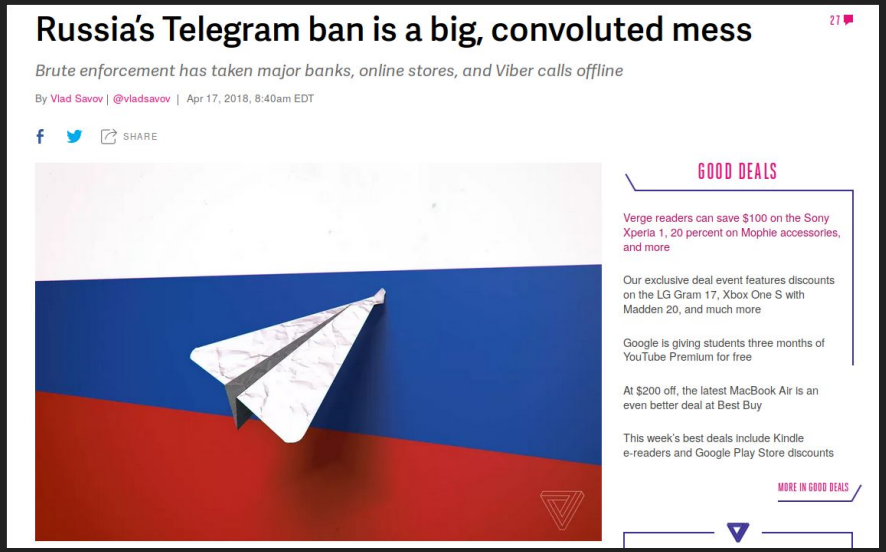
Improving Meek With Adversarial Techniques

Steven Sheffey, Ferrol Aderholdt

FOCI '19

Domain Fronting



Russia's Telegram ban is a big, convoluted mess

27

Brute enforcement has taken major banks, online stores, and Viber calls offline

By Vlad Savov | @vladsavov | Apr 17, 2018, 8:40am EDT

SHARE

GOOD DEALS

Verge readers can save $100 on the Sony Xperia 1, 20 percent on Mophie accessories, and more

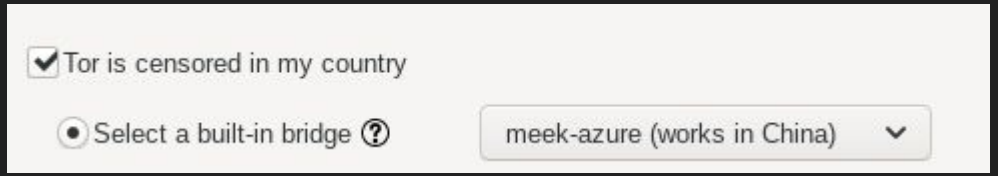
Our exclusive deal event features discounts on the LG Gram 17, Xbox One S with Madden 20, and much more

Google is giving students three months of YouTube Premium for free

At $200 off, the latest MacBook Air is an even better deal at Best Buy

This week's best deals include Kindle e-readers and Google Play Store discounts

MORE IN GOOD DEALS



Domain Fronting

Data Centre ▸ Networks

Google kills off domain fronting – and so secure comms just got tougher

Cloud tech tweaks end anti-censorship workaround

By Thomas Claburn in San Francisco 19 Apr 2018 at 22:01 11 SHARE ▼

Amazon blocks domain fronting, threatens to shut down Signal's account

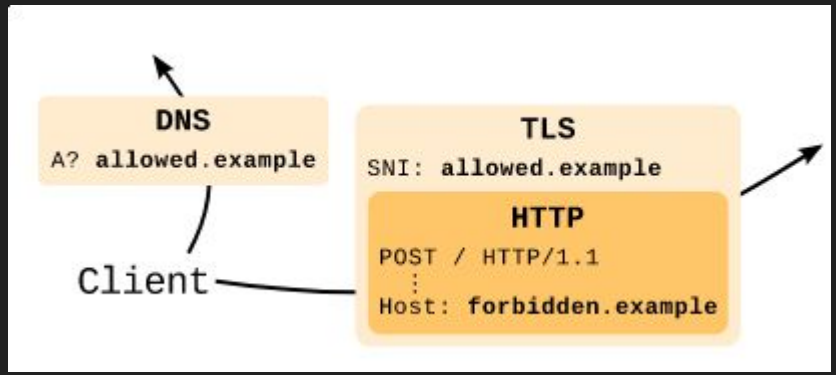
Move makes evasion of Middle Eastern countries' censorship of Signal more difficult.

SEAN GALLAGHER - MAY 2, 2018 6:50 PM UTC

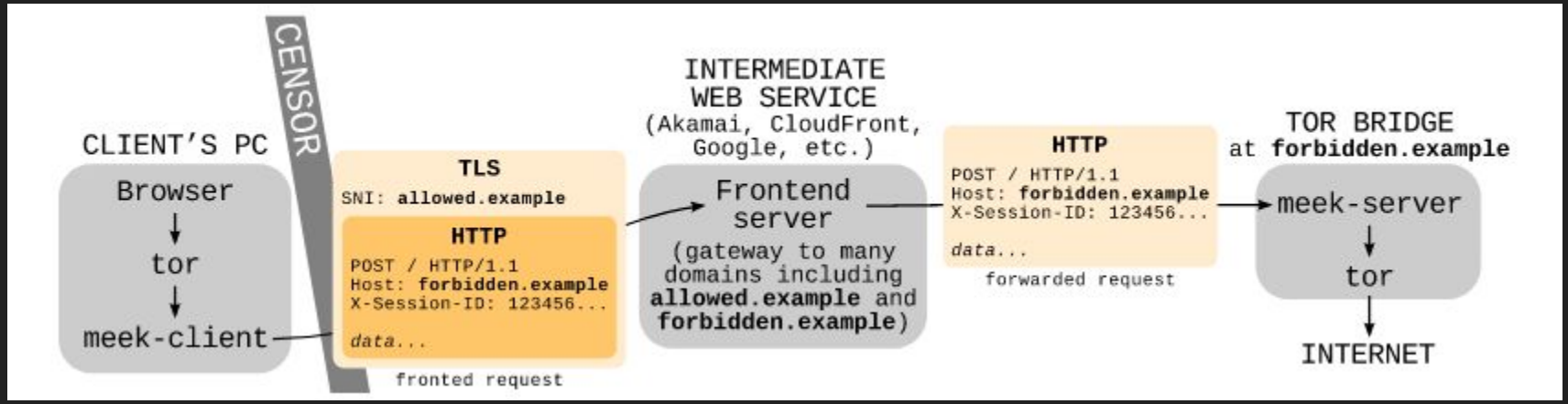
Enlarge / Moxie Marlinspike, founder of Signal.

Domain Fronting

- Makes a DNS request to an allowed host
- Indicates an allowed host in the SNI fields of the HTTPS traffic
- Sends traffic to a forbidden host by modifying the Host header
- All basic metadata fields indicate traffic to an allowed host



Meek



Meek uses domain fronting to tunnel Tor traffic

Related Work

- Machine learning attacks using packet sizes, inter-arrival times, ACK frequency can differentiate Meek from regular HTTPS traffic
  - Wang et al. in “Seeing through Network-Protocol Obfuscation” used a decision tree and achieved a FPR as low as 0.0002
  - Yao et al. in “Meek-based tor traffic identification with hidden markov model.” used a Markov model and achieved an accuracy of 99.98%
  - Nasr et al. in “Strong flow correlation attacks on tor using deep learning” were able to deanonymize Meek traffic with a FPR of 0.0005
- Verma et al in “Network Traffic Obfuscation: An Adversarial Machine Learning Approach” proposes shaping traffic to mimic generic protocol types(BULK, DATABASE, MAIL , SERVICES, P2P, WWW)

Proposed Approach

- Take features from Meek and HTTPs traffic commonly used to identify Meek traffic, and form a statistical signature
- Use a GAN to transform this signature in a way that makes Meek traffic less identifiable

Future goal:

- Use this modified distribution to inform packet shaping strategy

Data Collection

- Navigate to Alexa top 10K in Firefox with both regular HTTPS and Meek (meek-azure)
- Contained in Docker to isolate collection process and allow parallelization
- Collecting samples of Meek traffic requires restarting the Meek process. Parallelization escapes this bottleneck.
- Collected a total of 60k PCAPs
  - Dataset H - generated from residential internet
  - Dataset U - generated from MTSU internet
  - Dataset A - generated from AWS

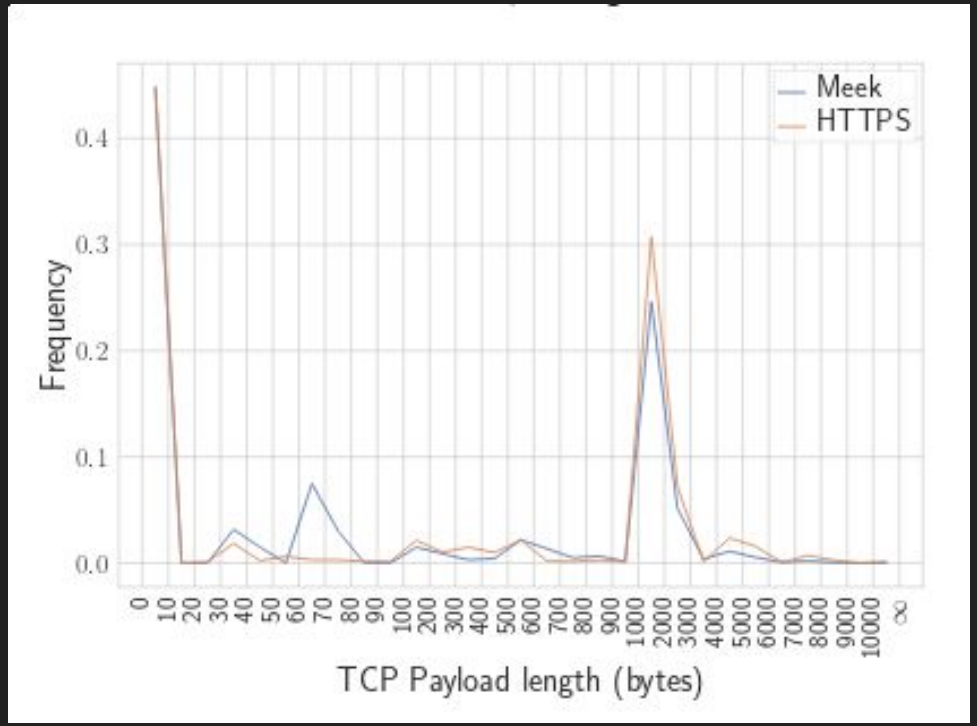
Data Collection Limitations

- Collection environments undiverse
  - 2 locations in Tennessee, and one in a Virginia datacenter
- Generic
  - Meek vs HTTPS for all traffic, and not for just traffic on the domain fronted CDN

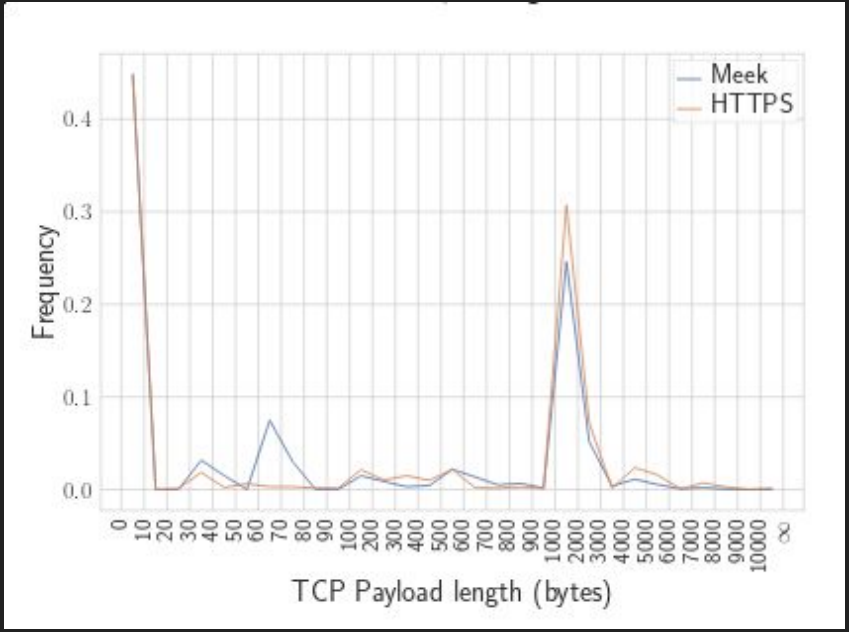
Feature Extraction

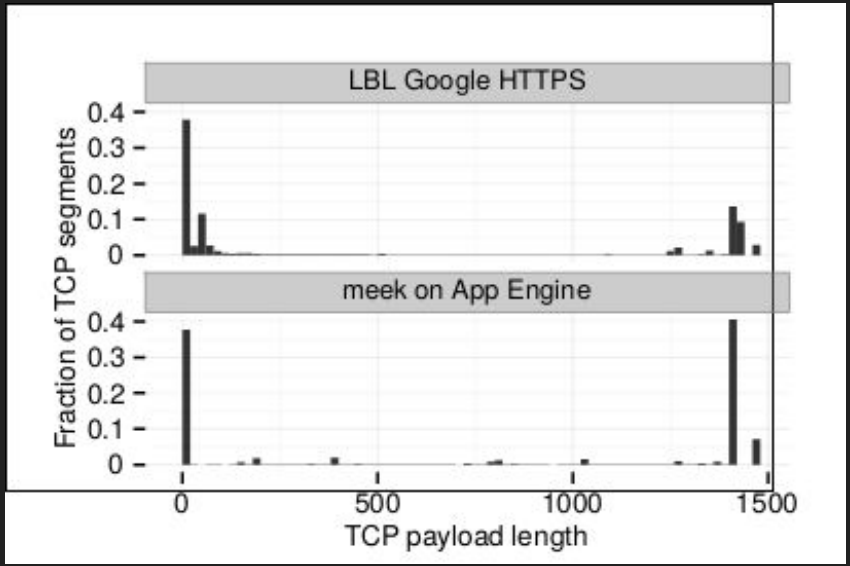
- For each PCAP:
  - Aggregate packets into flows and discard all non-HTTPS packets
  - Form histograms with logarithmic bins of:
    - Packet sizes
    - Interarrival times (to and from the client)

Payload Length Distribution

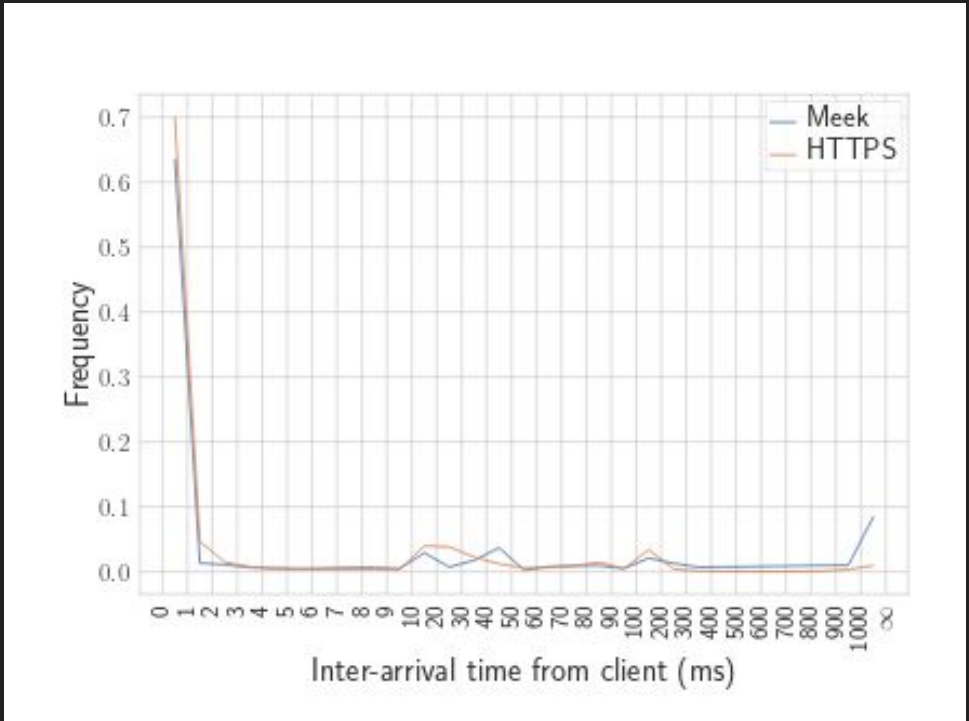


Differences in Observations

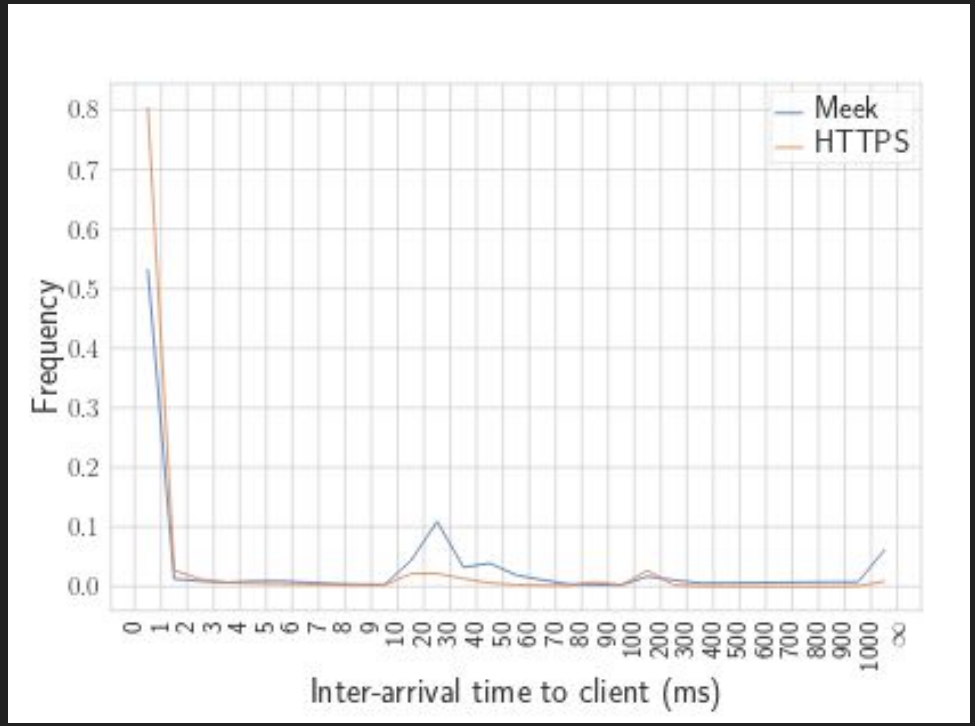




Inter-arrival Time Distribution (from client)



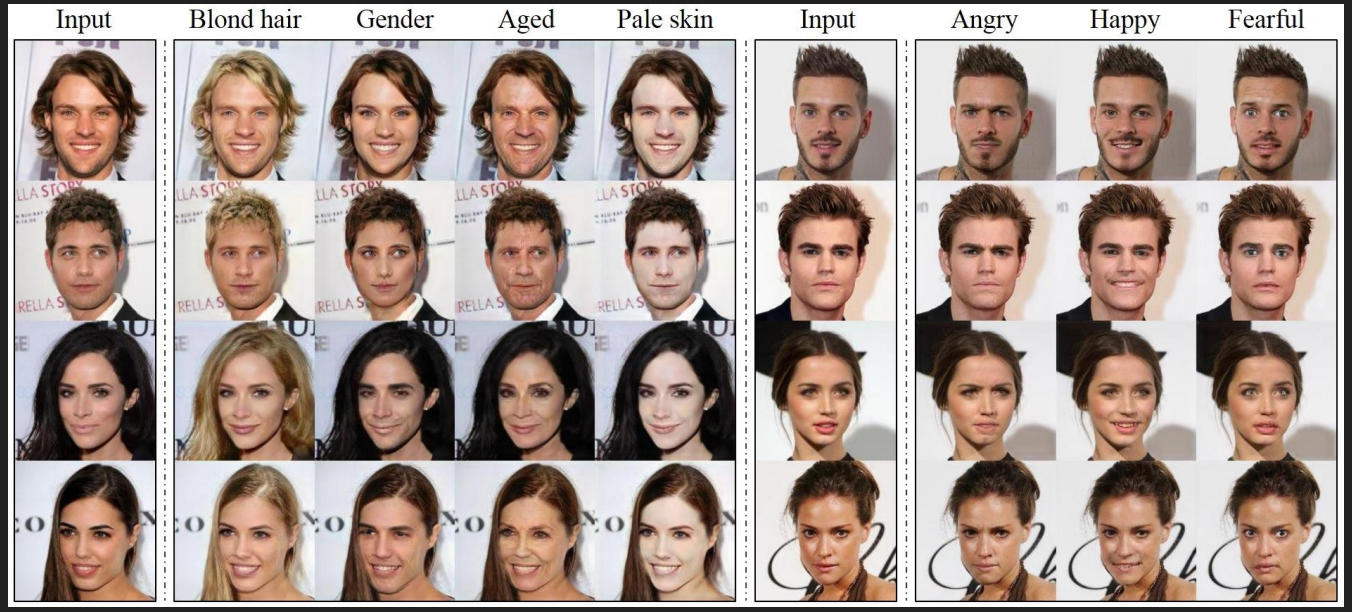
Inter-arrival Time Distribution (to client)



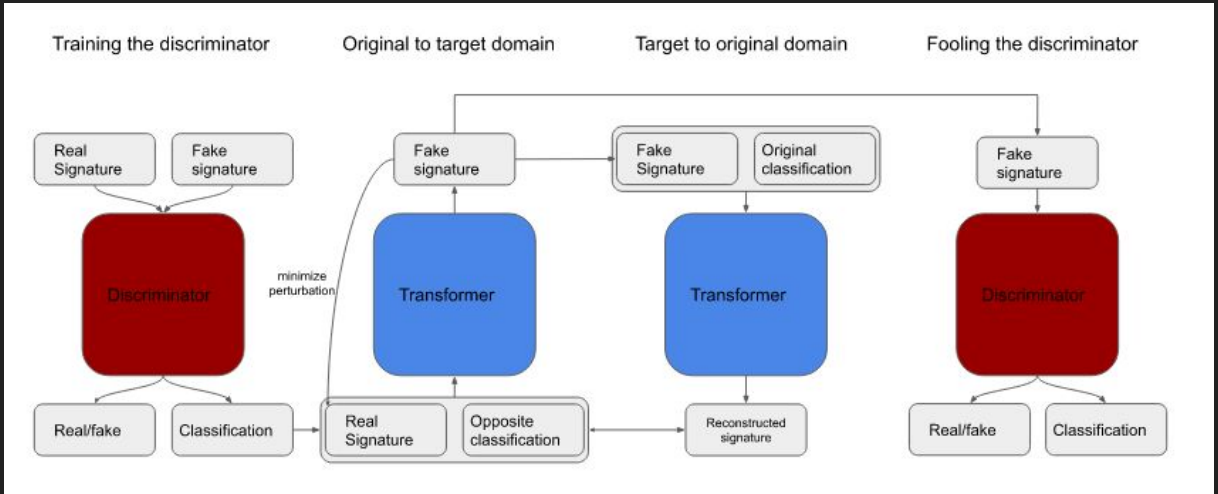
Adversarial Neural Networks

- Generator
  - produces inputs from noise
  - can also modify inputs (transformer)
- Discriminator
  - classifies inputs
- Both are trained simultaneously until an equilibrium is reached
- Well-suited to traffic obfuscation

StarGAN


Input
Blond hair
Gender
Aged
Pale skin
Input
Angry
Happy
Fearful

GAN Training



- Train the discriminator to classify real signatures as (real, their class)
- Train the discriminator to identify transformed signatures as fake
- Train the transformer to transform a signature in a way that tricks the classifier into classifying it as a specific class, while also minimizing perturbation
- Train the transformer to transform transformed signatures so that they are classified by the discriminator as their original class

GAN Objectives

- Neural network training aims to minimize loss
- Binary cross-entropy - Used to minimize differences in classification
  - Is the signature Meek or HTTPS
  - Is the signature real or fake?
- Gradient penalty - Used in GANs to stabilize the training process
- Mean Absolute error - Used to minimize differences in signatures
  - Does the transformer properly transform traffic back to its original class?
  - Does the transformer minimize modification?
- $Dloss = DLoss_{src} + Dloss_{cls} + DLoss'_{src} + 10DLoss_{gp}$
- $TLoss = TLoss_{cls} + TLoss_{src} + 10TLoss_{pert} + 10Tloss_{rec}$

Classifiers

- Decision tree
  - same model as Wang et al.
- Naive Neural Network
  - same architecture as the discriminator
- Informed Neural Network
  - same architecture as the discriminator
  - Trained on both untransformed and transformed samples

Training process

- Performed on each dataset
- 30% GAN training set
  - Trains a discriminator to classify traffic signatures as Normal/Meek and Real/Fake and a transformer capable of transforming traffic signatures given a target class
- 20% Classifier training set
  - Trains a classifier to classify signatures as Normal/Meek
- 50% Classifier testing set
  - Tests the classifier’s ability to identify traffic signatures and transformed signatures
- 6-fold cross validation over all permutations of these splits

Results

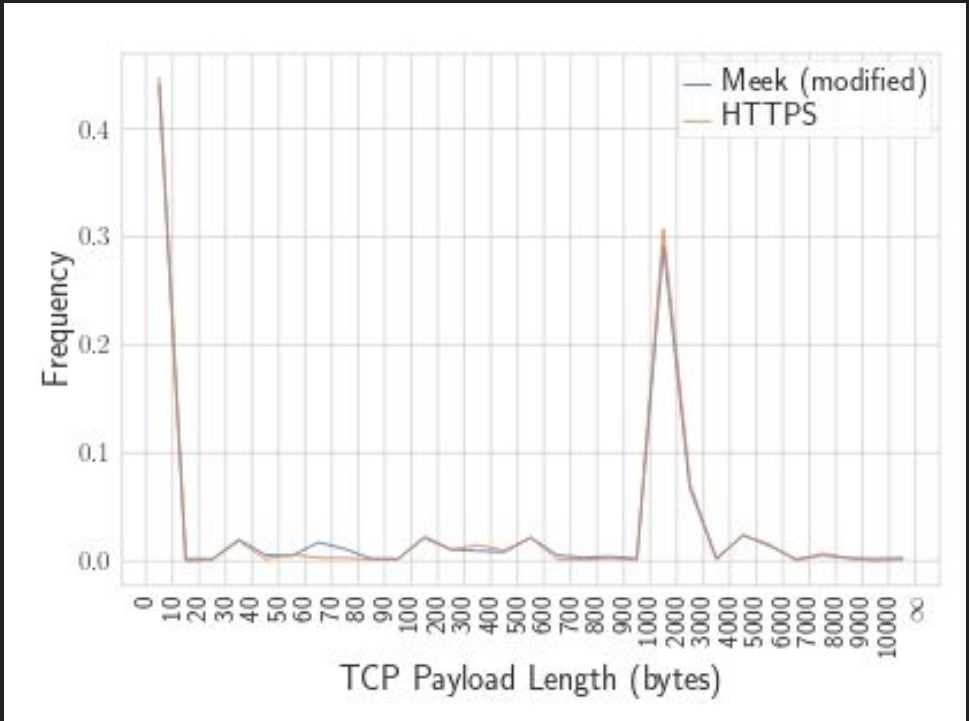
Data	Classifier	Baseline PR-AUC	Modified PR-AUC
H	Naive NN	0.999	0.309
	Informed NN	0.915	0.583
	Decision Tree	0.998	0.476
U	Naive NN	1.000	0.309
	Informed NN	0.999	0.428
	Decision Tree	1.000	0.503
A	Naive NN	1.000	0.309
	Informed NN	0.999	0.309
	Decision Tree	0.999	0.503
Avg	Naive NN	1.000	0.309
	Informed NN	0.971	0.440
	Decision Tree	0.999	0.494

Table 1: Effect of transformer on PR-AUC

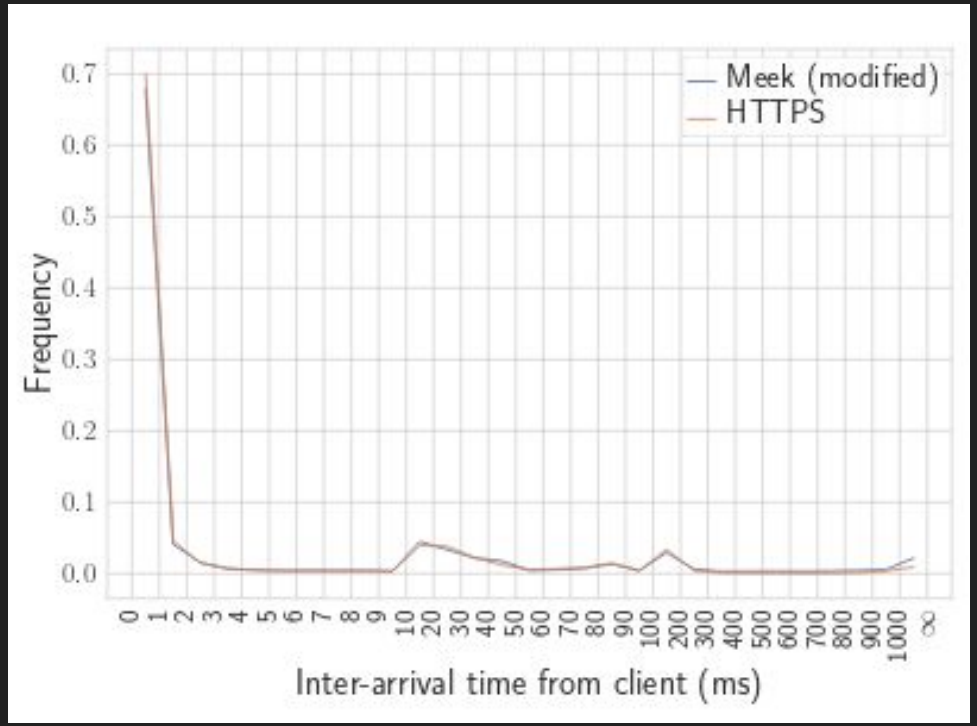
Data	Classifier	Baseline FPR	Modified FPR
H	Naive NN	0.005	1.000
	Informed NN	0.654	0.667
	Decision Tree	0.001	0.999
U	Naive NN	0.000	1.000
	Informed NN	0.351	0.501
	Decision Tree	0.000	1.000
A	Naive NN	0.002	1.000
	Informed NN	0.630	0.833
	Decision Tree	0.001	0.510
Avg	Naive NN	0.002	1.000
	Informed NN	0.545	0.667
	Decision Tree	0.001	0.836

Table 2: Effect of transformer on FPR

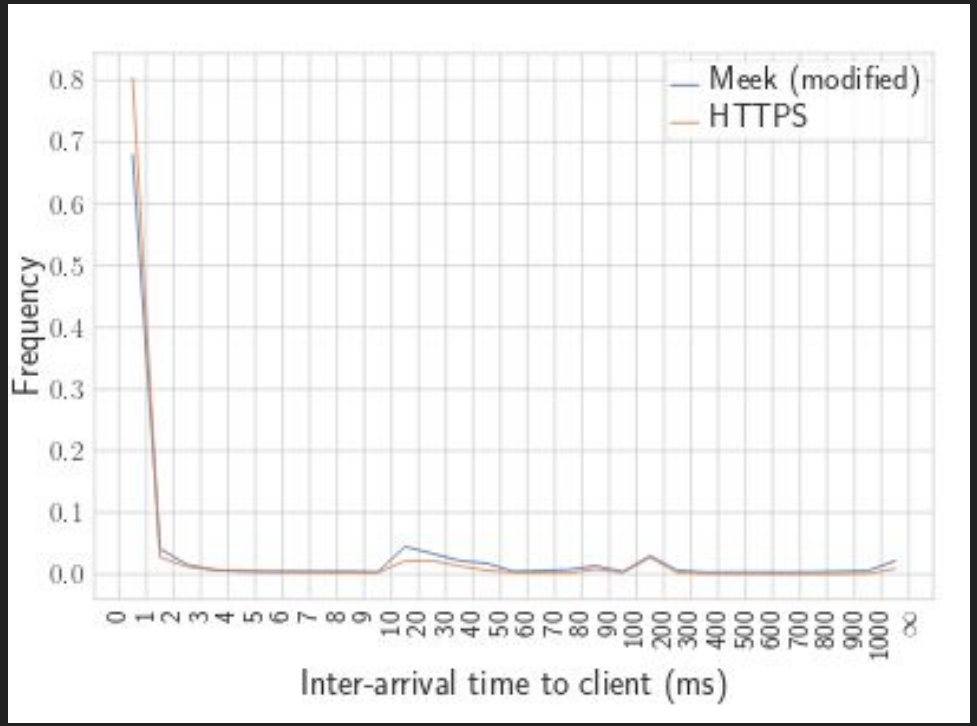
Payload Length Distribution



Inter-arrival Time Distribution (from client)



Inter-arrival Time Distribution (to client)



Future work

- Implementation.
  - Right now, the transformed distribution represents a theoretical change in packet shaping, but implementing this distribution is a more difficult problem.
- Increase data diversity.
  - Models generated based on “normal traffic” for a region may not be useful for other regions. A streamlined approach that trains a model based on a user’s “regular” browsing traffic could be useful.
  - The default meek bridge uses ajax.aspnetcdn.com, a website that typically serves single JS files. Mimicking the traffic patterns of generic HTTPS may not match traffic patterns to this specific host.

Availability

- https://github.com/starfys/packet_captor_sakura
- srs6p@mtmail.mtsu.edu