An ISP-Scale Deployment of TapDance

Sergey Frolov
Eric Wustrow
University of Colorado Boulder

Fred Douglas
Google

Will Scott
Allison McDonald
Benjamin VanderSloot
University of Michigan

Rod Hynes
Adam Kruger
Psiphon

Michalis Kallitsis
Merit Network

David G. Robinson
Upturn

Steve Schultze
Georgetown University Law Center

Nikita Borisov
University of Illinois

J. Alex Halderman
University of Michigan

ABSTRACT

In this talk, we will report initial results from the world's first ISP-scale field trial of a refraction networking system. Refraction networking is a next-generation censorship circumvention approach that locates proxy functionality in the middle of the network, at participating ISPs or other network operators. We built a high-performance implementation of the TapDance refraction networking scheme and deployed it on four ISP uplinks with an aggregate bandwidth of 100 Gbps. Over one week of operation, our deployment served more than 50,000 real users. The experience demonstrates that TapDance can be practically realized at ISP scale with good performance and at a reasonable cost, potentially paving the way for long-term, large-scale deployments of TapDance or other refraction networking schemes in the future. We will close by discussing interactions between refraction networking and emerging web standards.

1 INTRODUCTION

Censorship circumvention tools typically operate by connecting users to a proxy server located outside the censoring country [2, 11, 14, 17]. Although existing tools use a variety of techniques to conceal the locations of their proxies [3, 7, 12, 16, 18], governments are deploying increasingly sophisticated and effective means to discover and block the proxies [5, 6, 19].

Refraction networking [15][1] is a next-generation circumvention approach with the potential to escape from this cat-and-mouse game. Rather than running proxies at specific edge-hosts and attempting to hide them from censors, refraction works via Internet service providers (ISPs) or other network operators, who provide censorship circumvention functionality for any connection that *passes through* their networks. To accomplish this, clients make HTTPS connections to sites that they can reach, where such connections traverse a participating network. The participating network operator recognizes a steganographic signal from the client and appends the user's requested data to the encrypted connection response. From the perspective of the censor, these connections are indistinguishable from normal TLS connections to sites the censor has not blocked. To block the refraction connections, the censor would need to block all connections that traverse a participating network. The more ISPs participate in such a system, the greater the extent of collateral damage that would-be censors would suffer by blocking the refracted connections.

A variety of refraction networking systems have been proposed in recent years [1, 4, 9, 10, 20, 21], representing different trade-offs among practicality, stealthiness, and performance. The basic idea is to watch all of the traffic passing through a router, selecting flows which are steganographically tagged as participating in the protocol, and then modifying that traffic by extracting and making the encapsulated request on behalf of the client. While each of these schemes has been prototyped in the lab, implementing refraction within a real ISP poses significant additional challenges. An ISP-scale deployment must be able to:

- Identify client connections on high-speed backbone links operating at 10–40 Gbps or more. This is at the limits of commodity network hardware.
- Be built within reasonable cost constraints, in terms both of required hardware and of necessary rack space at crowded Internet exchange points.
- Operate reliably without disrupting the ISP's network or the reachable sites clients connect to.

[1] Previous works used the term *decoy routing*, which confusingly shares the name of a specific refraction scheme. We use refraction networking as an umbrella term to refer to all schemes.

- Have a mechanism for identifying reachable sites for which connections pass through the ISP, and for disseminating this information to clients.
- Coordinate traffic across multiple Internet uplinks or even multiple ISPs.

To demonstrate that these challenges can be solved, we constructed a large trial deployment of the TapDance refraction scheme [20] and operated a trial deployment in partnership with two mid-sized network operators: a regional ISP and a large university. Our goal was to understand: (*i*) the scale of traffic a refraction system built within reasonable constraints today can realistically process, (*ii*) the experience for users of refraction in contrast to traditional proxy-based circumvention, and (*iii*) the impact on ISPs of operating refraction infrastructure.

This talk will present initial results from that deployment, first reported in our FOCI'17 paper [8]. We will discuss the design and engineering considerations that we dealt with in its construction, and we will present data supporting the real-world practicality of refraction at ISP scale. We will close by discussing potential future interactions between refraction networking and emerging web standards such as HTTP2, TLS1.3, and QUIC.

2 DEPLOYMENT DETAILS

We partnered with two network operators: Merit Network, a medium-sized regional ISP and University of Colorado Boulder, a large public university. We worked with each to deploy TapDance stations in a configuration that would have visibility into most of the traffic entering and exiting their respective autonomous systems. In all, we deployed four stations, with three at Merit and one at the University of Colorado.

Bandwidth and traffic volume varied by station loca- tion, with two of the stations (both at Merit) operating on 40 Gbps links and the other two on 10 Gbps links. Space constraints at the peering locations limited each TapDance station to a single commodity 1U server.

To efficiently process packets at 40 Gbps line rates, our implementation is built on the PF_RING library and kernel driver [13], operating in zero-copy mode. By splitting incoming traffic onto multiple cores we were able to handle full line rate traffic with only occasional dropped packets, which, due to the design of TapDance, do not interfere with the normal operation of an ISP.

Our four stations ran on a total of 34 cores (excluding a dedicated PF_RING core per station), with the most loaded station using 14 cores. These 34 cores were able to comfortably handle a peak of close to 14,000 new TLS connections per second, with each connection being checked for a TapDance-tagged request. Our experience demonstrates that, even in large installations, a software-based implementation of TapDance can be practical, avoiding the need for costly specialized hardware.

3 USER TRAFFIC

Over the trial, we served over 50,000 unique users, ac- cording to Psiphon statistics. At peak, TapDance served over 4,000 users simultaneously, with peaks on a single station over 3,000 concurrent users.

During the trial, we also measured the impact of multiple clients using the same site to reach TapDance. Over the approximately 450 hosts available, the median load remained generally evenly spread, with the typical site seeing around 5 clients connected simultaneously, with only 10% of sites ever having more than 20 simultaneous users.

REFERENCES

[1] Cecylia Bocovich and Ian Goldberg. 2016. Slitheen: Perfectly imitated decoy routing through traffic replacement. In *23rd ACM Conference on Computer and Communications Security (CCS)*. 1702–1714.

[2] Roger Dingledine, Nick Mathewson, and Paul Syverson. 2004. Tor: The Second-Generation Onion Router. In *13th USENIX Security Symposium*. 303–320.

[3] K. P. Dyer, S. E. Coull, T. Ristenpart, and T Shrimpton. 2013. Protocol misidentification made easy with format-transforming encryption. In *20th ACM Conference on Computer and Communications Security (CCS)*. 61–72.

[4] Daniel Ellard, Alden Jackson, Christine Jones, Victoria Manfredi, W. Timothy Strayer, Bishal Thapa, and Megan Van Welie. 2015. Rebound: Decoy routing on asymmetric routes via error messages. In *40th IEEE Conference on Local Computer Networks (LCN)*. 91–99.

[5] R. Ensafi, D. Fifield, P. Winter, N. Feamster, N. Weaver, and V. Paxson. 2015. Examining How the Great Firewall Discovers Hidden Circumvention Servers. In *15th ACM Internet Measurement Conference (IMC)*. 445–458.

[6] R. Ensafi, P. Winter, M. Abdullah, and J. Crandall. 2015. Analyzing the Great Firewall of China Over Space and Time. In *Proceedings on Privacy Enhancing Technologies (PETS)*. 61–76.

[7] D. Fifield, C. Lan, R. Hynes, P. Wegmann, and V. Paxson. 2015. Blocking-resistant communication through domain fronting. In *Proceedings on Privacy Enhancing Technologies (PETS)*. 1–19.

[8] Sergey Frolov, Fred Douglas, Will Scott, Allison McDonald, Benjamin VanderSloot, Rod Hynes, Adam Kruger, Michalis Kallitsis, David G. Robinson, Steve Schultze, Nikita Borisov, Alex Halderman, and Eric Wustrow. 2017. An ISP-Scale Deployment of TapDance. In *7th USENIX Workshop on Free and Open Communications on the Internet (FOCI 17)*. USENIX Association, Vancouver, BC. https://www.usenix.org/conference/foci17/workshop-program/presentation/frolov

[9] Amir Houmansadr, Giang T. K. Nguyen, Matthew Caesar, and Nikita Borisov. 2011. Cirripede: Circumvention infrastructure using router redirection with plausible deniability. In *18th ACM Conference on Computer and Communications Security (CCS)*. 187–200.

[10] Josh Karlin, Daniel Ellard, Alden W. Jackson, Christine E. Jones, Greg Lauer, David P. Mankins, and W. Timothy Strayer. 2011. Decoy Routing: Toward Unblockable Internet Communication. In *1st USENIX Workshop on Free and Open Communications on the Internet (FOCI)*.

[11] lantern [n. d.]. Lantern. ([n. d.]). https://getlantern.org/.

[12] Moghaddam H. Mohajeri, B. Li, M. Derakhshani, and I Goldberg. 2012. Skypemorph: Protocol obfuscation for Tor bridges. In *19th ACM Conference on Computer and Communications Security (CCS)*. 97–108.

[13] Ntop. [n. d.]. PF_RING. http://www.ntop.org/products/pf_ring. ([n. d.]).

[14] psiphon [n. d.]. Psiphon. ([n. d.]). https://psiphon.ca.

[15] refraction routing internet [n. d.]. Refraction Networking: Internet freedom in the network's core. ([n. d.]). https://refraction.network/.

[16] The Tor Project. [n. d.]. obfs4 (The obfourscator) specification. ([n. d.]). https://gitweb.torproject.org/pluggable-transports/obfs4.git/tree/doc/obfs4-spec.txt.

[17] uproxy [n. d.]. uProxy. ([n. d.]). https://www.uproxy.org/.

[18] Zachary Weinberg, Jeffrey Wang, Vinod Yegneswaran, Linda Briesemeister, Steven Cheung, Frank Wang, and Dan Boneh. 2012. StegoTorus: A camouflage proxy for the Tor anonymity system. In *19th ACM Conference on Computer and Communications Security (CCS)*. 109–120.

[19] Tim Wilde. Jan. 7, 2012. Knock knock knockin' on bridges' doors. Tor Blog. (Jan. 7, 2012). https://blog.torproject.org/blog/knock-knock-knockin-bridges-doors.

[20] Eric Wustrow, Colleen M. Swanson, and J. Alex Halderman. 2014. TapDance: End-to-Middle Anticensorship without Flow Blocking. In *23rd USENIX Security Symposium*. 159–174.

[21] Eric Wustrow, Scott Wolchok, Ian Goldberg, and J. Alex Halderman. 2011. Telex: Anticensorship in the Network Infrastructure. In *20th USENIX Security Symposium*.