DNS - Devious Name Services

Destroying Privacy & Anonymity Without Your Consent

Jim Nitterauer

Senior Security Specialist

Disclaimer

Information disclosed in this presentation is intended to help improve your security & privacy posture and should not be used for unethical purposes

The concepts presented are in no way meant to imply original research on my part or on the part of my employer

Information presented here is gathered from public and private sources with proper references and credit provided where applicable

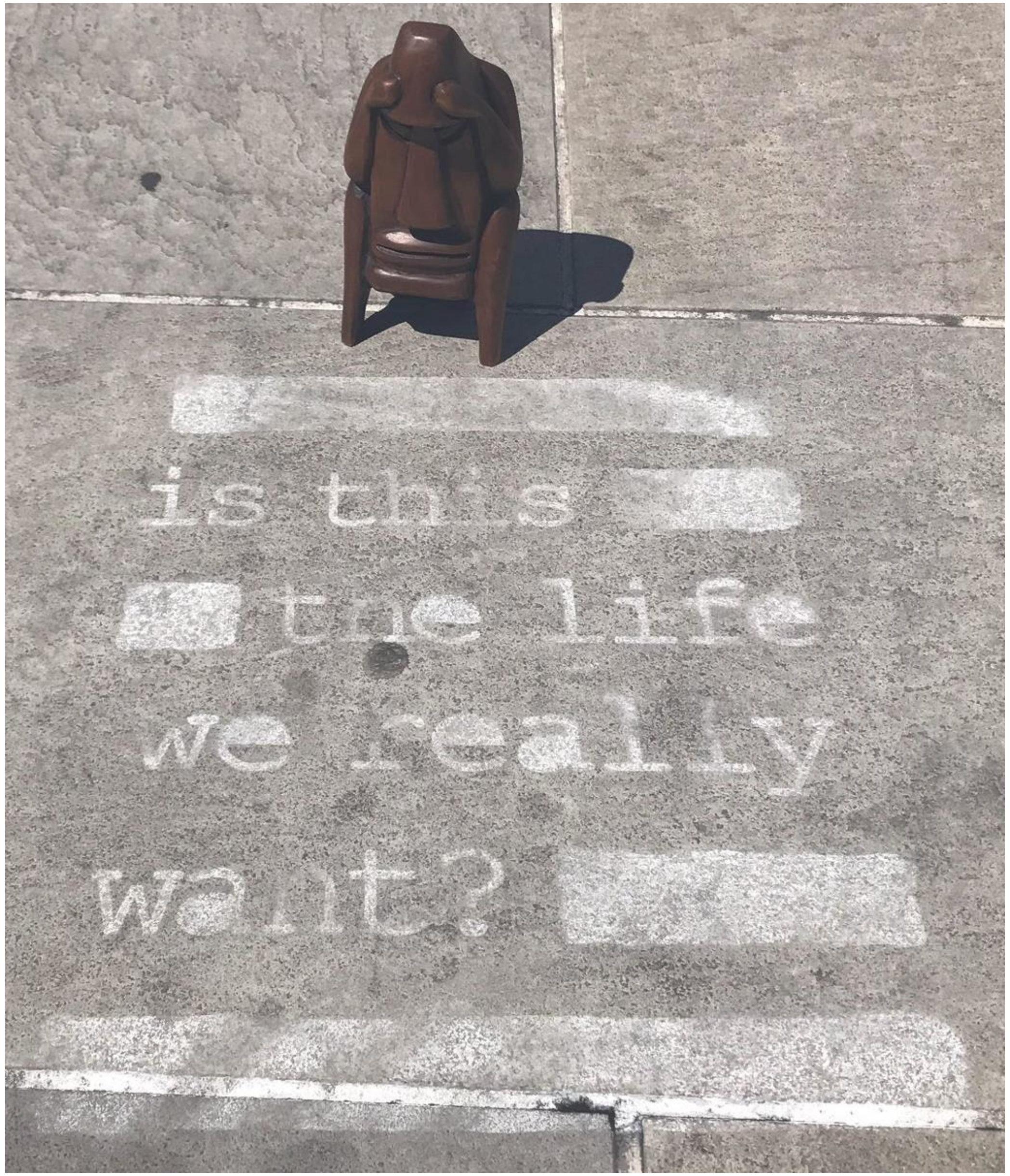
The views expressed in this talk are not necessarily the views of my employer

Why Am I here?

VPN Blocked in China:
The Great Firewall Strikes Again
Best VPN Provider
Get the facts, reviews and tips

http://bestvpnprovider.co/wp-content/uploads/2015/01/VPN-Blocked-in-China-The-Great-Firewall-Strikes-Again.png

Why Am I here?



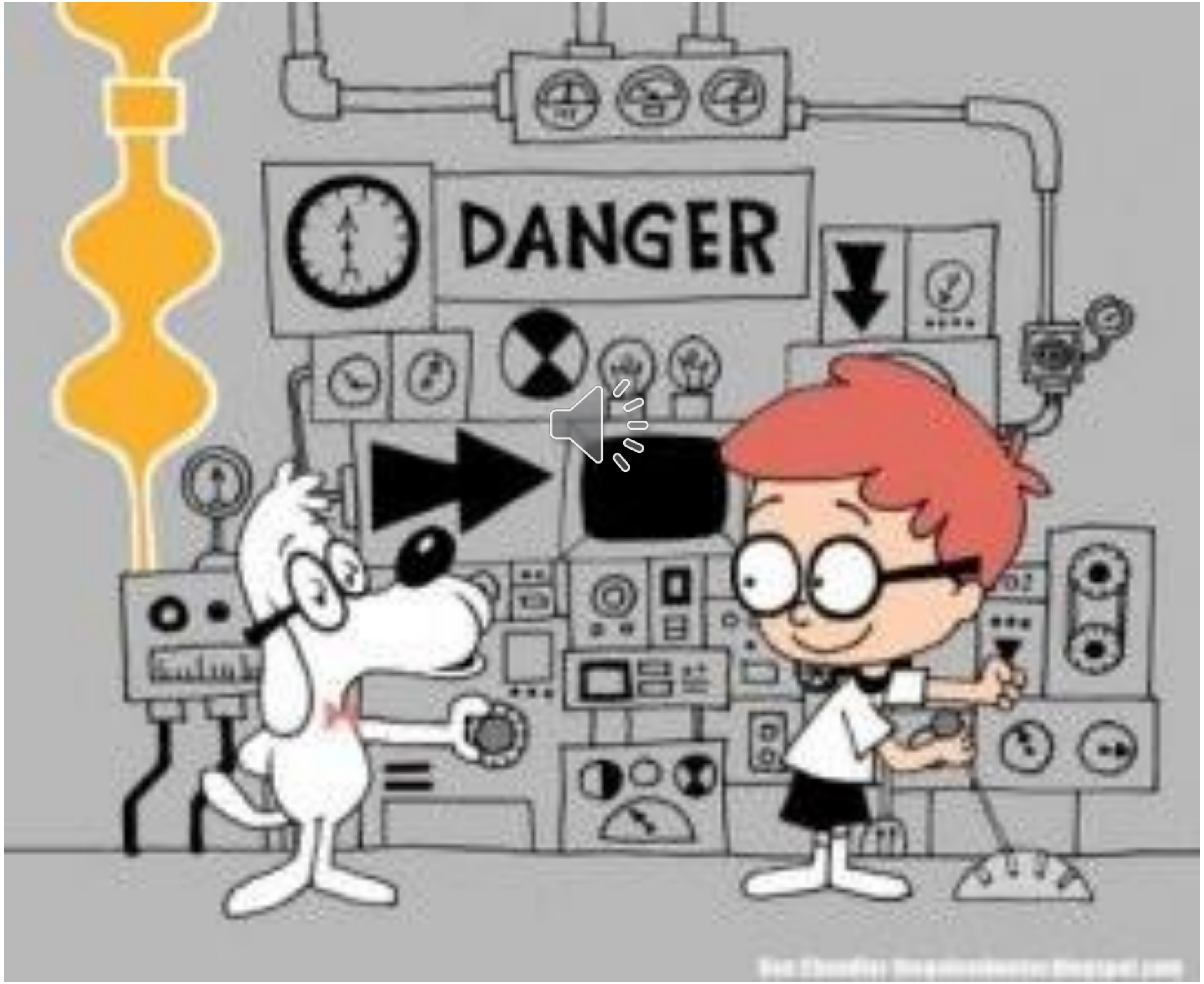
Agenda

- Review DNS, EDNS0 extensions and Option Codes
- Discuss the Rationale for EDNS0 Use
- Examine EDNS Client Subnet (ECS)
- Review DNS Resolver Support
- Examine Tools & Procedures for Testing
- Discuss Privacy Implications of EDNS0 OPT Codes
- Discuss Potential for Abuse
- Questions & answers

Goals for Today

- Understand the basics about EDNS OPT RRs
- Understand the potential threat to your privacy
- Have direction for detecting the use of EDNS OPTs
- Be able to better insure your online privacy

Brief History of DNS



Brief History of DNS

Introduced in 1983 by Paul Mockapetris & Jon Postel

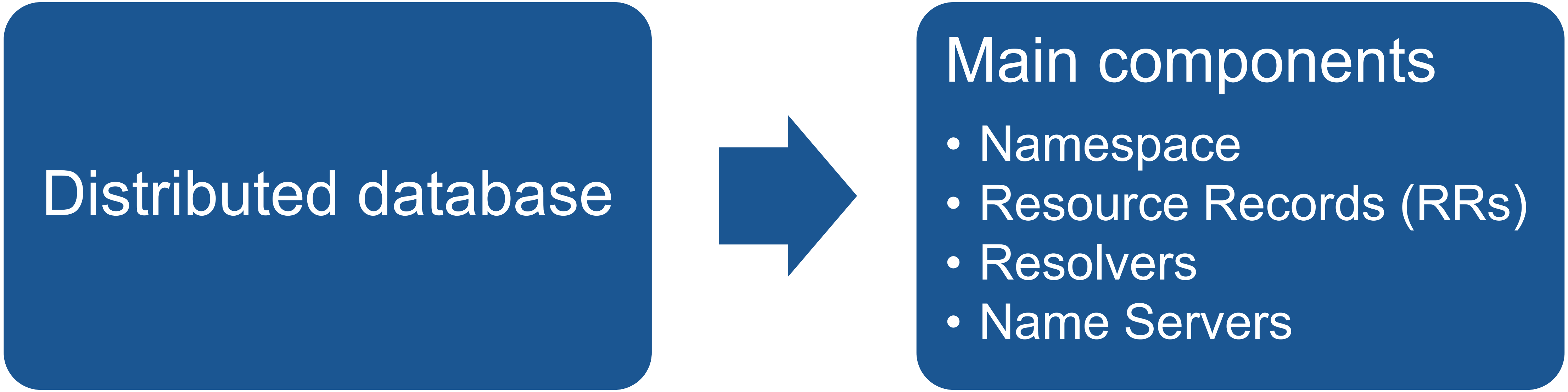
Information Sciences Institute – USC

RFC 882 & RFC 883 both updated by RFC 973 in 1986

Obsoleted by two RFCs in 1987

- RFC 1034 – Describes the data structure and exchange of data
- RFC 1035 – Describes record and infrastructure format

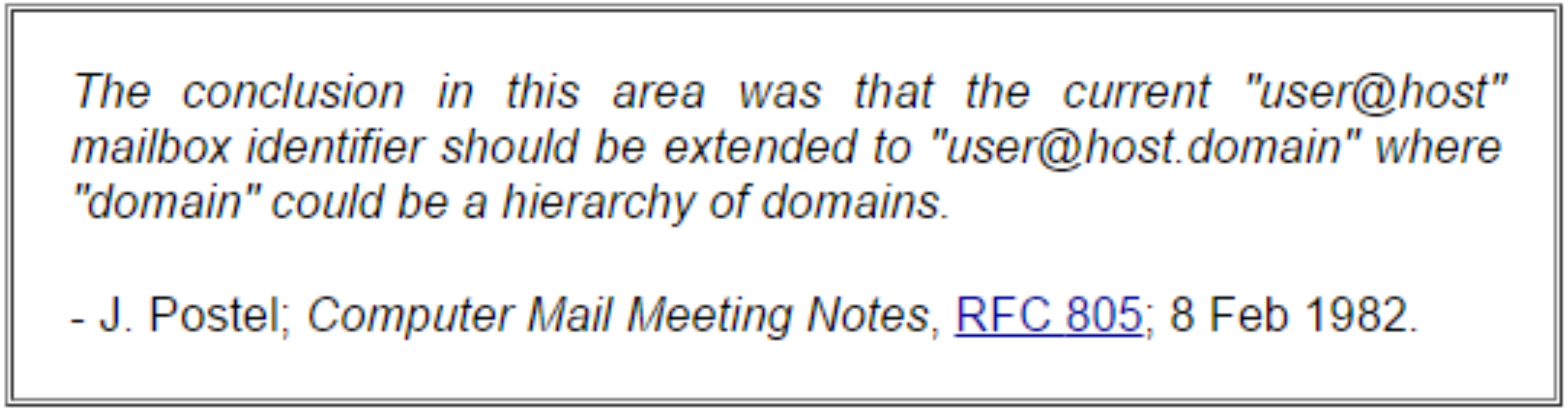
Brief History of DNS



Distributed database

Main components

- Namespace
- Resource Records (RRs)
- Resolvers
- Name Servers



The conclusion in this area was that the current "user@host" mailbox identifier should be extended to "user@host.domain" where "domain" could be a hierarchy of domains.

- J. Postel; *Computer Mail Meeting Notes*, RFC 805; 8 Feb 1982.

Brief History of DNS

Improved DNS by

- Defining Master (Primary) / Slave (Secondary) relationship
- Adding Notify
- Adding IXFR (Incremental Zone Transfers)

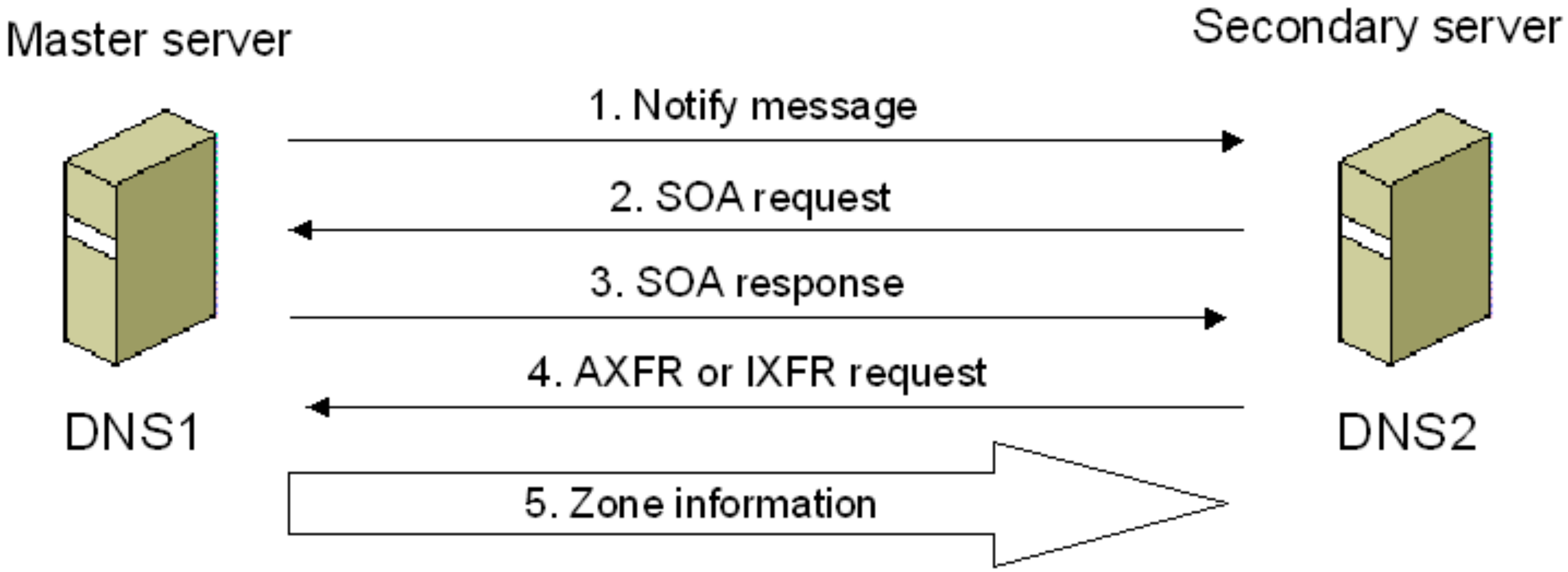


Image from https://technet.microsoft.com/en-us/library/bb962069.tcpipm09_big(l=en-us).gif

Brief History of DNS

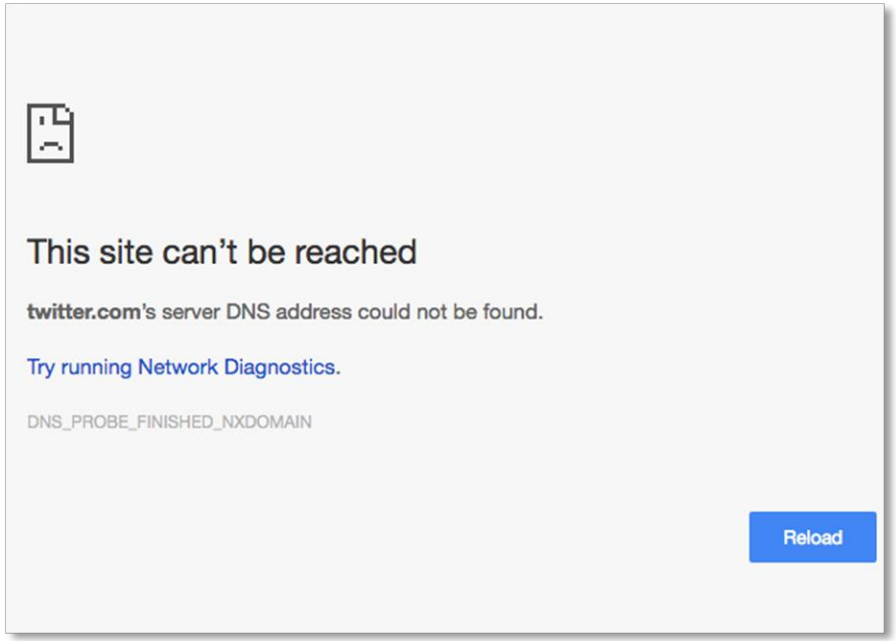
Improved DNS by

- Implementing Dynamic Updates – RFC 2136
- Adding Extension Mechanisms for DNS (EDNS0) - RFC 2671 & RFC 6891

Brief History of DNS

Improved DNS by

- Adding clarifications - RFC 2181
- Implementing provisions for negative responses - RFC 2308



This site can’t be reached

twitter.com’s server DNS address could not be found.

Try running Network Diagnostics.

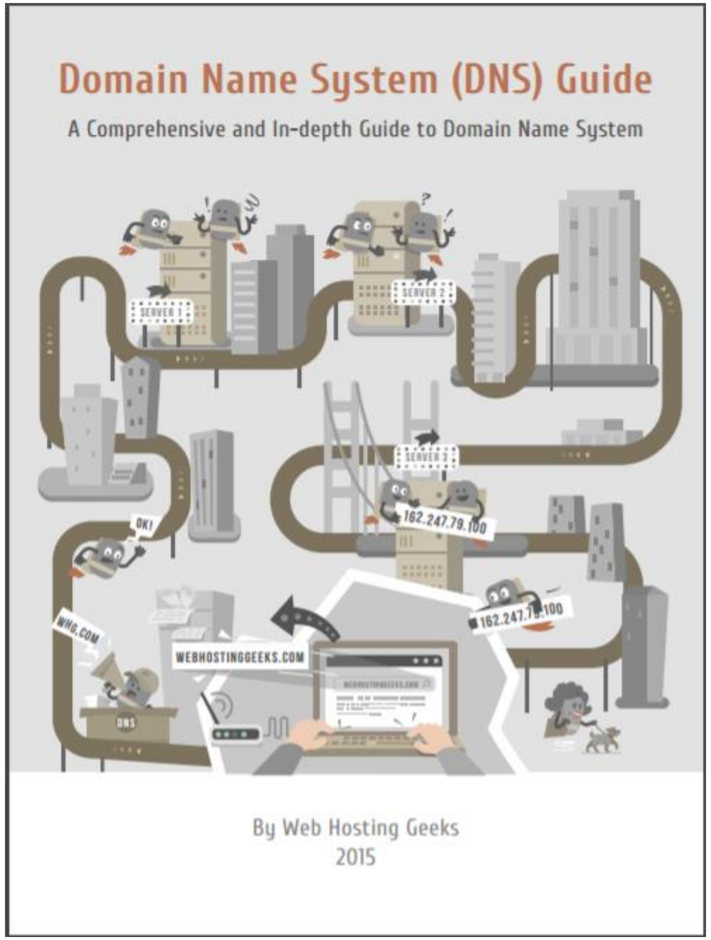
DNS_PROBE_FINISHED_NXDOMAIN

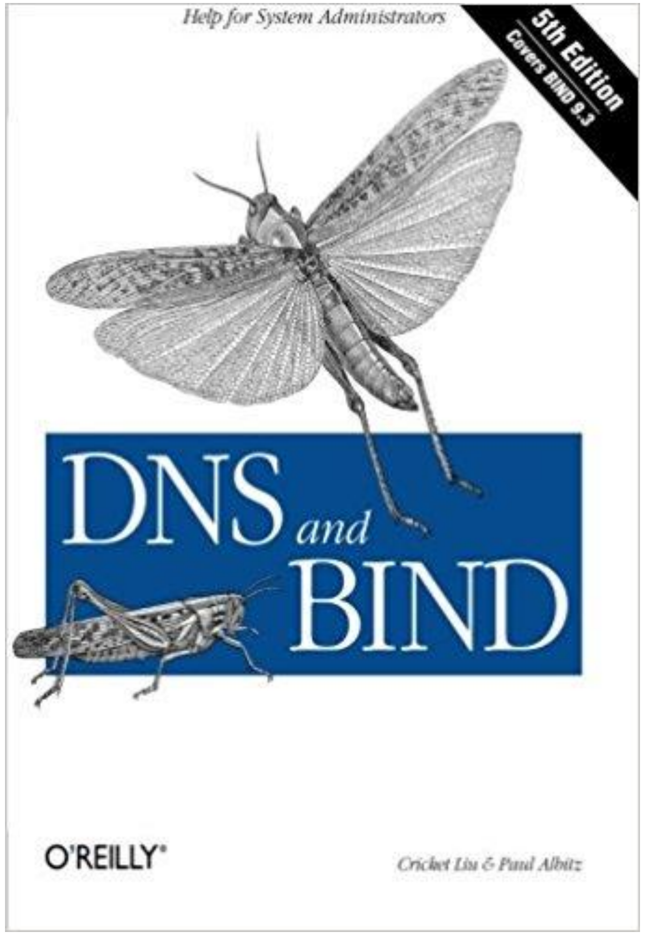
Brief History of DNS

Improved DNS by

- Implementing DNS Security (DNSSEC) - RFC 2535 now RFC 6840
- Promoting the use of EDNS OPT Codes

Brief History of DNS


Domain Name System (DNS) Guide
A Comprehensive and In-depth Guide to Domain Name System
By Web Hosting Geeks
2015



https://webhostinggeeks.com/guides/dns/DNS_221215.pdf

EDNS RFC 2671 & 6891

- RFC 2671 proposed by Paul Vixie in 1999`
- Replaced by RFC 6891 in 2013
- Overcomes 512 byte UDP packet size limit
- Support required for certain modern DNS features
- Defines transport standards
- Defines option format & assignments

EDNS Option Codes

Long list of RFCs and Drafts

- See https://www.iana.org/assignments/dns-parameters/dns-parameters.xhtml#dns-parameters-11

There are approximately 17 codes in use

- 11 RFCs
- 3 Drafts

65,535 possible code assignments

- Future expansion
- Don’t confuse w/ Opcode

EDNS Option Codes

Resource Record Type 41

- Extends RCODE field from 4 to 12 bytes

RFC 6891

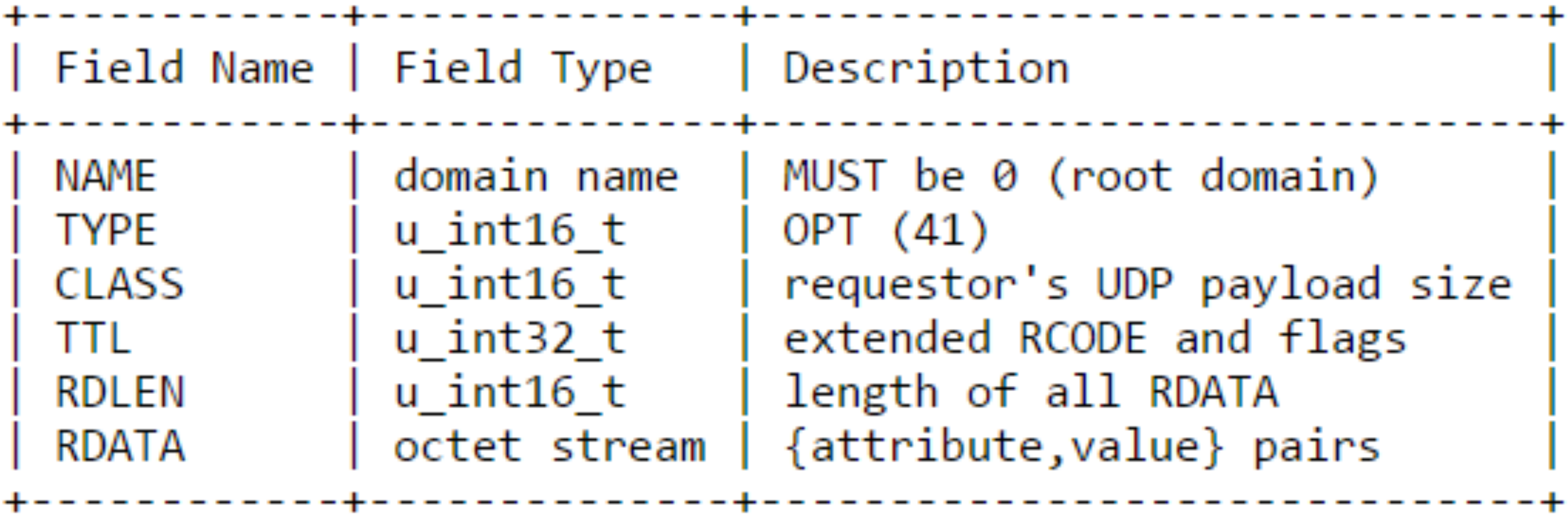
- EDNS(0) Defines OPT Record

RFC 3225

- Defines support for DNSSEC

EDNS Option Codes

OPT Resource Record Format

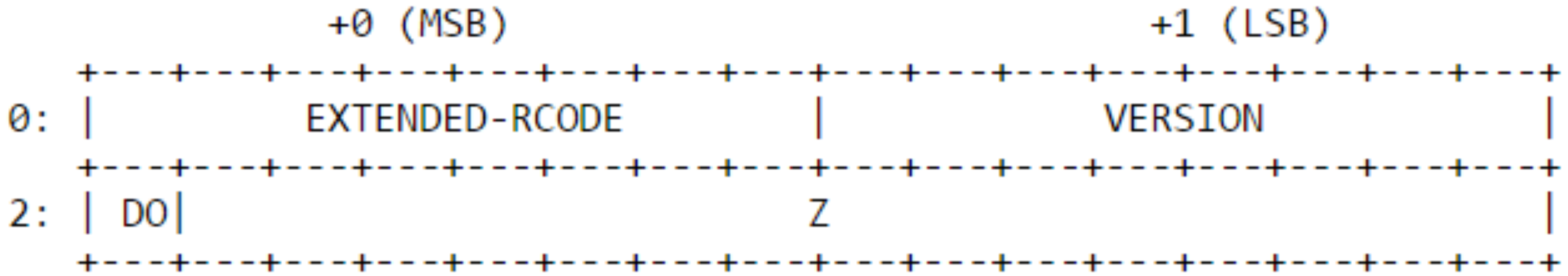


Field Name	Field Type	Description
NAME	domain name	MUST be 0 (root domain)
TYPE	u_int16_t	OPT (41)
CLASS	u_int16_t	requestor's UDP payload size
TTL	u_int32_t	extended RCODE and flags
RDLEN	u_int16_t	length of all RDATA
RDATA	octet stream	{attribute,value} pairs

OPT RR Format

EDNS Option Codes

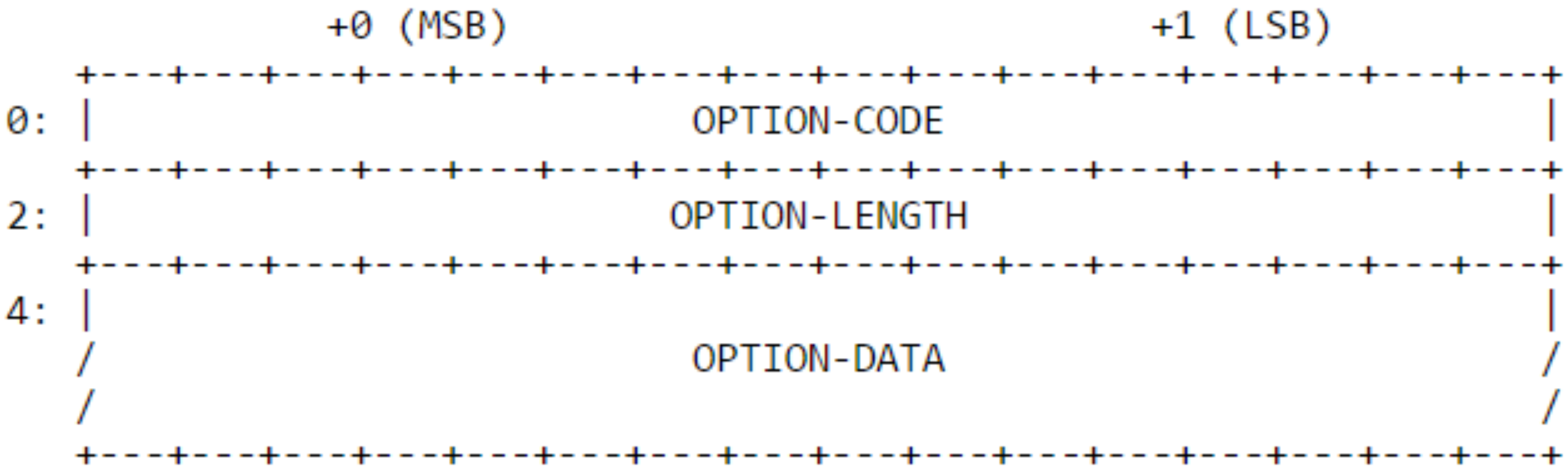
OPT Record TTL Field



```
              +0 (MSB)                            +1 (LSB)
   +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
0: |         EXTENDED-RCODE        |            VERSION            |
   +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
2: | DO|                           Z                               |
   +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

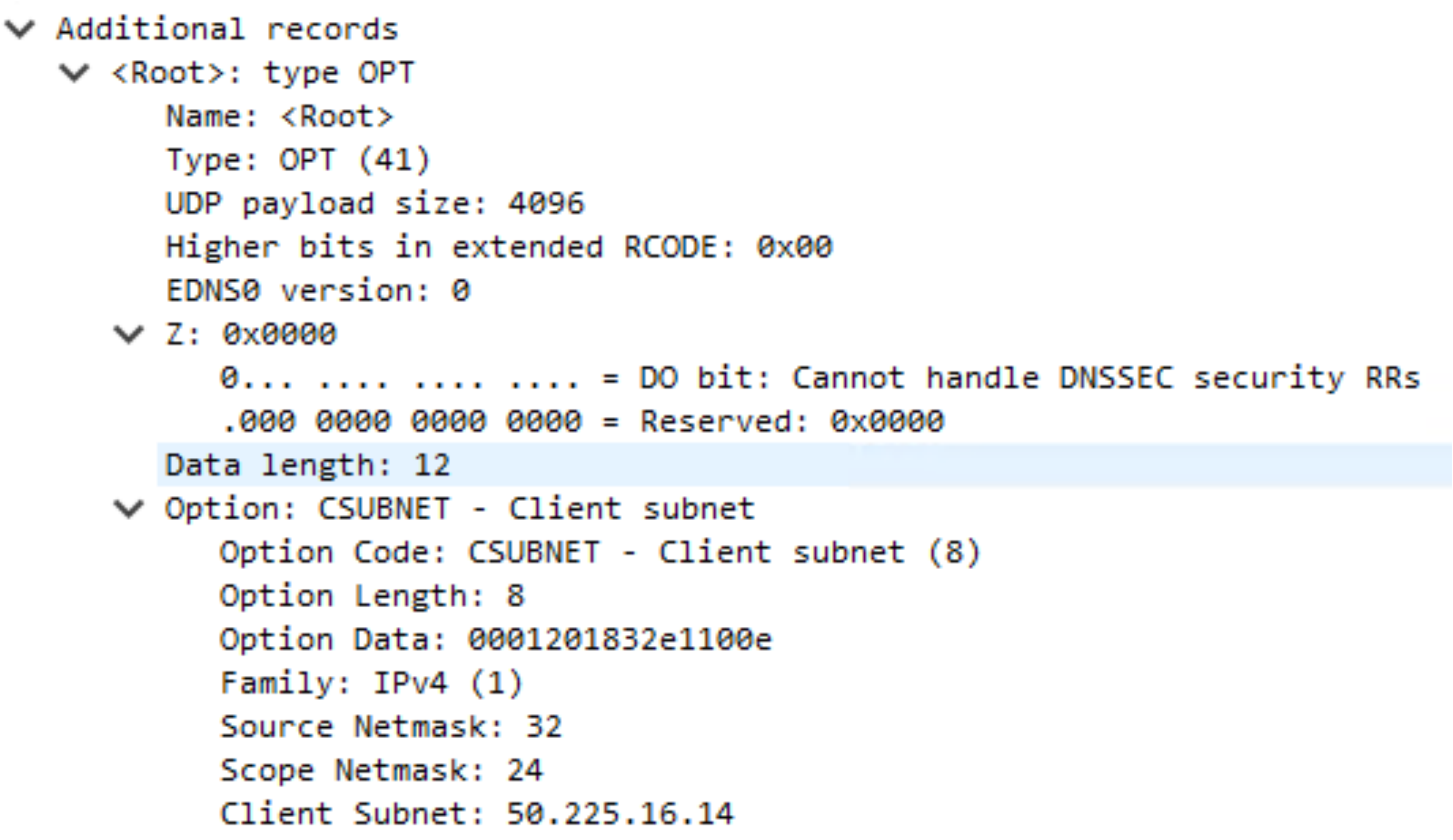
EDNS Option Codes

OPT RR RDATA Structure



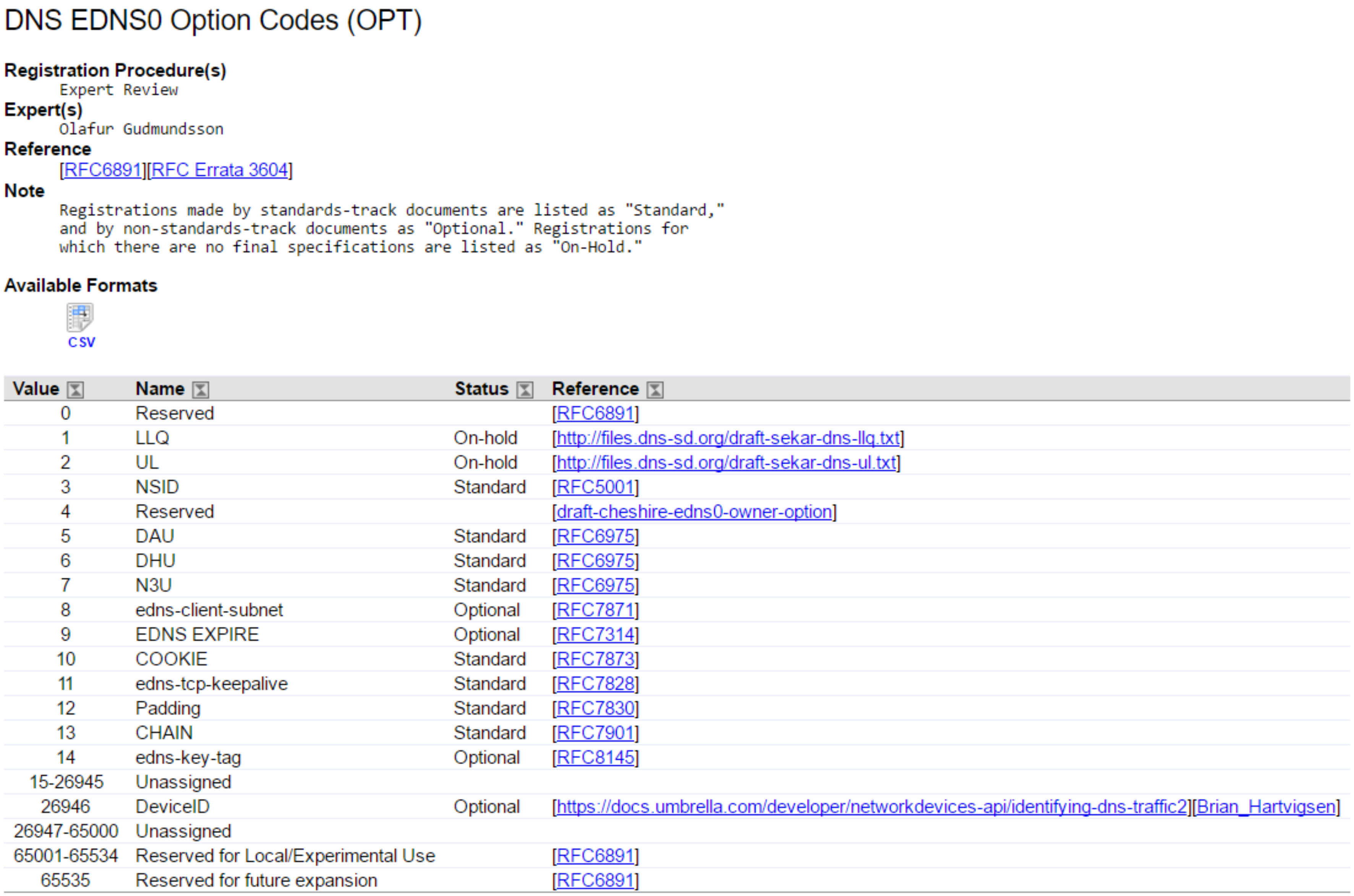
```
              +0 (MSB)                            +1 (LSB)
   +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
0: |                          OPTION-CODE                          |
   +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
2: |                         OPTION-LENGTH                         |
   +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
4: |                                                               |
   /                          OPTION-DATA                          /
   /                                                               /
   +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

EDNS Option Codes



```
⌄ Additional records
  ⌄ <Root>: type OPT
      Name: <Root>
      Type: OPT (41)
      UDP payload size: 4096
      Higher bits in extended RCODE: 0x00
      EDNS0 version: 0
    ⌄ Z: 0x0000
        0... .... .... .... = DO bit: Cannot handle DNSSEC security RRs
        .000 0000 0000 0000 = Reserved: 0x0000
      Data length: 12
    ⌄ Option: CSUBNET - Client subnet
        Option Code: CSUBNET - Client subnet (8)
        Option Length: 8
        Option Data: 0001201832e1100e
        Family: IPv4 (1)
        Source Netmask: 32
        Scope Netmask: 24
        Client Subnet: 50.225.16.14
```

EDNS Option Codes



DNS EDNS0 Option Codes (OPT)

Registration Procedure(s)
Expert Review
Expert(s)
Olafur Gudmundsson
Reference
[RFC6891][RFC Errata 3604]
Note
Registrations made by standards-track documents are listed as "Standard,"
and by non-standards-track documents as "Optional." Registrations for
which there are no final specifications are listed as "On-Hold."

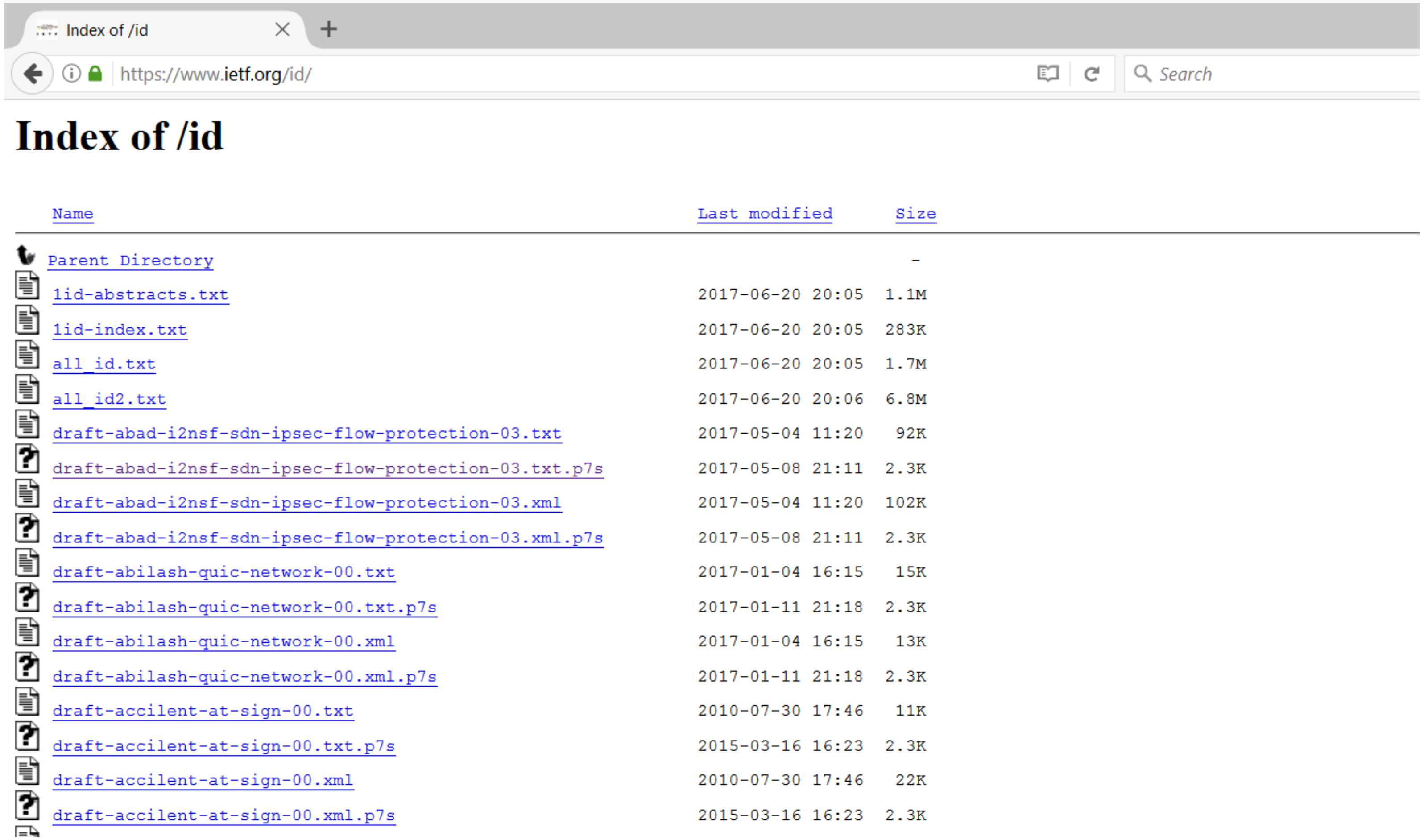
Available Formats

CSV

Value	Name	Status	Reference
0	Reserved		[RFC6891]
1	LLQ	On-hold	[http://files.dns-sd.org/draft-sekar-dns-llq.txt]
2	UL	On-hold	[http://files.dns-sd.org/draft-sekar-dns-ul.txt]
3	NSID	Standard	[RFC5001]
4	Reserved		[draft-cheshire-edns0-owner-option]
5	DAU	Standard	[RFC6975]
6	DHU	Standard	[RFC6975]
7	N3U	Standard	[RFC6975]
8	edns-client-subnet	Optional	[RFC7871]
9	EDNS EXPIRE	Optional	[RFC7314]
10	COOKIE	Standard	[RFC7873]
11	edns-tcp-keepalive	Standard	[RFC7828]
12	Padding	Standard	[RFC7830]
13	CHAIN	Standard	[RFC7901]
14	edns-key-tag	Optional	[RFC8145]
15-26945	Unassigned		
26946	DeviceID	Optional	[https://docs.umbrella.com/developer/networkdevices-api/identifying-dns-traffic2][Brian_Hartvigsen]
26947-65000	Unassigned		
65001-65534	Reserved for Local/Experimental Use		[RFC6891]
65535	Reserved for future expansion		[RFC6891]

https://www.iana.org/assignments/dns-parameters/dns-parameters.xhtml#dns-parameters-11

Additional Drafts



Index of /id

Name	Last modified	Size
Parent Directory		-
1id-abstracts.txt	2017-06-20 20:05	1.1M
1id-index.txt	2017-06-20 20:05	283K
all_id.txt	2017-06-20 20:05	1.7M
all_id2.txt	2017-06-20 20:06	6.8M
draft-abad-i2nsf-sdn-ipsec-flow-protection-03.txt	2017-05-04 11:20	92K
draft-abad-i2nsf-sdn-ipsec-flow-protection-03.txt.p7s	2017-05-08 21:11	2.3K
draft-abad-i2nsf-sdn-ipsec-flow-protection-03.xml	2017-05-04 11:20	102K
draft-abad-i2nsf-sdn-ipsec-flow-protection-03.xml.p7s	2017-05-08 21:11	2.3K
draft-abilash-quic-network-00.txt	2017-01-04 16:15	15K
draft-abilash-quic-network-00.txt.p7s	2017-01-11 21:18	2.3K
draft-abilash-quic-network-00.xml	2017-01-04 16:15	13K
draft-abilash-quic-network-00.xml.p7s	2017-01-11 21:18	2.3K
draft-accilent-at-sign-00.txt	2010-07-30 17:46	11K
draft-accilent-at-sign-00.txt.p7s	2015-03-16 16:23	2.3K
draft-accilent-at-sign-00.xml	2010-07-30 17:46	22K
draft-accilent-at-sign-00.xml.p7s	2015-03-16 16:23	2.3K

https://www.ietf.org/id/

EDNS Option Codes

Review of OPT Option Codes

Code	Name	Status	Description	Vendor
0		Reserved		
1	LLQ	Draft (expired)	DNS Long Lived Queries	Apple
2	UL	Draft (expired)	Dynamic DNS Update Leases	Apple
3	NSID	RFC 5001	DNS Name Server Identifier	ISC
4		Draft - Expired		
5	DAU	RFC 6975	DNSSEC	NIST
6	DHU	RFC 6975	DNSSEC	NIST
7	N3U	RFC 6975	DNSSEC	NIST
8	ECS	RFC 7871	EDNS Client Subnet	Google Akamai
9	EDNS EXP	RFC 7314	SOA Expire Identifier	ISC

EDNS Option Codes

Review of OPT Option Codes

Code	Name	Status	Description	Vendor
10	COOKIE	RFC 7873	DNS Cookies	ISC, Huawei
11	EDNS-TCP	RFC 7828	TCP Keepalive	Red Hat, Dyn, ISC
12	PADDING	RFC 7830	Random Padding	GmbH
13	CHAIN	RFC 7901	CHAIN Query Requests	Red Hat
14	EDNS KEY	RFC 8145	DNSSEC	Verisign, Google, ICANN
26946	DEVICEID	Experimental	Umbrella Device ID	Cisco

Proposed Drafts

UA	ISP LOC	Draft	ISP Location in DNS	CNNIC
UA	CLIENT ID	Draft	Client ID in Forwarded DNS	Charter, Akamai

EDNS Use Cases

OPT Option Codes 5,6 & 7

- All related to DNSSEC implementation
- Let’s resolvers know which cryptographic algorithm was used to generate the digital signature
- Specifies a way for validating end-system resolvers to tell a server in a DNS query which digital signature and/or hash algorithms they support
- OPT 5 – DNSSEC Algorithm Understood (DAU)
- OPT 6 – DS Hash Understood (DHU)
- OPT 7 – NSEC3 Hash Understood (N3U)

EDNS Use Cases

OPT Option Code 8

- Client subnet in DNS queries
- EDNS Client Subnet (ECS)
- Let’s all resolvers know the IPv4 WAN or IPv6 address subnet of the requester
- Developed to enable Content Delivery Networks via DNS
- We will look at bit more into the details shortly

EDNS Use Cases

OPT Option Code 26946

- DeviceID
- Used by Cisco Umbrella (Formerly OpenDNS)
- Sends the following data
  - Organization ID
  - Remote “Internal” IP
  - Remote IPv6
- Built into Umbrella Client or Umbrella enabled gateways
- https://docs.umbrella.com/developer/networkdevices-api/identifying-dns-traffic2

Draft ISP Location

- ISP Location in DNS Queries
- Proposed by China Internet Network Information Center (CNNIC)
- draft-pan-dnsop-edns-isp-location-01
- Claims to be an improvement to privacy
- EIL data includes
  - Country
  - Area
  - ISP

Draft Client ID

- Client ID in Forwarded DNS Queries
- Proposed by Akamai
- draft-tale-dnsop-edns0-clientid-01
- Purpose is to provide more precise client identity
- Ex
  - Parental control
  - Domain access restriction
  - Compromise attribution

EDNS Client Subnet

Initial Draft

- Draft-vandergaast-edns-client-subnet-00

Submitted January 27th, 2011

- C. Contavalli & W. van der Gaast – Google
- S. Leach – Verisign
- D. Rodden – Neustar

Revision 02 submitted on July 4th, 2013

- Note the date
- Ironic that it was changed on Independence Day

Resubmitted May 26th, 2015

- Draft-ietf-dnsop-edns-client-subnet-01
- Added - D. Lawrence – Akamai & W. Kumari – Google

EDNS Client Subnet

Revision 02

- July 6th, 2015

Revision 03

- August 24th, 2015

Revision 04

- September 25th, 2015

Revision 05

- December 14th, 2015

EDNS Client Subnet

Revision 06	• December 15th, 2015
Revision 07	• March 21st, 2016
Revision 08	• April 19th, 2016
RFC 7871	• May, 2016

EDNS Client Subnet

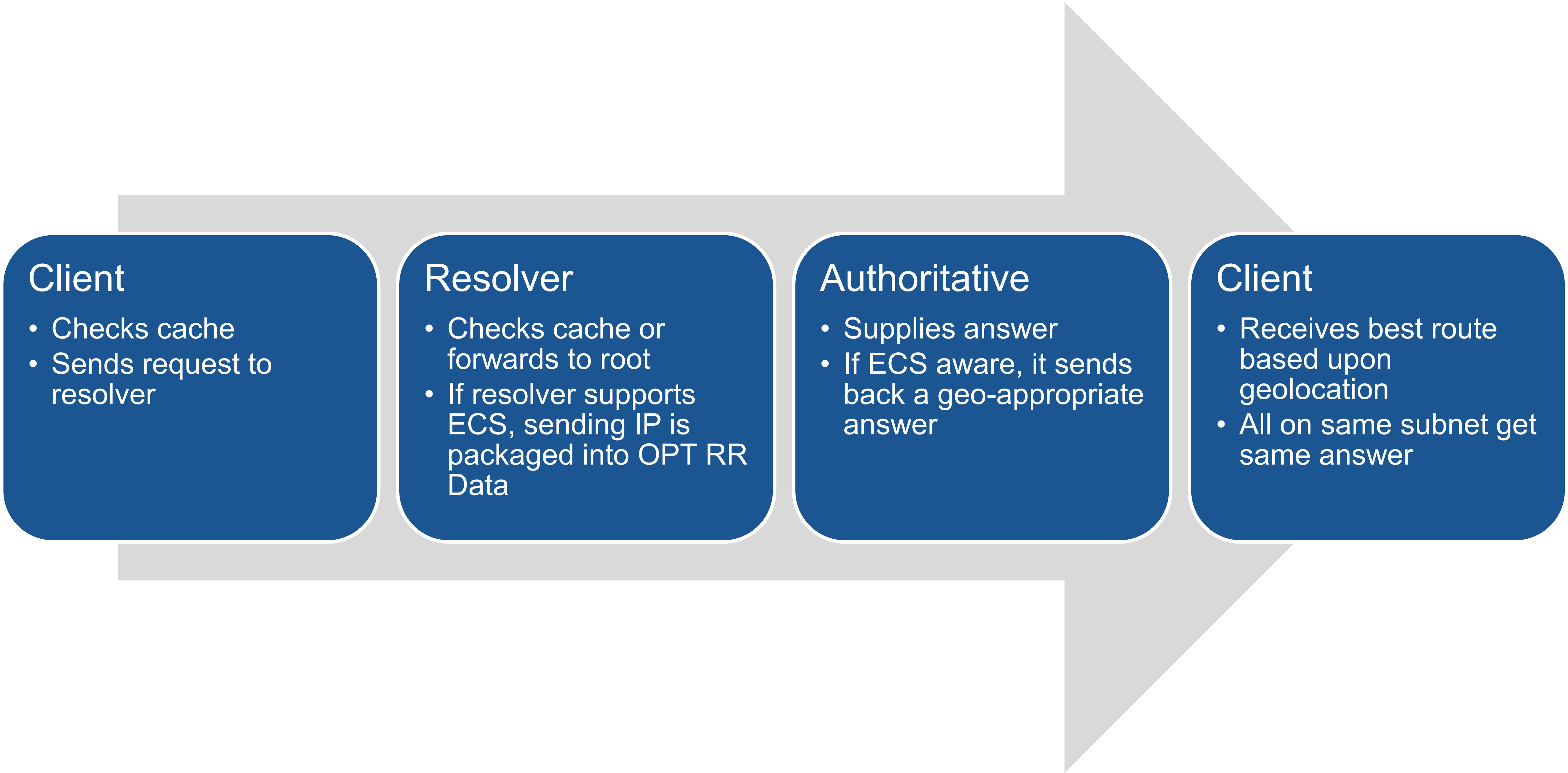
Patent submitted April 30th, 2012

- Number WO2013164007 A1
- Jan Seedorf & Mayutan Arumaithurai - Nec Europe Ltd.

Still shows as Application so not granted

- U.S. Patent number US20150134730 A1
- Interesting precedent

EDNS Client Subnet



Client

- Checks cache
- Sends request to resolver

Resolver

- Checks cache or forwards to root
- If resolver supports ECS, sending IP is packaged into OPT RR Data

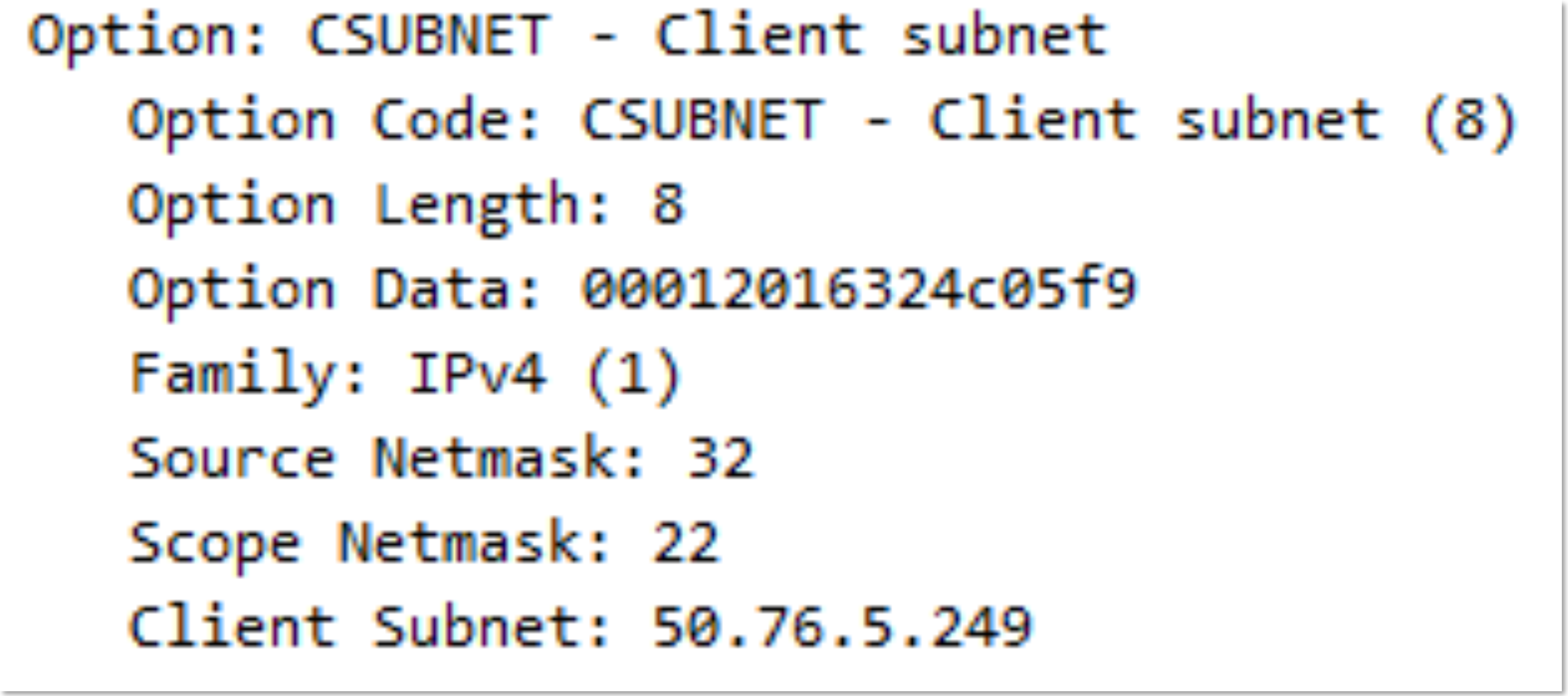
Authoritative

- Supplies answer
- If ECS aware, it sends back a geo-appropriate answer

Client

- Receives best route based upon geolocation
- All on same subnet get same answer

EDNS Client Subnet



```
Option: CSUBNET - Client subnet
    Option Code: CSUBNET - Client subnet (8)
    Option Length: 8
    Option Data: 00012016324c05f9
    Family: IPv4 (1)
    Source Netmask: 32
    Scope Netmask: 22
    Client Subnet: 50.76.5.249
```

EDNS Client Subnet

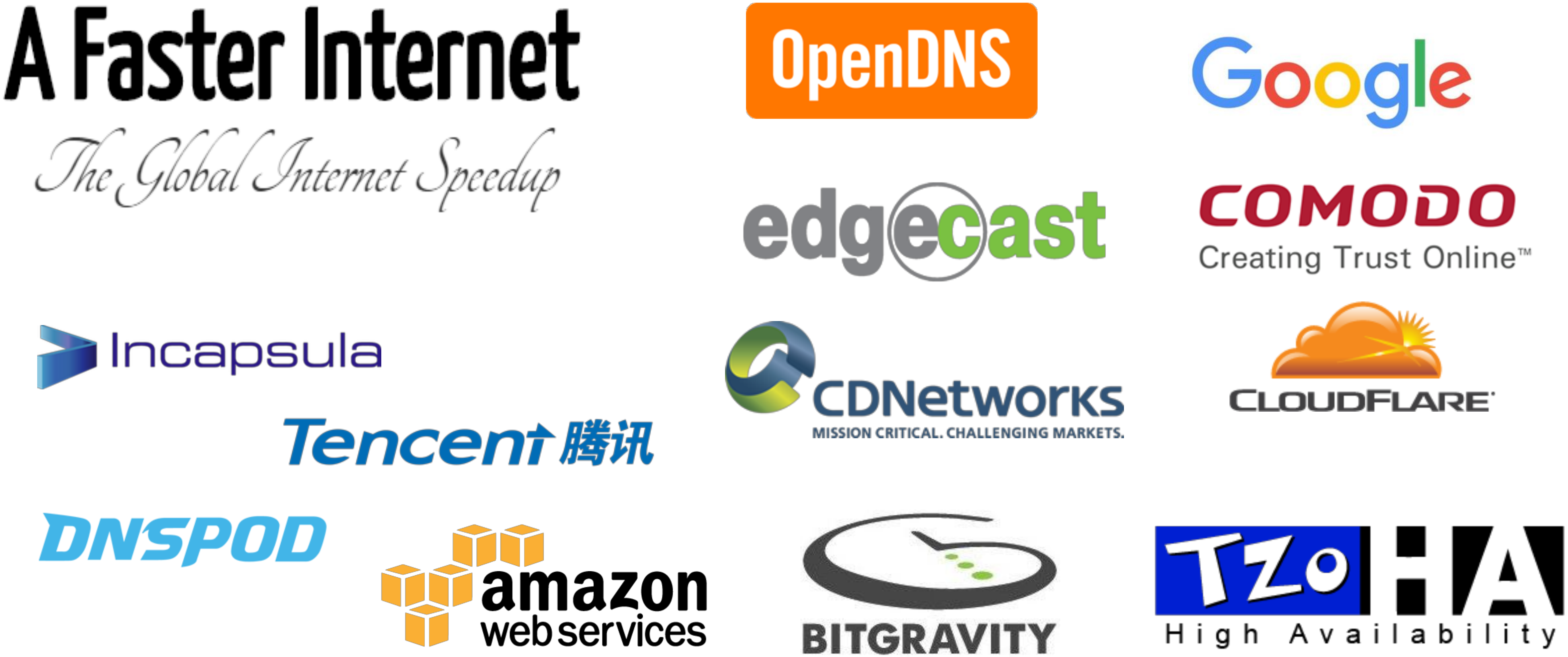
Authoritative

- Google
- Akamai
- NS1
- OpenDNS
- UltraDNS
- PowerDNS
- BIND 9.11
- Amazon CloudFront

Recursive

- Unbound 1.6.2
- PowerDNS
- Google
- OpenDNS
- BIND 9.11
- Amazon CloudFront

EDNS Client Subnet



Tools For Evaluating Use

Name Service Providers

- There is no up-to-date listing or registry showing name service provider support ECS compliant DNS records
- You are relegated reading provider tech material or asking
- A Faster Internet is not current

Recursive Providers

- Again no listing or registry.
- Rely upon material provided by the DNS provider to uncover support
- A Faster Internet is not current

Tools For Evaluating Use

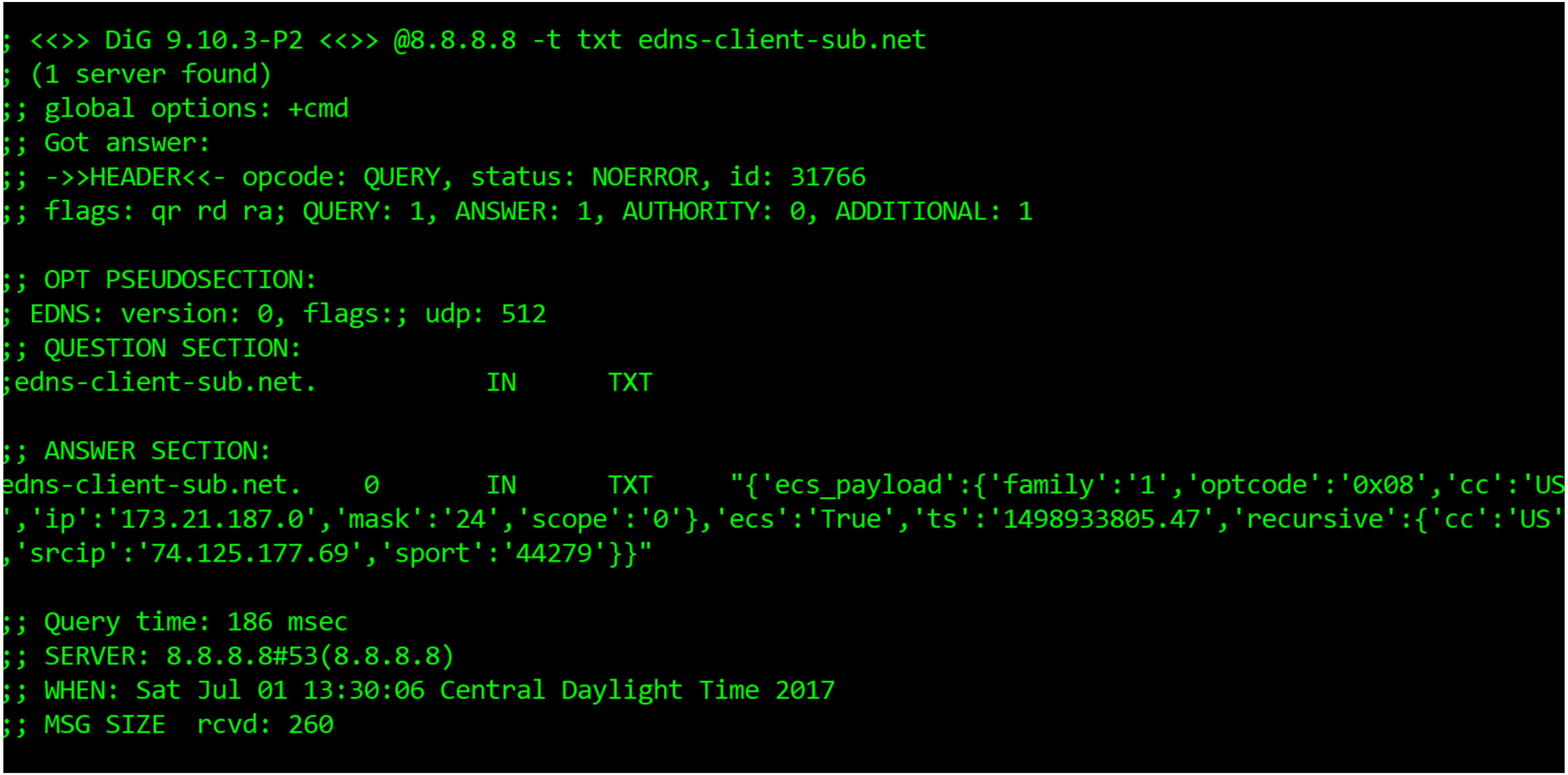
dig @8.8.8.8 +short -t txt edns-client-sub.net

- Targets the name server – 8.8.8.8
- Returns a JSON packet
- https://www.ietf.org/mail-archive/web/dnsop/current/msg16055.html

dig @x.x.x.x –t ns avaliddomain.com +subnet=y.y.y.y

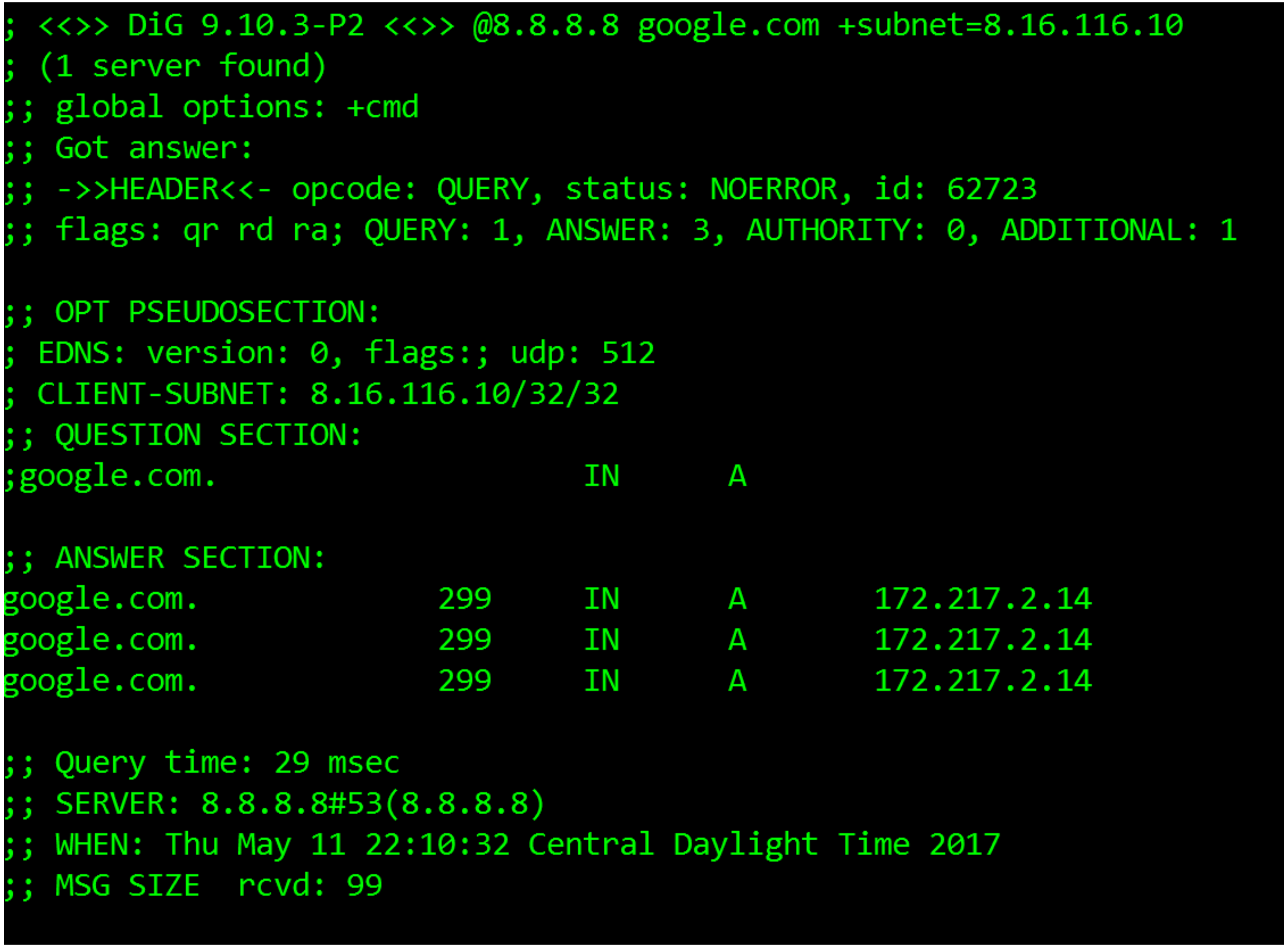
- Targets x.x.x.x
- Supplies ECS data y.y.y.y
- Check OPT PSEUDOSECTION
  - CLIENT-SUBNET: y.y.y.y/M1/M2
  - M1 is Source Netmask
  - M2 is Scope Netmask

Tools For Evaluating Use



```
; <<>> DiG 9.10.3-P2 <<>> @8.8.8.8 -t txt edns-client-sub.net
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 31766
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;edns-client-sub.net.           IN      TXT

;; ANSWER SECTION:
edns-client-sub.net.    0       IN      TXT     "{'ecs_payload':{'family':'1','optcode':'0x08','cc':'US
','ip':'173.21.187.0','mask':'24','scope':'0'},'ecs':'True','ts':'1498933805.47','recursive':{'cc':'US'
,'srcip':'74.125.177.69','sport':'44279'}}"

;; Query time: 186 msec
;; SERVER: 8.8.8.8#53(8.8.8.8)
;; WHEN: Sat Jul 01 13:30:06 Central Daylight Time 2017
;; MSG SIZE  rcvd: 260
```

Tools For Evaluating Use



```
; <<>> DiG 9.10.3-P2 <<>> @8.8.8.8 google.com +subnet=8.16.116.10
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 62723
;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
; CLIENT-SUBNET: 8.16.116.10/32/32
;; QUESTION SECTION:
;google.com.                    IN      A

;; ANSWER SECTION:
google.com.             299     IN      A       172.217.2.14
google.com.             299     IN      A       172.217.2.14
google.com.             299     IN      A       172.217.2.14

;; Query time: 29 msec
;; SERVER: 8.8.8.8#53(8.8.8.8)
;; WHEN: Thu May 11 22:10:32 Central Daylight Time 2017
;; MSG SIZE  rcvd: 99
```

Tools For Evaluating Use

Install Packetbeat on your local DNS resolvers

- Configure to capture DNS traffic
- Forward to Graylog instance Beats input

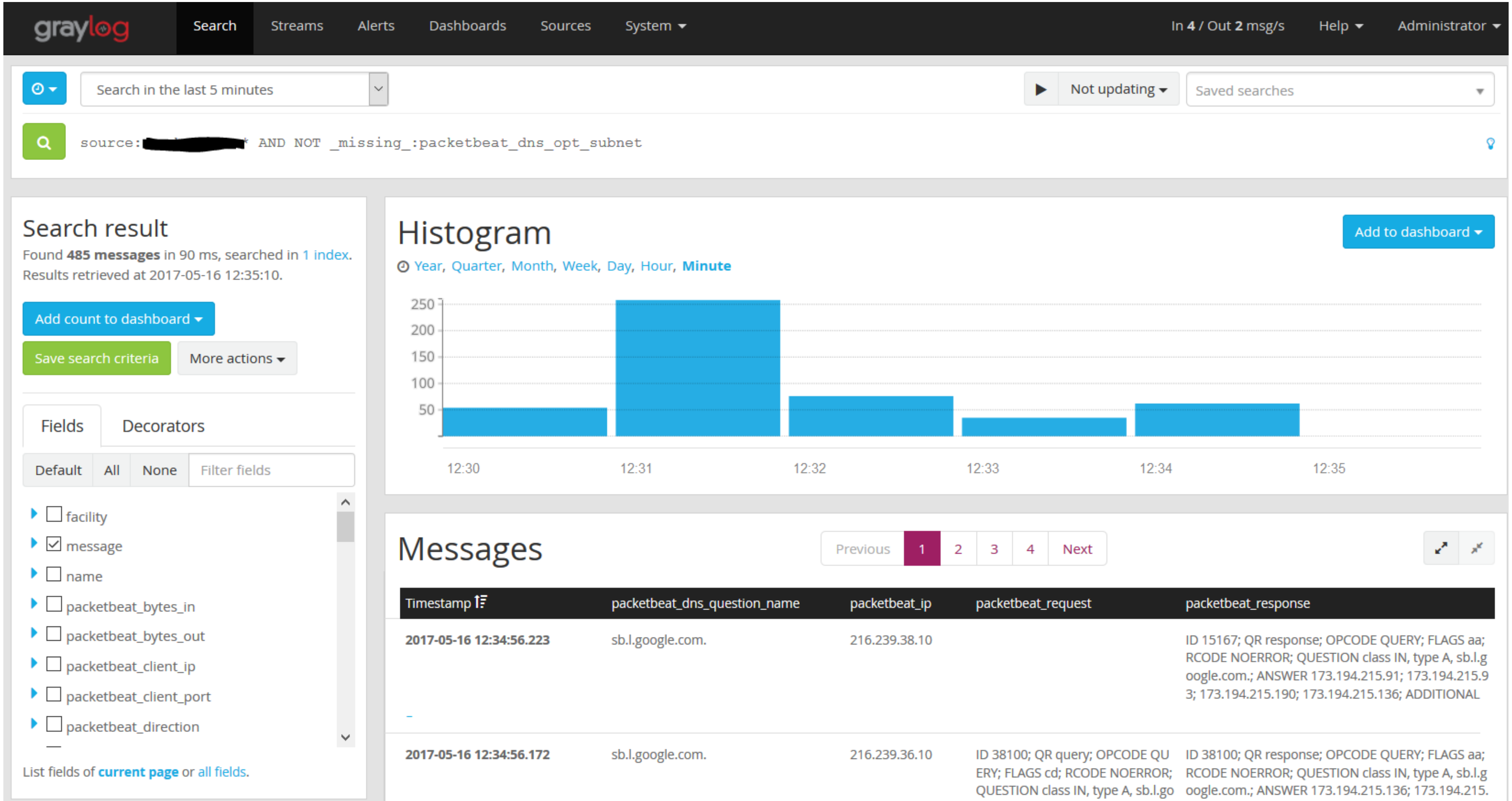
Build a Graylog instance (or download VM)

- Configure Beats input
- Validate data collection

Create a Stream

- Tag all DNS messages w/ data in packetbeat_dns_opt_subnet field
- Correlate to source & destination IPs

Tools For Evaluating Use



```
source:[redacted]* AND NOT _missing_:packetbeat_dns_opt_subnet
```

Search result

Found **485 messages** in 90 ms, searched in 1 index.
Results retrieved at 2017-05-16 12:35:10.

□ facility
☑ message
□ name
□ packetbeat_bytes_in
□ packetbeat_bytes_out
□ packetbeat_client_ip
□ packetbeat_client_port
□ packetbeat_direction

List fields of **current page** or all fields.

Histogram

Messages

Timestamp	packetbeat_dns_question_name	packetbeat_ip	packetbeat_request	packetbeat_response
2017-05-16 12:34:56.223	sb.l.google.com.	216.239.38.10		ID 15167; QR response; OPCODE QUERY; FLAGS aa; RCODE NOERROR; QUESTION class IN, type A, sb.l.google.com.; ANSWER 173.194.215.91; 173.194.215.93; 173.194.215.190; 173.194.215.136; ADDITIONAL
2017-05-16 12:34:56.172	sb.l.google.com.	216.239.36.10	ID 38100; QR query; OPCODE QUERY; FLAGS cd; RCODE NOERROR; QUESTION class IN, type A, sb.l.go	ID 38100; QR response; OPCODE QUERY; FLAGS aa; RCODE NOERROR; QUESTION class IN, type A, sb.l.google.com.; ANSWER 173.194.215.136; 173.194.215.

Tools For Evaluating Use

Capture on local interface

- Run on AD DNS server or span port
- Or open TCP dump file from Linux host

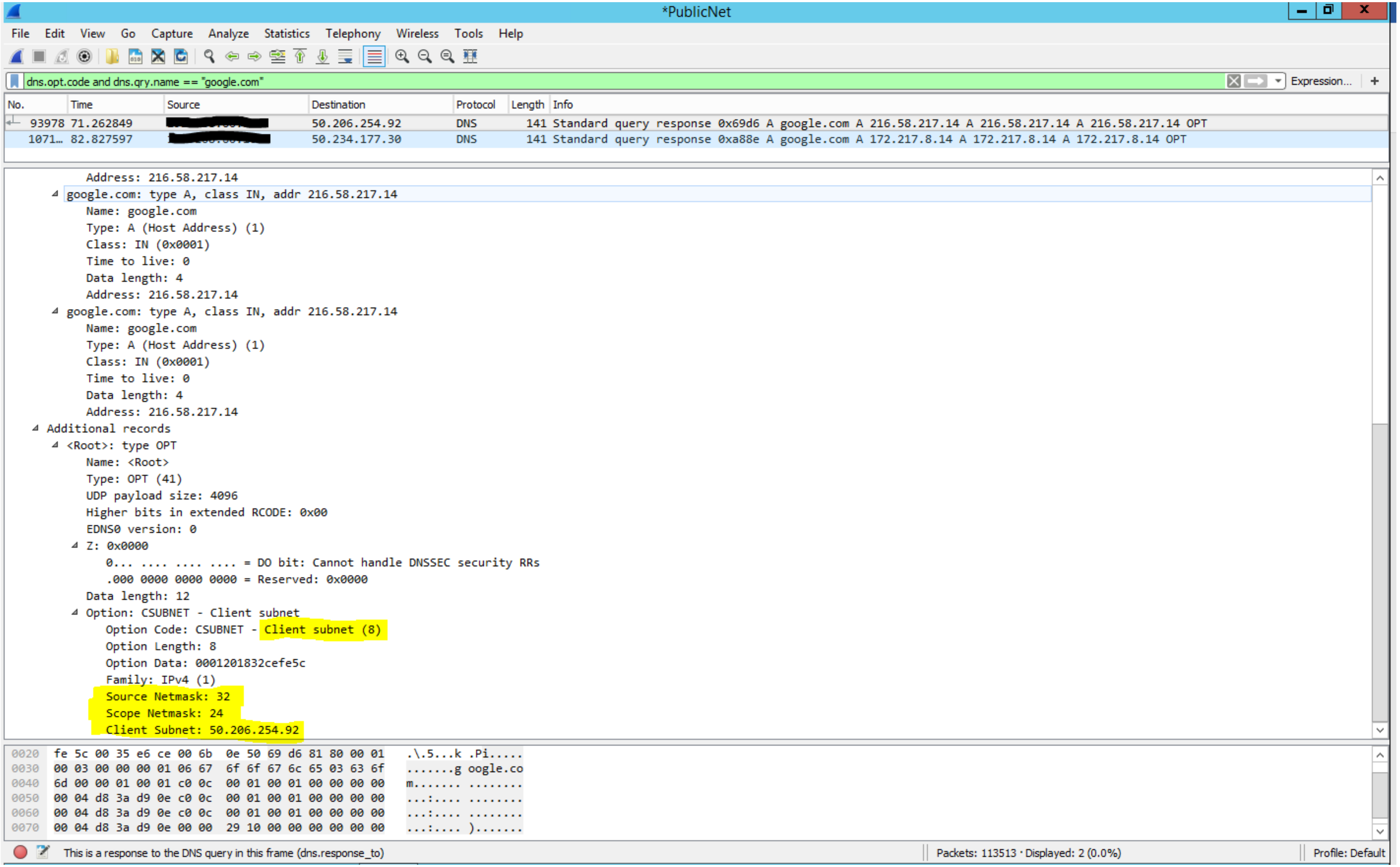
Filter by dns.opt.code == 8

- Remember there are 65,535 possible OPTS
- See what else lurks in your DNS!

Note the data included in RRDATA

- Full IP of requester
- Subnet mask
- Scope mask (all IPs in this mask would get same response)

Tools For Evaluating Use



*PublicNet
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Expression…

```
dns.opt.code and dns.qry.name == "google.com"
```

```
No.      Time       Source        Destination      Protocol  Length  Info
  93978  71.262849  [redacted]    50.206.254.92    DNS          141  Standard query response 0x69d6 A google.com A 216.58.217.14 A 216.58.217.14 A 216.58.217.14 OPT
  1071…  82.827597  [redacted]    50.234.177.30    DNS          141  Standard query response 0xa88e A google.com A 172.217.8.14 A 172.217.8.14 A 172.217.8.14 OPT
```

```
        Address: 216.58.217.14
    ◢ google.com: type A, class IN, addr 216.58.217.14
        Name: google.com
        Type: A (Host Address) (1)
        Class: IN (0x0001)
        Time to live: 0
        Data length: 4
        Address: 216.58.217.14
    ◢ google.com: type A, class IN, addr 216.58.217.14
        Name: google.com
        Type: A (Host Address) (1)
        Class: IN (0x0001)
        Time to live: 0
        Data length: 4
        Address: 216.58.217.14
  ◢ Additional records
    ◢ <Root>: type OPT
        Name: <Root>
        Type: OPT (41)
        UDP payload size: 4096
        Higher bits in extended RCODE: 0x00
        EDNS0 version: 0
      ◢ Z: 0x0000
          0... .... .... .... = DO bit: Cannot handle DNSSEC security RRs
          .000 0000 0000 0000 = Reserved: 0x0000
        Data length: 12
      ◢ Option: CSUBNET - Client subnet
          Option Code: CSUBNET - Client subnet (8)
          Option Length: 8
          Option Data: 0001201832cefe5c
          Family: IPv4 (1)
          Source Netmask: 32
          Scope Netmask: 24
          Client Subnet: 50.206.254.92
```

```
0020  fe 5c 00 35 e6 ce 00 6b  0e 50 69 d6 81 80 00 01   .\.5...k .Pi.....
0030  00 03 00 00 00 01 06 67  6f 6f 67 6c 65 03 63 6f   .......g oogle.co
0040  6d 00 00 01 00 01 c0 0c  00 01 00 01 00 00 00 00   m....... ........
0050  00 04 d8 3a d9 0e c0 0c  00 01 00 01 00 00 00 00   ...:.... ........
0060  00 04 d8 3a d9 0e c0 0c  00 01 00 01 00 00 00 00   ...:.... ........
0070  00 04 d8 3a d9 0e 00 00  29 10 00 00 00 00 00 00   ...:.... ).......
```

This is a response to the DNS query in this frame (dns.response_to)

Packets: 113513 · Displayed: 2 (0.0%)

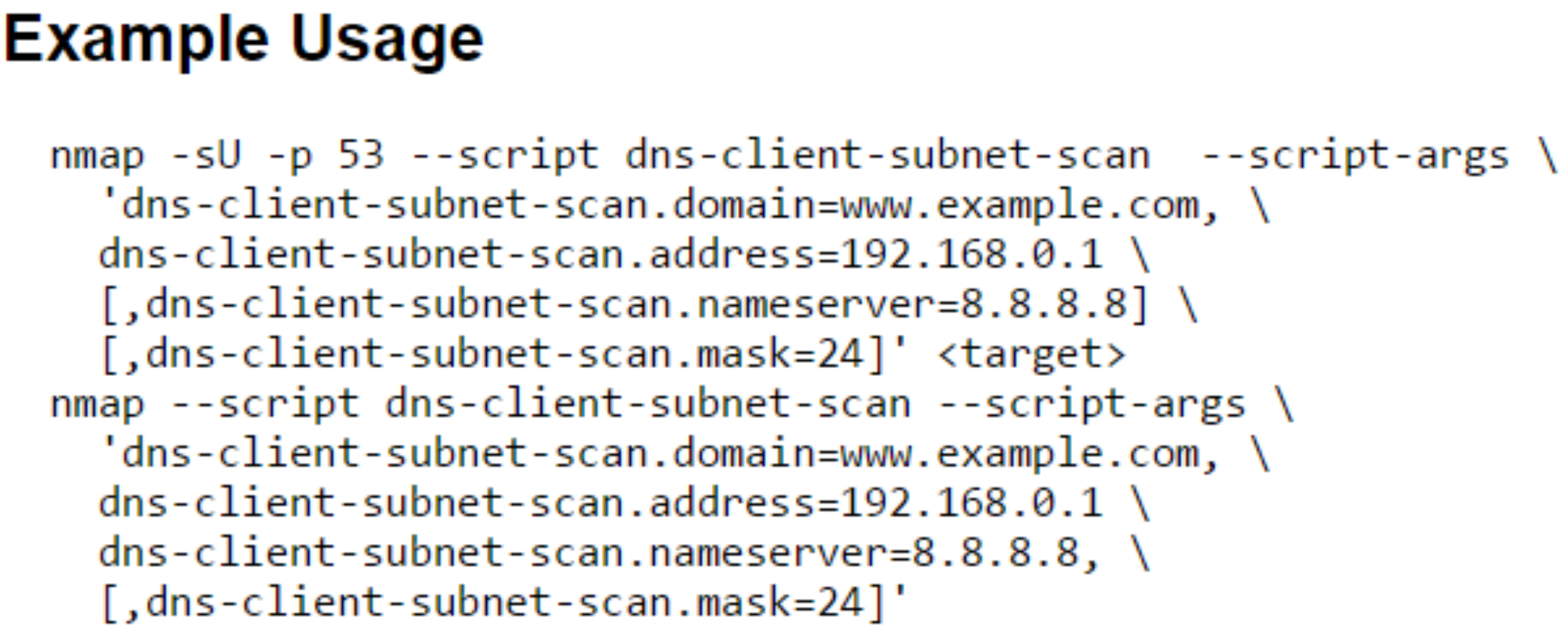
Profile: Default

Tools For Evaluating Use

https://svn.nmap.org/nmap/scripts/dns-client-subnet-scan.nse

Was developed before dig had +subnet= option added

May prove useful for isolated testing



Example Usage

```
nmap -sU -p 53 --script dns-client-subnet-scan  --script-args \
  'dns-client-subnet-scan.domain=www.example.com, \
  dns-client-subnet-scan.address=192.168.0.1 \
  [,dns-client-subnet-scan.nameserver=8.8.8.8] \
  [,dns-client-subnet-scan.mask=24]' <target>
nmap --script dns-client-subnet-scan --script-args \
  'dns-client-subnet-scan.domain=www.example.com, \
  dns-client-subnet-scan.address=192.168.0.1 \
  dns-client-subnet-scan.nameserver=8.8.8.8, \
  [,dns-client-subnet-scan.mask=24]'
```

Tools For Evaluating Use

- Download Alexa Top 1 Million sites - http://s3.amazonaws.com/alexa-static/top-1m.csv.zip
- Get 2nd column containing domain names to a new file
- Run dig to get the nameservers for each domain to a new file
- Sort that file and remove duplicates
- Use Python script to query each nameserver & supply ECS data
- Parse the options returned and record any that return ClientSubnetOption (or any other option)

Tools For Evaluating Use

DNS Packages Supporting EDNS OPT

- .NET
  - ARSoft.Tools.Net
  - Did not find others that supported OPT record manipulation
- Python
  - Dnspython (Nomium project)
  - Twisted Matrix (limited but growing support)
  - Getdns-python-bindings
  - Pydig
- PHP
  - NET_DNS2

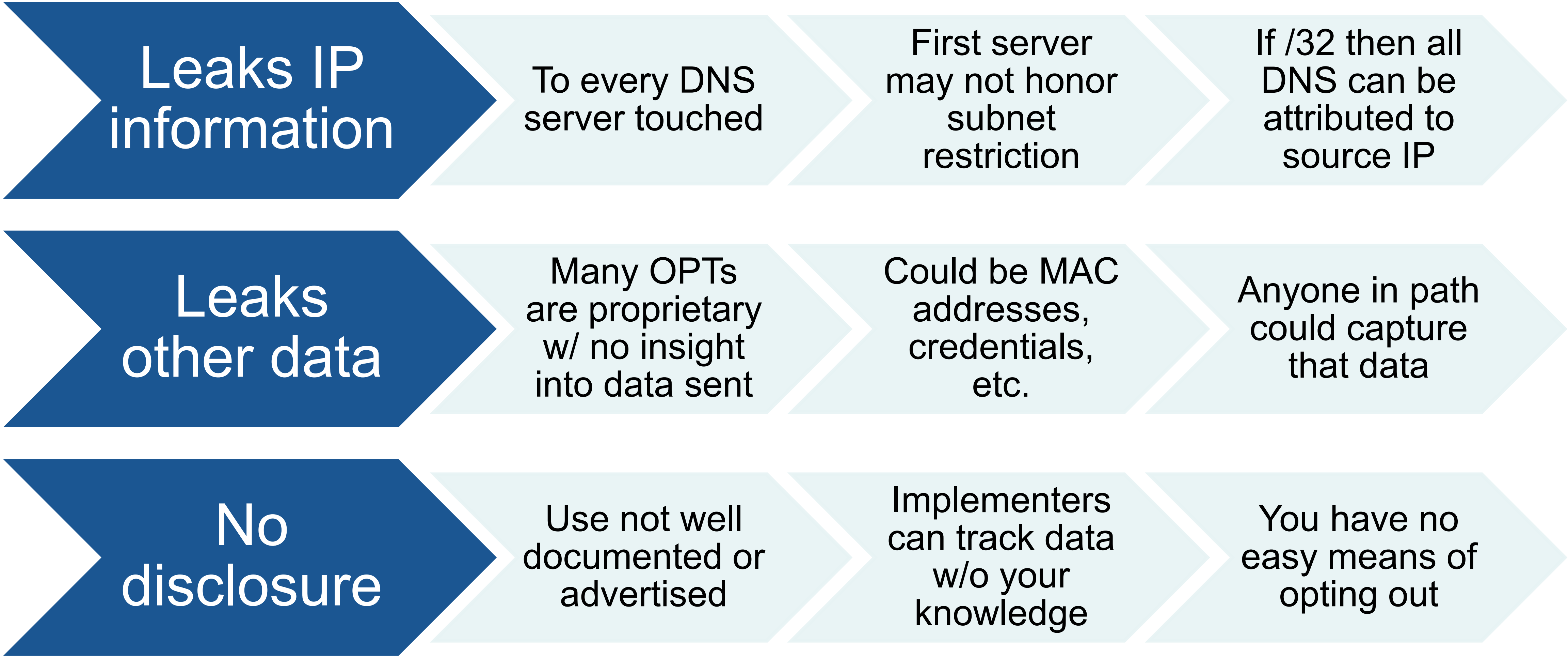
Tools For Evaluating Use

- Scapy
  - Has some DNS functionality for manipulating OPT RR data
  - Only DNSSEC related info (RFC 2671)
  - None for working with Option Codes

Privacy & Security Implications

WE LOST ALL OF OUR COMPANY DATA AND OUR BACKUPS, TOO.
SO I HACKED INTO OUR GOVERNMENT'S SECRET DATABASE WHERE THEY KEEP RECORDS OF EVERYTHING WE SAY OR DO AND GOT IT ALL BACK.
I FEEL AS IF I SHOULD BE DOING SOMETHING NOW.
NAH. EVERYTHING IS WORKING FINE.
Dilbert.com DilbertCartoonist@gmail.com
9-6-13 ©2013 Scott Adams, Inc./Dist. by Universal Uclick

Privacy & Security Implications



Leaks IP information

- To every DNS server touched
- First server may not honor subnet restriction
- If /32 then all DNS can be attributed to source IP

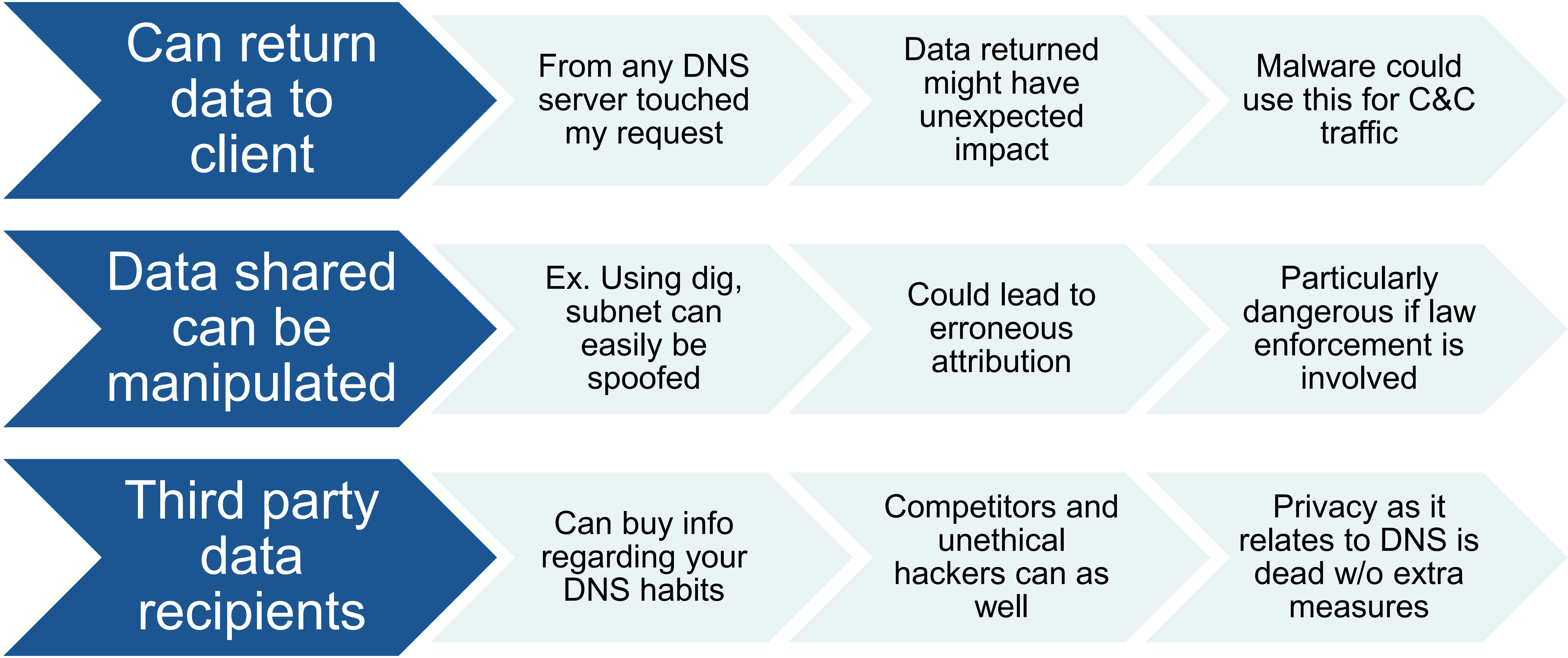
Leaks other data

- Many OPTs are proprietary w/ no insight into data sent
- Could be MAC addresses, credentials, etc.
- Anyone in path could capture that data

No disclosure

- Use not well documented or advertised
- Implementers can track data w/o your knowledge
- You have no easy means of opting out

Privacy & Security Implications



Can return data to client

From any DNS server touched my request

Data returned might have unexpected impact

Malware could use this for C&C traffic

Data shared can be manipulated

Ex. Using dig, subnet can easily be spoofed

Could lead to erroneous attribution

Particularly dangerous if law enforcement is involved

Third party data recipients

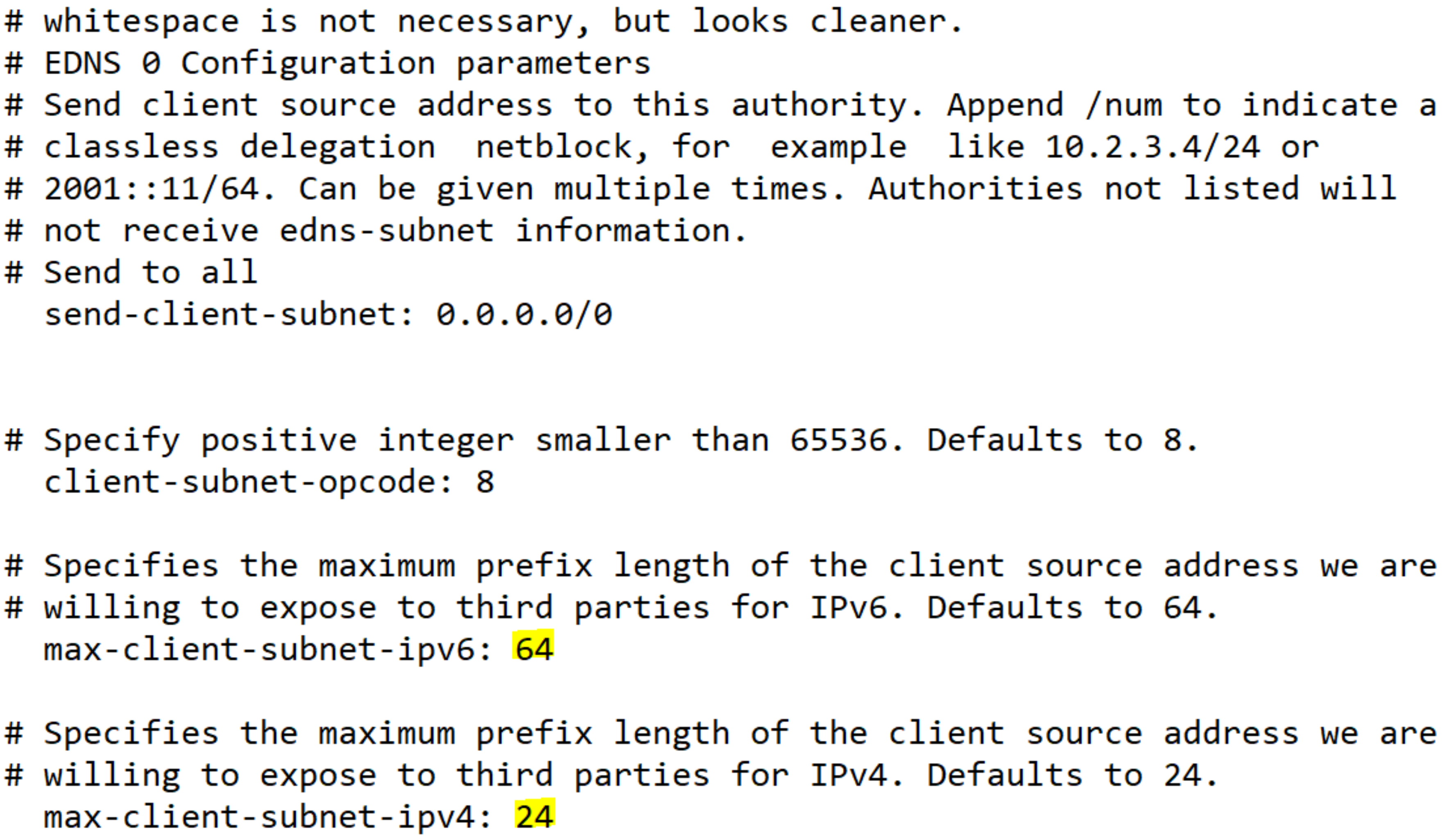
Can buy info regarding your DNS habits

Competitors and unethical hackers can as well

Privacy as it relates to DNS is dead w/o extra measures

Privacy & Security Implications

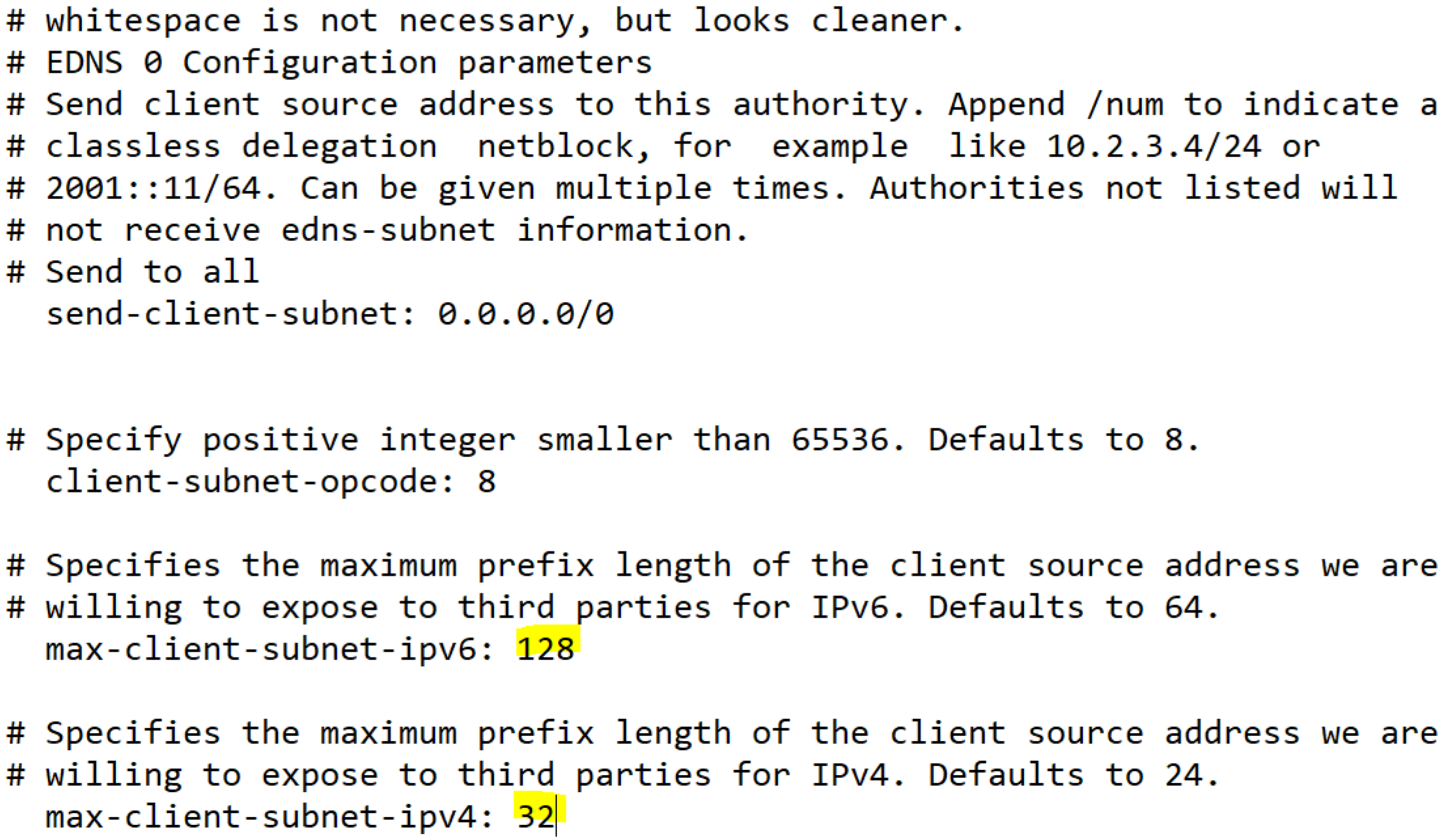
Example Correct Configuration in Unbound



```
# whitespace is not necessary, but looks cleaner.
# EDNS 0 Configuration parameters
# Send client source address to this authority. Append /num to indicate a
# classless delegation  netblock, for  example  like 10.2.3.4/24 or
# 2001::11/64. Can be given multiple times. Authorities not listed will
# not receive edns-subnet information.
# Send to all
  send-client-subnet: 0.0.0.0/0


# Specify positive integer smaller than 65536. Defaults to 8.
  client-subnet-opcode: 8

# Specifies the maximum prefix length of the client source address we are
# willing to expose to third parties for IPv6. Defaults to 64.
  max-client-subnet-ipv6: 64

# Specifies the maximum prefix length of the client source address we are
# willing to expose to third parties for IPv4. Defaults to 24.
  max-client-subnet-ipv4: 24
```

Privacy & Security Implications

Example Incorrect Configuration in Unbound



```
# whitespace is not necessary, but looks cleaner.
# EDNS 0 Configuration parameters
# Send client source address to this authority. Append /num to indicate a
# classless delegation  netblock, for  example  like 10.2.3.4/24 or
# 2001::11/64. Can be given multiple times. Authorities not listed will
# not receive edns-subnet information.
# Send to all
  send-client-subnet: 0.0.0.0/0


# Specify positive integer smaller than 65536. Defaults to 8.
  client-subnet-opcode: 8

# Specifies the maximum prefix length of the client source address we are
# willing to expose to third parties for IPv6. Defaults to 64.
  max-client-subnet-ipv6: 128

# Specifies the maximum prefix length of the client source address we are
# willing to expose to third parties for IPv4. Defaults to 24.
  max-client-subnet-ipv4: 32
```

Privacy & Security Implications

Defensive Options

- Know what’s normal
- Understand IPv6 vs IPv4
- Route ALL DNS to knowns recursive resolvers that do not pass EDNS OPT data or pass fake data
- Lock out non validated DNS at edge
- Disable EDNS(0) all together (not cool)
- Monitor DNS using Packetbeat / Graylog & full capture if needed (Bro, Security Onion, etc.)
- Create IPS rules as needed
- Enforce DNSSEC (Stuff will break!)

Privacy & Security Implications

Offensive Options

- Create noise – generate scripted DNS w/ forged OPT data to confuse the upstream collectors
- Use full VPN tunnel to route ALL traffic through “safe” exit point
- Tor past that safe end point
- Account for IPv6 traffic as well
- Disable IPv6 temporarily
- Use TorGhost (Only works with IPv4)
- Test w/ Wireshark or TCPDump

Summary

OPT data have several useful purposes

- Allow CDN via DNS
- Enables DNSSEC
- Enables multi-tenant cache servers to cache data for differing end points
- DNS responses can be altered quickly in case of traffic overload
- Signature of compromise can be attributed to IP or MAC

OPT data use have privacy concerns

- All servers in DNS path if EDNS capable can track data
- No standard has been developed for opting out
- Privacy is compromised when EDNS OPT data is forwarded
- No mechanism to verify OPT data is accurate
- Data mining likely once providers fully implement

OPT data have potential for abuse

- Data could be easily spoofed
- 65,535 possible OPT choices
- Botnet C&C
- Data exfiltration

Wrap Up

- Questions & Answers
- Contact Info
  - jnitterauer@appriver.com
  - @jnitterauer
  - https://www.linkedin.com/in/jnitterauer
  - 850-932-5338 ext. 6468