Privacy at the communication layer

Seeing Through Network-Protocol Obfuscation
Wang, Dyer, Akella, Ristenpart, and Shrimpton 2015

CS-721

Carmela Troncoso
http://carmelatroncoso.com/

Beyond anonymity: Censorship prevention

Adversary's goal: prevent communication between two parties

2-step process:


finding the flow: fingerprinting


prevent communication: direct censor

Sheharbano Khattak, Tariq Elahi, Laurent Simon, Colleen M. Swanson, Steven J. Murdoch, and Ian Goldberg.
SoK: Making Sense of Censorship Resistance Systems. Proceedings on Privacy Enhancing Technologies (PoPETS 2016)

Finding the flow: fingerprinting

Destination:

IP addresses, hosts, ports,... Tor (or other anon comm)

Content:

protocol-strings, keywords, domains, http hosts,... Encryption

Flow properties:

length, inter-arrival times, bursts, Obfuscation, mimic

Protocol semantics:

protocol behavior (mostly active attacks)

Sheharbano Khattak, Tariq Elahi, Laurent Simon, Colleen M. Swanson, Steven J. Murdoch, and Ian Goldberg.
SoK: Making Sense of Censorship Resistance Systems. Proceedings on Privacy Enhancing Technologies (PoPETS 2016)

FINDING THE FLOW: FINGERPRINTING

DESTINATION:

IP addresses, hosts, ports,... TOR (OR OTHER ANON COMM)

CONTENT:

protocol-strings, keywords, domains, http hosts,... ENCRYPTION

FLOW PROPERTIES:

length, inter-arrival times, bursts, OBFUSCATION, MIMIC

PROTOCOL SEMANTICS:

protocol behavior (mostly active attacks)

Sheharbano Khattak, Tariq Elahi, Laurent Simon, Colleen M. Swanson, Steven J. Murdoch, and Ian Goldberg.
SoK: Making Sense of Censorship Resistance Systems. Proceedings on Privacy Enhancing Technologies (PoPETS 2016)

ScrambleSuit

Pseudo-random payload: ScrambleSuit computationally indistinguishable from randomness. i.e., no DPI fingerprints.

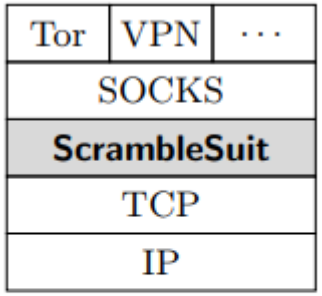


Figure 1: ScrambleSuit's protocol stack.

Philipp Winter, Tobias Pulls, and Juergen Fuss
ScrambleSuit: A Polymorphic Network Protocol to Circumvent Censorship (WPES13)

ScrambleSuit

Pseudo-random payload: ScrambleSuit computationally indistinguishable from randomness. i.e., no DPI fingerprints.

Polymorphic: changes shape to hinder classification.

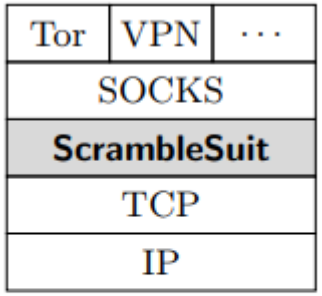


Figure 1: ScrambleSuit's protocol stack.

Philipp Winter, Tobias Pulls, and Juergen Fuss
ScrambleSuit: A Polymorphic Network Protocol to Circumvent Censorship (WPES13)

ScrambleSuit

Pseudo-random payload: ScrambleSuit computationally indistinguishable from randomness. i.e., no DPI fingerprints.

Polymorphic: changes shape to hinder classification.

Usable: integrated in Tor & moderate overhead.

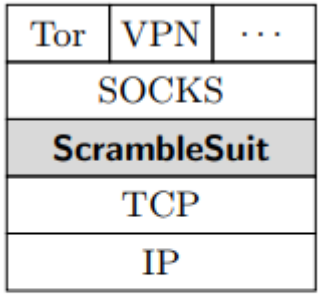


Figure 1: ScrambleSuit's protocol stack.

Philipp Winter, Tobias Pulls, and Juergen Fuss
ScrambleSuit: A Polymorphic Network Protocol to Circumvent Censorship (WPES13)

ScrambleSuit

Pseudo-random payload: ScrambleSuit computationally indistinguishable from randomness. i.e., no DPI fingerprints.

Polymorphic: changes shape to hinder classification.

Usable: integrated in Tor & moderate overhead.

Defense against active probing:

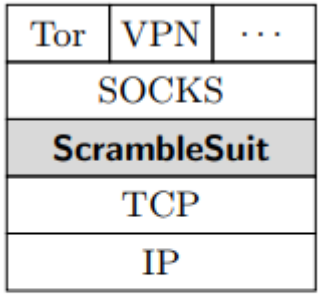


Figure 1: ScrambleSuit's protocol stack.

Philipp Winter, Tobias Pulls, and Juergen Fuss
ScrambleSuit: A Polymorphic Network Protocol to Circumvent Censorship (WPES13)

ScrambleSuit

Pseudo-random payload: ScrambleSuit computationally indistinguishable from randomness. i.e., no DPI fingerprints.

Polymorphic: changes shape to hinder classification.

Usable: integrated in Tor & moderate overhead.

Defense against active probing:

Ticket system?

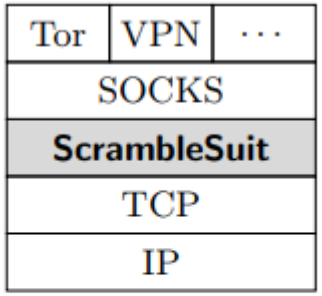


Figure 1: ScrambleSuit's protocol stack.

Philipp Winter, Tobias Pulls, and Juergen Fuss
ScrambleSuit: A Polymorphic Network Protocol to Circumvent Censorship (WPES13)

ScrambleSuit

Pseudo-random payload: ScrambleSuit computationally indistinguishable from randomness. i.e., no DPI fingerprints.

Polymorphic: changes shape to hinder classification.

Usable: integrated in Tor & moderate overhead.

Defense against active probing: use of a secret which is shared between client and server and exchanged out-of-band.

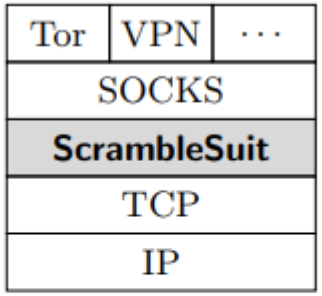
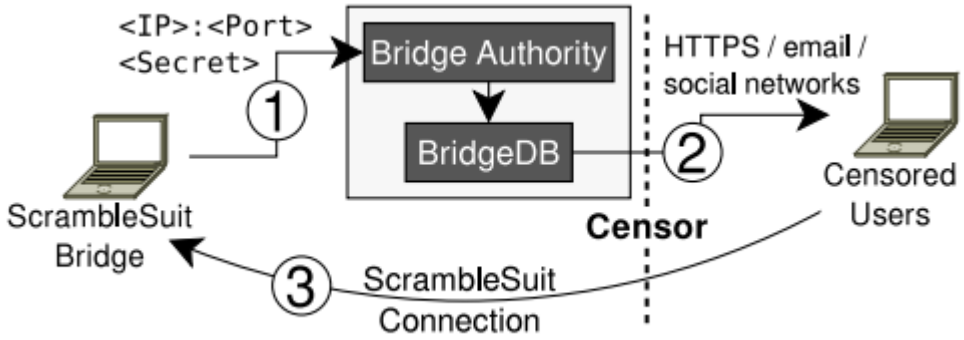


Figure 1: ScrambleSuit's protocol stack.

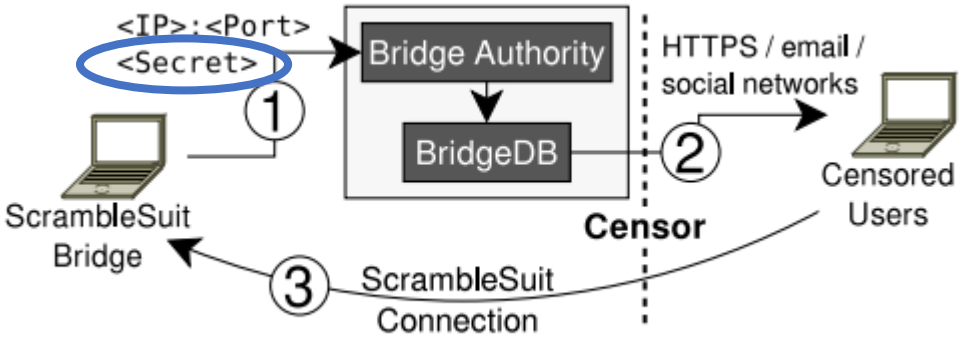
Philipp Winter, Tobias Pulls, and Juergen Fuss
ScrambleSuit: A Polymorphic Network Protocol to Circumvent Censorship (WPES13)

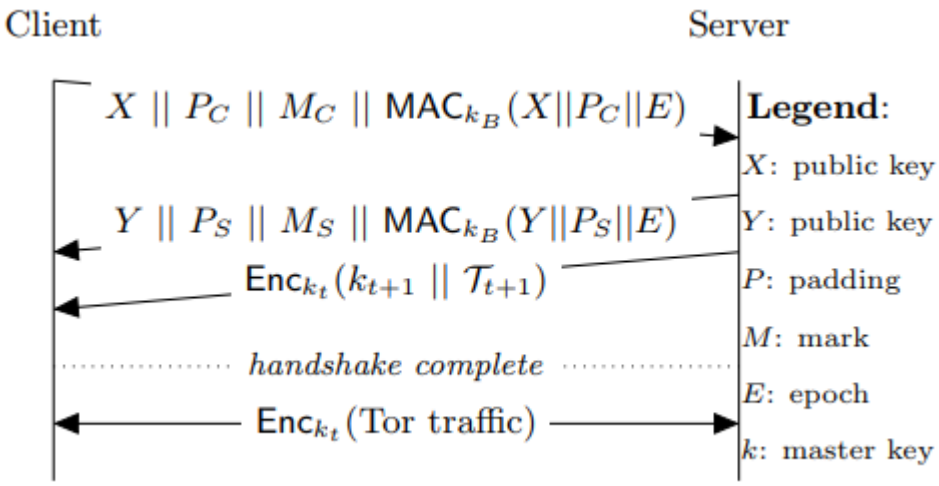
ScrambleSuit: defending against active probing



Philipp Winter, Tobias Pulls, and Juergen Fuss
ScrambleSuit: A Polymorphic Network Protocol to Circumvent Censorship (WPES13)

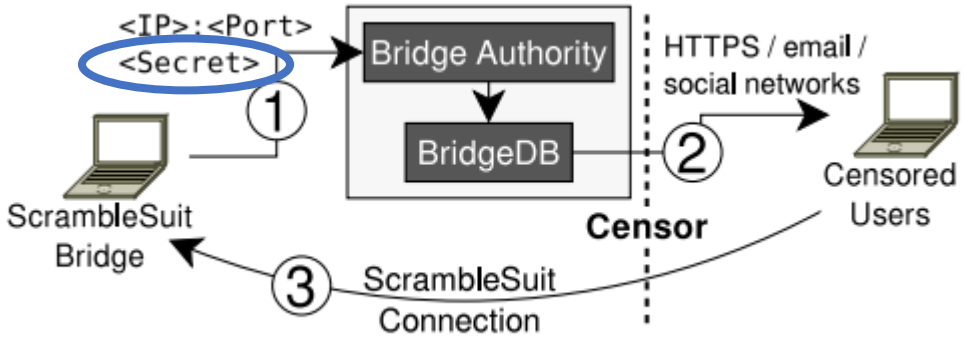
ScrambleSuit

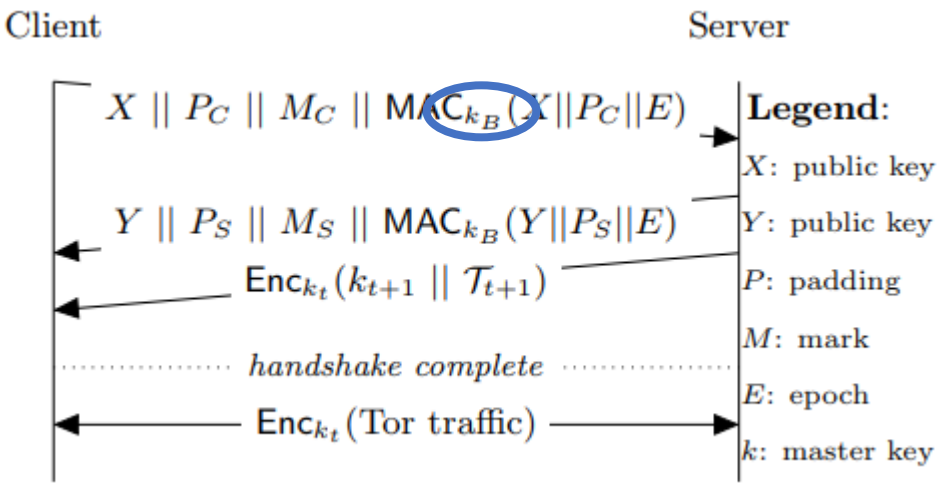




Philipp Winter, Tobias Pulls, and Juergen Fuss
ScrambleSuit: A Polymorphic Network Protocol to Circumvent Censorship (WPES13)

ScrambleSuit





Philipp Winter, Tobias Pulls, and Juergen Fuss
ScrambleSuit: A Polymorphic Network Protocol to Circumvent Censorship (WPES13)

ScrambleSuit: Shaping

Shaping approach:

PROTOCOL POLYMORPHISM: one protocol shape for every server

PACKET LENGTHS and INTER-ARRIVAL TIMES

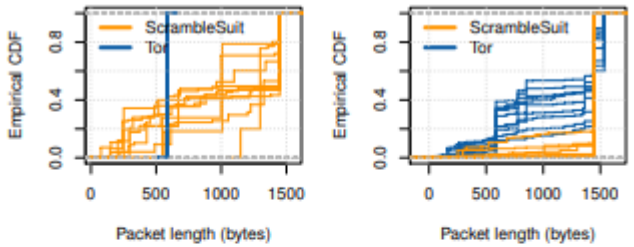
ON BOOTSTRAPPING:

generates a 256-bit seed to obtain two discrete probability distributions
seed transmitted to clients so that it is two-way

Philipp Winter, Tobias Pulls, and Juergen Fuss
ScrambleSuit: A Polymorphic Network Protocol to Circumvent Censorship (WPES13)

Finding the flow: ScrambleSuit

It is difficult to evaluate the effectiveness of our obfuscation techniques since ScrambleSuit does not have a cover protocol to mimic. Otherwise, our evaluation would simply investigate the similarity between our protocol and its cover protocol. Instead of measuring ScrambleSuit's closeness to a mimicked protocol, we measure the deviation from its transported application, i.e., Tor. **Intuitively, higher deviation would imply better obfuscation.**



(a) Client-to-server. (b) Server-to-client.

(c) Client-to-server. (d) Server-to-client.

Philipp Winter, Tobias Pulls, and Juergen Fuss
ScrambleSuit: A Polymorphic Network Protocol to Circumvent Censorship (WPES13)

Tor Pluggable Transports

- **obfs4**
  - **Description:** Is a transport with the same features as **ScrambleSuit** but utilizing Dan Bernstein's elligator2 technique for public key obfuscation, and the ntor protocol for one-way authentication. This results in a faster protocol.
  - **Language:** Go
  - **Maintainer:** Yawning Angel
  - **Evaluation:** obfs4 Evaluation

- **meek**
  - **Description:** Is a transport that uses HTTP for carrying bytes and TLS for obfuscation. Traffic is relayed through a third-party server (Google App Engine). It uses a trick to talk to the third party so that it looks like it is talking to an unblocked server.
  - **Language:** Go
  - **Maintainer:** David Fifield
  - **Evaluation:** meek Evaluation

- **Format-Transforming Encryption** (FTE)
  - **Description:** It transforms Tor traffic to arbitrary formats using their language descriptions. See the research paper.
  - **Language:** Python/C++
  - **Maintainer:** Kevin Dyer
  - **Evaluation:** FTE Evaluation

- **ScrambleSuit**
  - **Description:** Is a pluggable transport that protects against follow-up probing attacks and is also capable of changing its network fingerprint (packet length distribution, inter-arrival times, etc.).
  - **Language:** Python
  - **Maintainer:** Philipp Winter
  - **Evaluation:** ScrambleSuit Evaluation

Tor Pluggable Transports

- obfs4
  - **Description:** Is a transport with the same features as ScrambleSuit but utilizing Dan Bernstein's elligator2 technique for public key obfuscation, and the ntor protocol for one-way authentication. This results in a faster protocol.
  - **Language:** Go

Undeployed PTs

These Pluggable Transports exist but are not deployed as part of the Tor Browser.

- n

⇉obfs3

layed

ooks like

- Description: Look-like-nothing pluggable transport (in ⇉obfsproxy)
- Language: Python
- Maintainer: asn
- Evaluation: obfs3 Evaluation

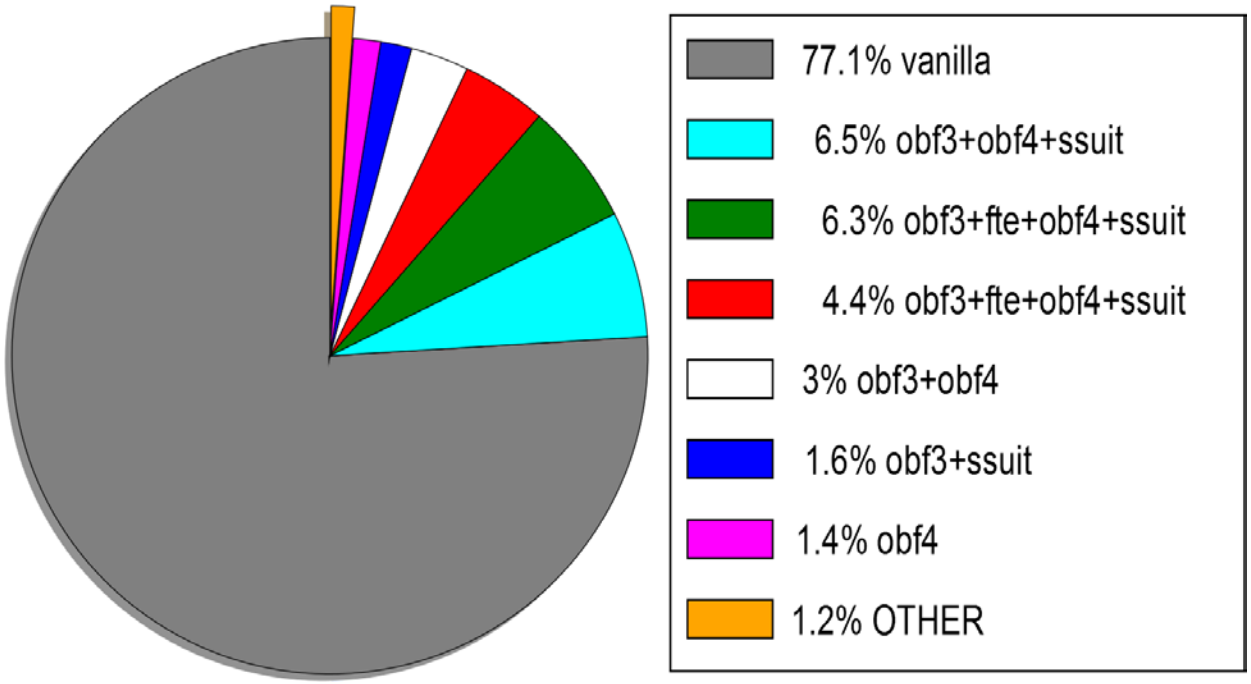
⇉obfs2

- F
  - Description: Look-like-nothing pluggable transport (in ⇉obfsproxy)
  - Language: Python
  - Notes: Superseded by obfs3
  - Maintainer: asn
  - Evaluation: obfs2 Evaluation
  - **Maintainer:** Kevin Dyer
  - **Evaluation:** FTE Evaluation

- ScrambleSuit
  - **Description:** Is a pluggable transport that protects against follow-up probing attacks and is also capable of changing its network fingerprint (packet length distribution, inter-arrival times, etc.).
  - **Language:** Python
  - **Maintainer:** Philipp Winter
  - **Evaluation:** ScrambleSuit Evaluation

PT DEPLOYMENT

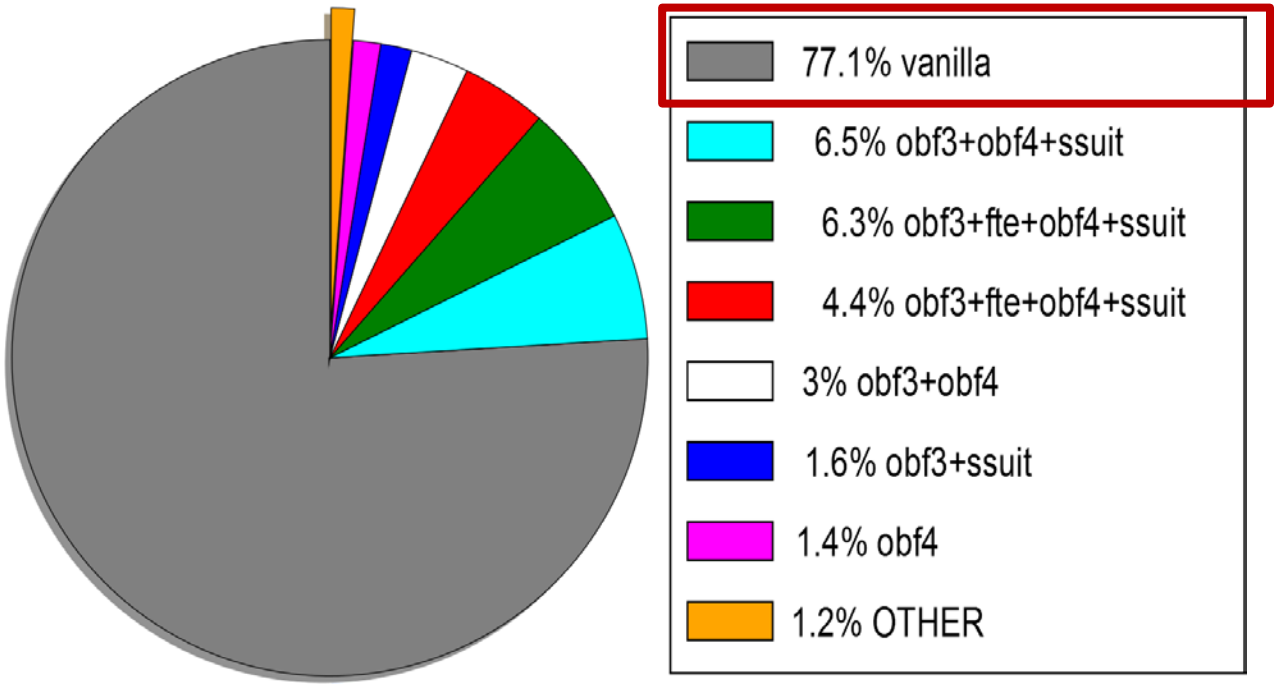
April 2016



Srdjan Matic, Carmela Troncoso, and Juan Caballero
Dissecting Tor Bridges: a Security Evaluation of their Private and Public Infrastructures (NDSS 2017)

PT DEPLOYMENT

April 2016

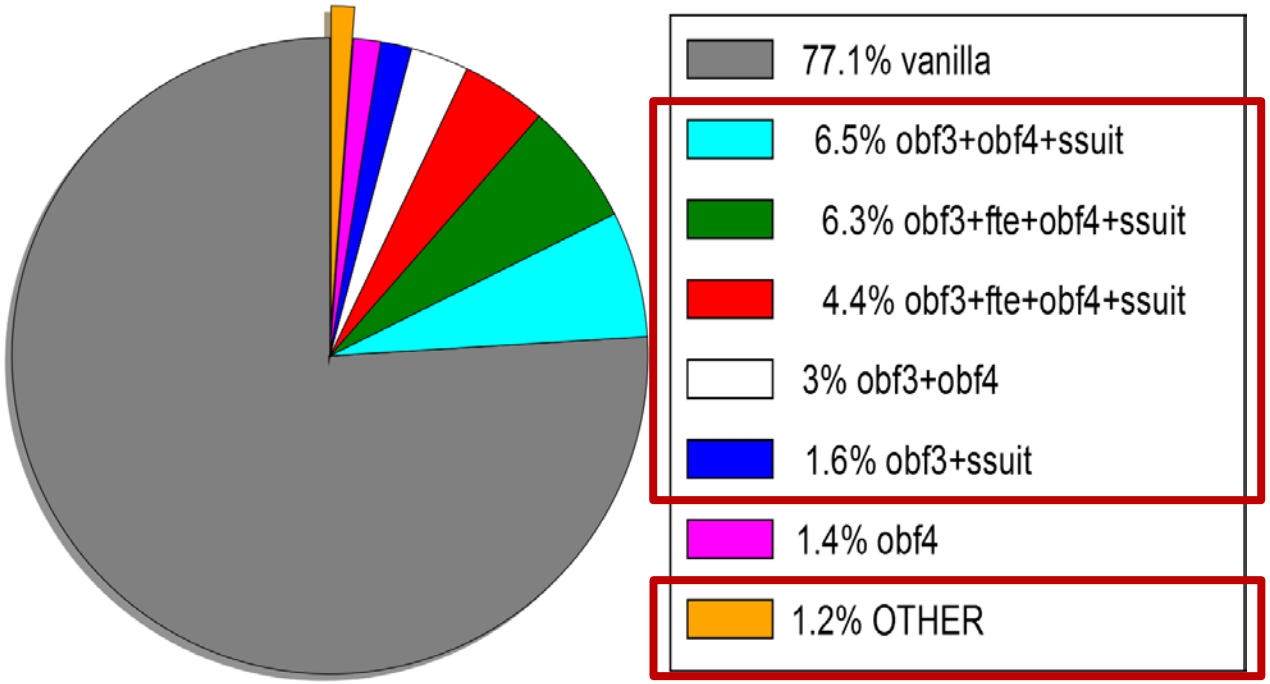


BLOCKABLE!

Srdjan Matic, Carmela Troncoso, and Juan Caballero
Dissecting Tor Bridges: a Security Evaluation of their Private and Public Infrastructures (NDSS 2017)

PT DEPLOYMENT

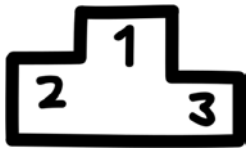
April 2016



CONFLICTING SECURITY PROPERTIES!

Srdjan Matic, Carmela Troncoso, and Juan Caballero
Dissecting Tor Bridges: a Security Evaluation of their Private and Public Infrastructures (NDSS 2017)

Usage of PTs – Ranking

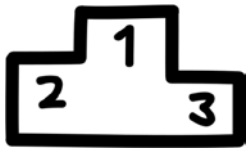


PT	Used Brid.	Clients	Top 20 (Default)	Total Default
vanilla	1,967	14,939	5.6% (0.0%) [0]	1.2% [21]
obfs2	13	158	100.0% (25.8%) [1]	25.8% [1]
obfs3	898	63,088	92.0% (90.8%) [4]	90.8% [4]
obfs4	792	204,095	95.4% (94.7%) [11]	94.7% [11]
ssuit	467	4,483	52.4% (46.3%) [1]	46.3% [1]
meek	4	22,685	100.0% (∼100%) [3]	∼100% [3]

TABLE III. BRIDGE IMPORTANCE PER PT (APR'16).

Srdjan Matic, Carmela Troncoso, and Juan Caballero
Dissecting Tor Bridges: a Security Evaluation of their Private and Public Infrastructures (NDSS 2017)

Usage of PTs – Ranking



PT	Used Brid.	Clients	Top 20 (Default)	Total Default
vanilla	1,967	14,939	5.6% (0.0%) [0]	1.2% [21]
obfs2	13	158	100.0% (25.8%) [1]	25.8% [1]
obfs3	898	63,088	92.0% (90.8%) [4]	90.8% [4]
obfs4	792	204,095	95.4% (94.7%) [11]	94.7% [11]
ssuit	467	4,483	52.4% (46.3%) [1]	46.3% [1]
meek	4	22,685	100.0% (∼100%) [3]	∼100% [3]

TABLE III. BRIDGE IMPORTANCE PER PT (APR'16).

94% obs4 in default!

Useless reply protection...

Srdjan Matic, Carmela Troncoso, and Juan Caballero
Dissecting Tor Bridges: a Security Evaluation of their Private and Public Infrastructures (NDSS 2017)

Takeaways

- Privacy is not only about accuracy, False positives matter
- Semantic attacks may not work as well as thought
- Obfuscating is as hard as mimic
  - Too random is as noticeable as non random
  - ML to learn patterns is very powerful

Identifying the flow: website fingerprinting

Flow properties:

length, inter-arrival times, bursts,

Obfuscation, mimic

IDENTIFYING THE FLOW: WEBSITE FINGERPRINTING

FLOW PROPERTIES:

length, inter-arrival times, bursts,

OBFUSCATION, MIMIC

HIDE A CLASS

IDENTIFYING THE FLOW: WEBSITE FINGERPRINTING

FLOW PROPERTIES:

length, inter-arrival times, bursts,

IDENTIFYING A PARTICULAR FLOW

OBFUSCATION, MIMIC

HIDE A CLASS

Identifying the flow: website fingerprinting

Flow properties:

length, inter-arrival times, bursts,

Obfuscation, mimic

hide a CLASS

Identifying a particular FLOW

Why does it work?

Identifying the flow: website fingerprinting

Flow properties:
length, inter-arrival times, bursts,

Obfuscation, mimic
hide a CLASS

Identifying a particular FLOW

Why does it work?

Next week

Peek-a-Boo, I Still See You:
Why Efficient Traffic Analysis Countermeasures Fail
Dyer, Coull, Ristenpart, and Shrimpton.