<td>Payload Splitting</td>
<td>![Checkmark]</td>
<td>![Checkmark]</td>
<td>Break packet into five TCP segments</td>
</tr>
<tr>
<td>Payload Reordering</td>
<td>![Checkmark]</td>
<td>![Checkmark]</td>
<td>Reverse the transmission of first two segments</td>
</tr>
<tr>
<td>Classification flushing</td>
<td>![Checkmark]</td>
<td>![Checkmark]</td>
<td>TTL-limited RST packet before classification</td>
</tr>
</tbody>
</table>
Evaluation
T mobile ‘Binge On’

<table>
<thead>
<tr>
<th>Technique</th>
<th>Testbed</th>
<th>T mobile</th>
<th>Example technique</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inert packet insertion</td>
<td>IP</td>
<td>✔️</td>
<td>Lower TTL to only reach classifier</td>
</tr>
<tr>
<td></td>
<td>TCP</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td></td>
<td>UDP</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>Payload Splitting</td>
<td>✔️</td>
<td>✔️</td>
<td>Break packet into five TCP segments</td>
</tr>
<tr>
<td>Payload Reordering</td>
<td>✔️</td>
<td>✔️</td>
<td>Reverse the transmission of first two segments</td>
</tr>
<tr>
<td>Classification flushing</td>
<td>✔️</td>
<td>✔️</td>
<td>TTL-limited RST packet before classification</td>
</tr>
</tbody>
</table>

- Classified video (HTTP/S) was throttled to 1.5 Mbps and zero-rated
- Efficiency:
  - **One-time overhead** (phase 1) : 30 minutes
  - Run-time overhead (phase 2) : tens of bytes per flow
## Evaluation

### T mobile ‘Binge On’

<table>
<thead>
<tr>
<th>Technique</th>
<th>Testbed</th>
<th>T mobile</th>
<th>Example technique</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inert packet insertion</td>
<td>IP</td>
<td>✔️</td>
<td>Lower TTL to only reach classifier</td>
</tr>
<tr>
<td></td>
<td>TCP</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td></td>
<td>UDP</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>Payload Splitting</td>
<td>✔️</td>
<td>✔️</td>
<td>Break packet into five TCP segments</td>
</tr>
<tr>
<td>Payload Reordering</td>
<td>✔️</td>
<td>✔️</td>
<td>Reverse the transmission of first two segments</td>
</tr>
<tr>
<td>Classification flushing</td>
<td>✔️</td>
<td>✔️</td>
<td>TTL-limited RST packet before classification</td>
</tr>
</tbody>
</table>

- Classified video (HTTP/S) was throttled to 1.5 Mbps and zero-rated
- Efficiency:
  - **One-time overhead** (phase 1) : 30 minutes
  - Run-time overhead (phase 2) : tens of bytes per flow
- Effectiveness:
  - UDP traffic (e.g., Youtube video in QUIC) was not classified
# Evaluation

T mobile ‘Binge On’

<table>
<thead>
<tr>
<th>Technique</th>
<th>Testbed</th>
<th>T mobile</th>
<th>Example technique</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inert packet insertion</td>
<td>IP</td>
<td>✓</td>
<td>Lower TTL to only reach classifier</td>
</tr>
<tr>
<td></td>
<td>TCP</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td></td>
<td>UDP</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td><strong>Payload Splitting</strong></td>
<td></td>
<td>✓</td>
<td>Break packet into five TCP segments</td>
</tr>
<tr>
<td>Payload Reordering</td>
<td></td>
<td>✓</td>
<td>Reverse the transmission of first two segments</td>
</tr>
<tr>
<td>Classification flushing</td>
<td></td>
<td>✓</td>
<td>TTL-limited RST packet before classification</td>
</tr>
</tbody>
</table>

- Classified video (HTTP/S) was throttled to 1.5 Mbps and zero-rated
- Efficiency:
  - **One-time overhead** (phase 1) : 30 minutes
  - Run-time overhead (phase 2) : tens of bytes per flow
- Effectiveness:
  - UDP traffic (e.g., Youtube video in QUIC) was not classified
  - Breaking packet into 5 TCP segments evaded classification

- Classification flushing

---

23
Evaluation
T mobile ‘Binge On’

<table>
<thead>
<tr>
<th>Technique</th>
<th>Testbed</th>
<th>T mobile</th>
<th>Example technique</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inert packet insertion</td>
<td>IP</td>
<td>✔️</td>
<td>Lower TTL to only reach classifier</td>
</tr>
<tr>
<td></td>
<td>TCP</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td></td>
<td>UDP</td>
<td>🟢</td>
<td></td>
</tr>
<tr>
<td>Payload Splitting</td>
<td>✔️</td>
<td>✔️</td>
<td>Break packet into five TCP segments</td>
</tr>
<tr>
<td>Payload Reordering</td>
<td>✔️</td>
<td>✔️</td>
<td>Reverse the transmission of first two segments</td>
</tr>
<tr>
<td>Classification flushing</td>
<td>✔️</td>
<td>✔️</td>
<td>TTL-limited RST packet before classification</td>
</tr>
</tbody>
</table>

- Classified video (HTTP/S) was throttled to 1.5 Mbps and zero-rated
- Efficiency:
  - **One-time overhead** (phase 1) : 30 minutes
  - Run-time overhead (phase 2) : tens of bytes per flow
- Effectiveness:
  - UDP traffic (e.g., Youtube video in QUIC) was not classified
  - Breaking packet into 5 TCP segments evaded classification
  - Reversing the order of initial packets was effective
## Evaluation

### T mobile ‘Binge On’

<table>
<thead>
<tr>
<th>Technique</th>
<th>Testbed</th>
<th>T mobile</th>
<th>Example technique</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inert packet insertion</td>
<td>IP</td>
<td>✓</td>
<td>Lower TTL to only reach classifier</td>
</tr>
<tr>
<td></td>
<td>TCP</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td></td>
<td>UDP</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Payload Splitting</td>
<td>IP</td>
<td>✓</td>
<td>Break packet into five TCP segments</td>
</tr>
<tr>
<td></td>
<td>TCP</td>
<td>✓</td>
<td>Reverse the transmission of first two segments</td>
</tr>
<tr>
<td></td>
<td>UDP</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Payload Reordering</td>
<td>IP</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TCP</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td></td>
<td>UDP</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Classification flushing</td>
<td>IP</td>
<td>✓</td>
<td>TTL-limited RST packet before classification</td>
</tr>
<tr>
<td></td>
<td>TCP</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td></td>
<td>UDP</td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>

- Classified video (HTTP/S) was throttled to 1.5 Mbps and zero-rated
- Efficiency:
  - **One-time overhead** (phase 1) : 30 minutes
  - Run-time overhead (phase 2) : tens of bytes per flow
- Effectiveness:
  - UDP traffic (e.g., Youtube video in QUIC) was not classified
  - Breaking packet into 5 TCP segments evaded classification
  - Reversing the order of initial packets was effective
Evaluation
The Great Firewall of China

<table>
<thead>
<tr>
<th>Technique</th>
<th>Testbed</th>
<th>T mobile</th>
<th>GFC</th>
<th>Example technique</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inert packet insertion</td>
<td>IP</td>
<td>✔</td>
<td>✔</td>
<td>Lower TTL to only reach classifier</td>
</tr>
<tr>
<td></td>
<td>TCP</td>
<td>❌</td>
<td>✔</td>
<td>Wrong Checksum</td>
</tr>
<tr>
<td></td>
<td>UDP</td>
<td>✔</td>
<td>❌</td>
<td></td>
</tr>
<tr>
<td>Payload Splitting</td>
<td>✔</td>
<td>✔</td>
<td>❌</td>
<td></td>
</tr>
<tr>
<td>Payload Reordering</td>
<td>✔</td>
<td>✔</td>
<td>❌</td>
<td></td>
</tr>
<tr>
<td>Classification flushing</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>Pause for $t$ seconds before classification</td>
</tr>
</tbody>
</table>
## Evaluation

### The Great Firewall of China

<table>
<thead>
<tr>
<th>Technique</th>
<th>Testbed</th>
<th>T mobile</th>
<th>GFC</th>
<th>Example technique</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inert packet insertion</td>
<td>IP</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>TCP</td>
<td>✓</td>
<td>✗</td>
<td>Lower TTL to only reach classifier</td>
</tr>
<tr>
<td></td>
<td>UDP</td>
<td>✓</td>
<td>✗</td>
<td>Wrong Checksum</td>
</tr>
<tr>
<td>Payload Splitting</td>
<td></td>
<td>✓</td>
<td>✗</td>
<td></td>
</tr>
<tr>
<td>Payload Reordering</td>
<td></td>
<td>✓</td>
<td>✗</td>
<td></td>
</tr>
<tr>
<td>Classificationflushing</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>Pause for $t$ seconds before classification</td>
</tr>
</tbody>
</table>

- Classified HTTP content was blocked by 3-5 RST packets
- Efficiency:
  - **One-time overhead** (phase 1): 20 minutes
  - Run-time overhead (phase 2): tens of bytes per flow
## Evaluation

### The Great Firewall of China

<table>
<thead>
<tr>
<th>Technique</th>
<th>Testbed</th>
<th>T mobile</th>
<th>GFC</th>
<th>Example technique</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inert packet insertion</td>
<td>IP</td>
<td>✓</td>
<td>✓</td>
<td>Lower TTL to only reach classifier</td>
</tr>
<tr>
<td></td>
<td>TCP</td>
<td>✓</td>
<td>✗</td>
<td>Wrong Checksum</td>
</tr>
<tr>
<td></td>
<td>UDP</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Payload Splitting</td>
<td></td>
<td>✓</td>
<td>✗</td>
<td></td>
</tr>
<tr>
<td>Payload Reordering</td>
<td></td>
<td>✓</td>
<td>✗</td>
<td></td>
</tr>
<tr>
<td>Classification flushing</td>
<td></td>
<td>✓</td>
<td></td>
<td>Pause for $t$ seconds before classification</td>
</tr>
</tbody>
</table>

- Classified HTTP content was blocked by 3-5 RST packets
- Efficiency:
  - **One-time overhead** (phase 1) : 20 minutes
  - Run-time overhead (phase 2) : tens of bytes per flow
- Effectiveness:
  - Both IP/ TCP inert insertion succeeded
## Evaluation

The Great Firewall of China

<table>
<thead>
<tr>
<th>Technique</th>
<th>Testbed</th>
<th>T mobile</th>
<th>GFC</th>
<th>Example technique</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inert packet insertion</td>
<td>IP</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>TCP</td>
<td></td>
<td>✗</td>
<td></td>
</tr>
<tr>
<td></td>
<td>UDP</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Payload Splitting</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>Payload Reordering</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>Classification flushing</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

- Classified HTTP content was blocked by 3-5 RST packets
- Efficiency:
  - **One-time overhead** (phase 1) : 20 minutes
  - Run-time overhead (phase 2) : tens of bytes per flow
- Effectiveness:
  - Both IP/ TCP inert insertion succeeded
  - Flushing classification by pausing succeeded
Evaluation
The Great Firewall of China

- Classified HTTP content was blocked by 3-5 RST packets
- Efficiency:
  - **One-time overhead** (phase 1): 20 minutes
  - Run-time overhead (phase 2): tens of bytes per flow
- Effectiveness:
  - Both IP/TCP inert insertion succeeded
  - Flushing classification by pausing succeeded
Evaluation
The Great Firewall of China

Time-of-day effects when flushing classification
Evaluation

The Great Firewall of China

Time-of-day effects when flushing classification

![Graph showing time-of-day effects with different success rates.](image-url)
Evaluation
The Great Firewall of China

Time-of-day effects when flushing classification
Evaluation
The Great Firewall of China

Time-of-day effects when flushing classification

- Red circle: None succeed
- Green circle: 1/3 succeed
- Light green circle: 2/3 succeed
- Dark green circle: All succeed

60 seconds successfully evaded
2:30 AM

60 seconds
Evaluation

The Great Firewall of China

Time-of-day effects when flushing classification

- 240 seconds failed to evade
- 60 seconds successfully evaded

Legend:
- Red: None succeed
- Green: 1/3 succeed
- Yellow: 2/3 succeed
- Black: All succeed
Evaluation
The Great Firewall of China

Time-of-day effects when flushing classification

- None succeed
- 1/3 succeed
- 2/3 succeed
- All succeed
Evaluation
The Great Firewall of China

Time-of-day effects when flushing classification

quiet hours (4:00 AM to 9:00 AM) — using long delays did not evade
Evaluation
The Great Firewall of China

Time-of-day effects when flushing classification

quiet hours (4:00 AM to 9:00 AM) — using long delays did not evade

busy hours (3:00 PM to 10:00 PM) — using short delays evaded
Conclusion

- A tool that **automatically** and **efficiently** evades differentiation
- A **taxonomy of evasion techniques**
- An **empirical measurement** of traffic classifiers
- Liberate **evaded** classifiers with low run-time overhead
- **Public, open-source** tools and datasets
- **Future work:** more resilient evasion techniques
Thanks

For more details about liberate, code, and data:
http://dd.meddle.mobi/liberate

NSF

Google