ODNS: Oblivious DNS

Anne Edmundson, Paul Schmitt, Nick Feamster, Jennifer Rexford, Princeton University
Allison Mankin, Salesforce

OARC 28
9 March 2018
Conventional DNS

- **Client**
- **Recursive**
- **Root Server**
- **TLD Server 1.1.1.1**
- **Name Server 2.2.2.2**
Conventional DNS

Client → www.foo.com ?? → Recursive

Root Server
TLD Server 1.1.1.1
Name Server 2.2.2.2
Conventional DNS
Conventional DNS

Client → www.foo.com ?? → Recursive

Recursive → www.foo.com ?? → com 1.1.1.1

com 1.1.1.1 → www.foo.com ?? → foo.com 2.2.2.2

foo.com 2.2.2.2 → Name Server

Name Server 2.2.2.2

TLD Server 1.1.1.1

Root Server
Conventional DNS

- **Client**: www.foo.com ??
- **Recursive Server**: www.foo.com ??
- **Root Server**: com 1.1.1.1
- **TLD Server**: foo.com 2.2.2.2
- **Name Server**: www.foo.com ??

Example Domain
- **Client**: www.foo.com ??
- **Recursive Server**: www.foo.com ??
- **Root Server**: com 1.1.1.1
- **TLD Server**: foo.com 2.2.2.2
- **Name Server**: www.foo.com ??

Example Domain
Conventional DNS
Threat Model

- **User Data**: a user’s identity, queries, browsing patterns

- **Attacker’s Goal**: learn about user data

- **Attacker’s Capabilities**:
  - monitor traffic
  - gain access to DNS logs
Threat Model

- **User Data**: a user’s identity, queries, browsing patterns

- **Attacker’s Goal**: learn about user data

- **Attacker’s Capabilities**:
  - monitor traffic
  - gain access to DNS logs

*DNS operators could be targets of data requests*
Existing Approaches

- DNS Query Name Minimization
- DNS-Over-TLS
- Quad9
- Onion Services (via Tor)
ODNS Overview

Client → Recursive Resolver → Name Servers
ODNS Overview

Client

Recursive Resolver

ODNS Authoritative

Name Servers

Stub Resolver
ODNS Privacy Overview

Stub encrypts & formats domain with a session key
ODNS authoritative acts as a recursive resolver
Prototype & Testbed

Client

Recursive Resolver

VPS in Atlanta, GA (Authoritative)

Princeton (Stub Resolver)

Name Servers
ODNS Overhead (1)
ODNS Overhead (2)
ODNS Overhead (3)
ODNS Function Overhead

![Bar chart showing ODNS function overhead times in milliseconds for different operations.]
Conclusion

- Designed ODNS to protect privacy by decoupling clients’ identities with their requests
- Implemented and evaluated a prototype of the design
- Future
  - Implementation based on Knot
  - Explore authoritative replication
  - Examine impact on caching
Questions?

pschmitt@cs.princeton.edu