DNS

Vitaly Shmatikov

presented by Eugene Bagdasaryan

Google DNS 8.8.8.8/32 was hijacked for ~22min yesterday, affecting networks in Brazil & Venezuela #bgp #hijack #dns pic.twitter.com/wlBuui8dwO

↩ Reply ⇄ Retweet ★ Favorite ••• More

BGP

🏠 HOME

My Alerts
....
Alerts Details

🛠 Tools

It is suspected that hackers exploited a well-known vulnerability in the so-called Border Gateway Protocol (BGP)

Detected Origin AS: 7908
Expected Origin AS: 15169

RETWEETS	FAVORITES
805	156

Turkey (2014)

DNS Hostname vs. IP Address

◆ DNS hostname (e.g., www.cs.cornell.edu)

- Mnemonic name understood by humans
- Variable length, full alphabet of characters
- Provides little (if any) information about location

◆ IP address (e.g., 128.84.202.53)

- Numerical address understood by routers
- Fixed length, decimal number
- Hierarchical address space, related to host location

Uses of DNS

◆ Hostname to IP address translation

- Reverse lookup: IP address to hostname translation

◆ Host name aliasing: other DNS names for a host

- Alias hostnames point to canonical hostname

◆ Email: look up domain's mail server by domain name

Different DNS Mappings

◆ 1-1 mapping between domain name and IP addr

- www.cs.cornell.edu maps to 132.236.207.20

◆ Multiple domain names maps to the same IP addr

- eecs.mit.edu and cs.mit.edu both map to 18.62.1.6

◆ Single domain name maps to multiple IP addrs

- aol.com and www.aol.com map to multiple IP addrs.

◆ Some valid domain names don't map to any IP addr

- cmcl.cs.cmu.edu

Original Design of DNS

◆ Per-host file named /etc/hosts
- Flat namespace: each line = IP address & DNS name
- SRI (Menlo Park, California) kept the master copy
- Everyone else downloaded regularly

◆ A single server doesn't scale
- Traffic explosion (lookups and updates)
- Single point of failure

◆ Need a distributed, hierarchical collection of servers

Goals of DNS

◆ A wide-area distributed database

- Possibly biggest such database in the world!

◆ Goals

- Scalability; decentralized maintenance
- Robustness
- Global scope
- Names mean the same thing everywhere
- Distributed updates/queries
- Good performance

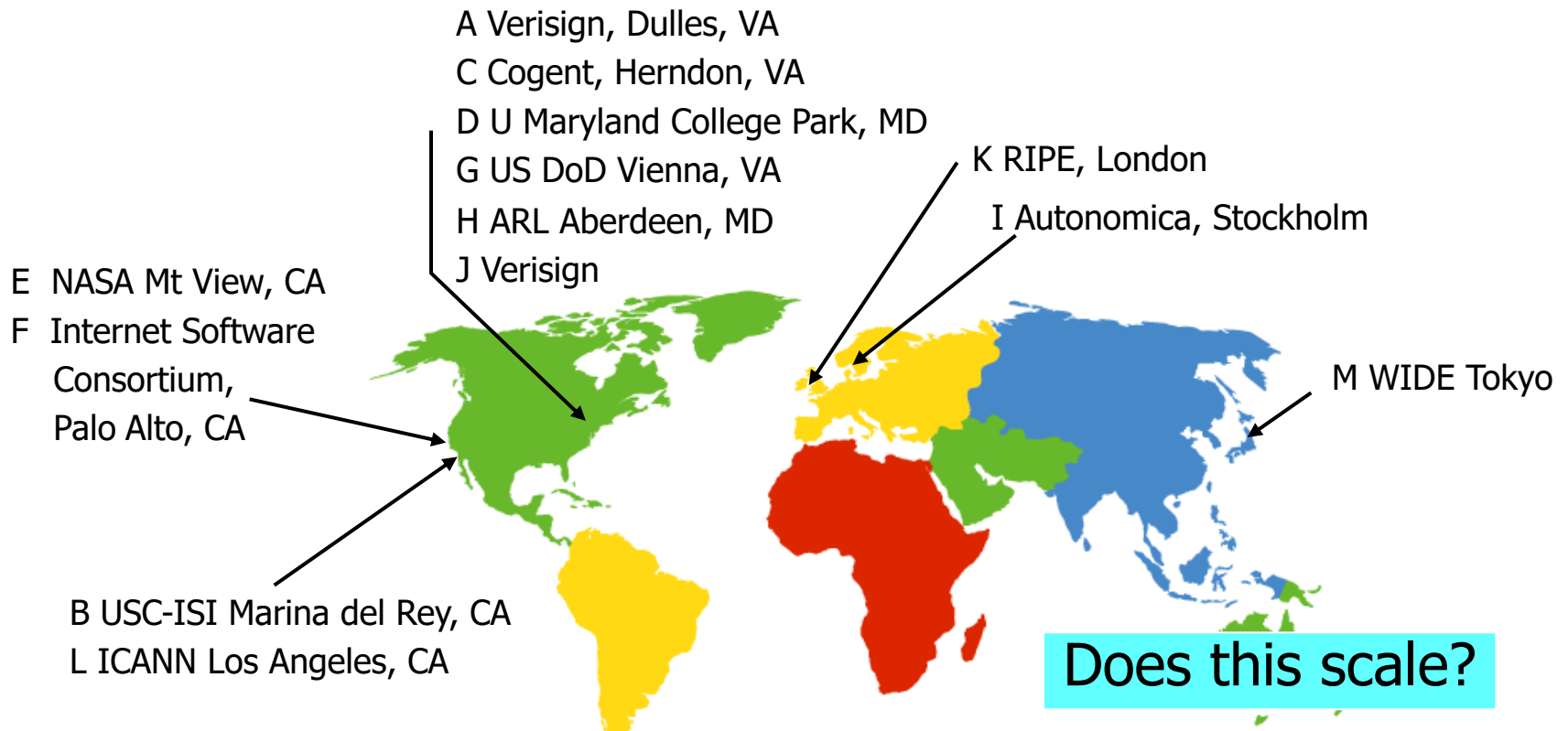◆ Do <u>not</u> need strong consistency properties

DNS

◆ Hierarchical name space divided into contiguous sections called zones

- Zones are distributed over a collection of DNS servers

◆ Hierarchy of DNS servers

- Root servers (identity hardwired into other servers)
- Top-level domain (TLD) servers
- Authoritative DNS servers

◆ Performing the translations

- Local DNS servers located near clients
- Resolver software running on clients

Hierarchical Structure of DNS

Root

. ← Root

TLDs: com.   gov.   edu.

fcc.gov.   cornell.edu.   nyu.edu.

cs.cornell.edu.
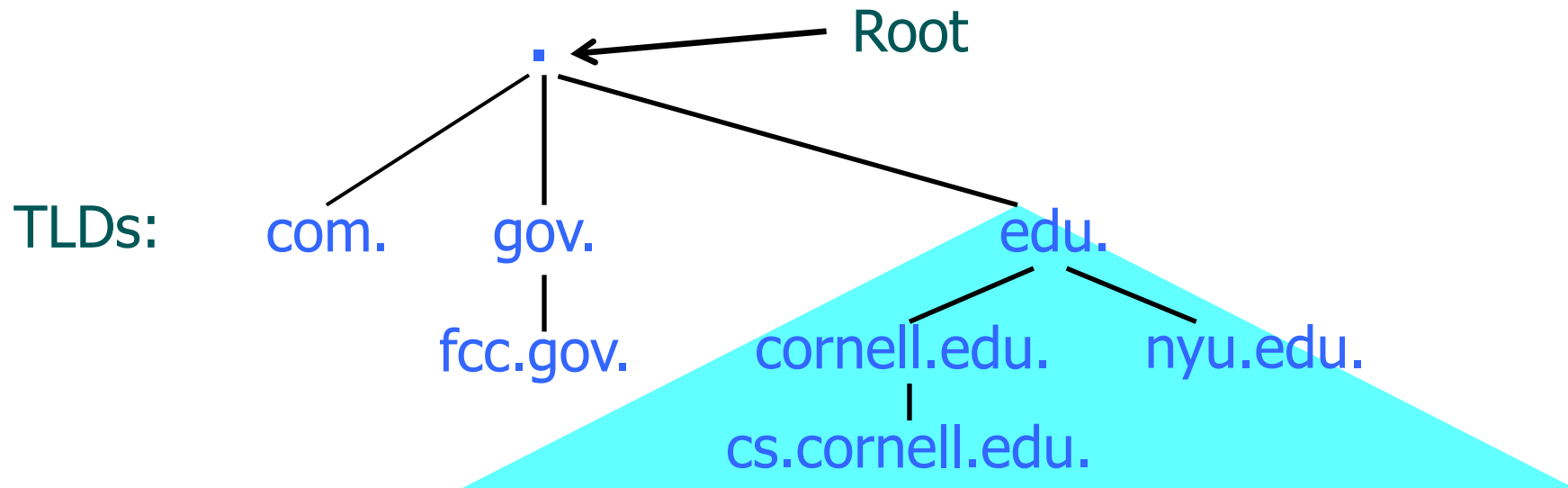
◆ Hierarchy of namespace matches hierarchy of servers
◆ Set of nameservers answers queries for names within zone
◆ Nameservers store names and links to other servers in tree

13 DNS Root Nameservers

A Verisign, Dulles, VA
C Cogent, Herndon, VA
D U Maryland College Park, MD
G US DoD Vienna, VA
H ARL Aberdeen, MD
J Verisign

K RIPE, London
I Autonomica, Stockholm

E NASA Mt View, CA
F Internet Software
   Consortium,
   Palo Alto, CA

M WIDE Tokyo

B USC-ISI Marina del Rey, CA
L ICANN Los Angeles, CA

Does this scale?

Each server is really a cluster of servers (some distributed over a small geographical region), replicated via IP anycast which routes DNS queries to any server in that cluster of servers, to spread load

TLD and Authoritative Servers

◆ Top-level domain (TLD) servers

- Responsible for com, org, net, edu, etc, and all top-level country domains: uk, fr, ca, jp
- Network Solutions maintains servers for .com TLD
- Educause (non-profit) for .edu TLD

◆ Authoritative DNS servers

- An organization's DNS servers, providing authoritative information for that organization
- May be maintained by organization itself, or ISP

Local Name Servers

◆ Each ISP (or company, or university) has one
  • No strict hierarchy

◆ Also called default or caching name server

◆ When a host makes DNS query, query is sent to its local DNS server, which acts as proxy and forwards query into hierarchy

DNS Resource Records

◆ DNS is a distributed database storing resource records

◆ Resource record includes: (name, type, value, time-to-live)

Type = A (address)
- name = hostname
- value is IP address

Type = CNAME
- name = alias for some "canonical" (real) name
- value = canonical name

Type = NS (name server)
- name = domain (e.g. cornell.edu)
- Value = hostname of authoritative name server for this domain
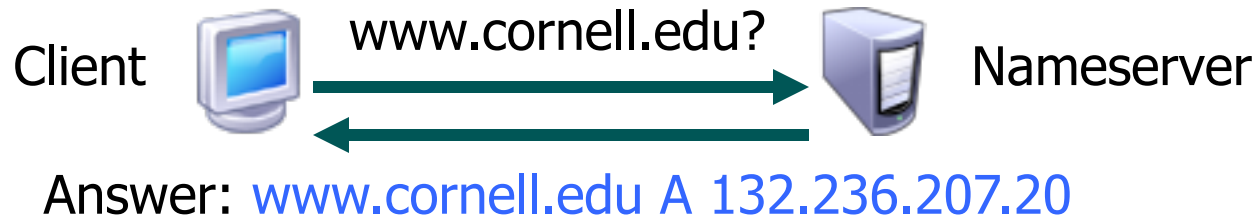
Type = MX (mail exchange)
- name = domain
- value = name of mail server for that domain

DNS in Operation

Most queries and responses are UDP datagrams

◆ **Recursive**

- Nameserver responds with answer or error

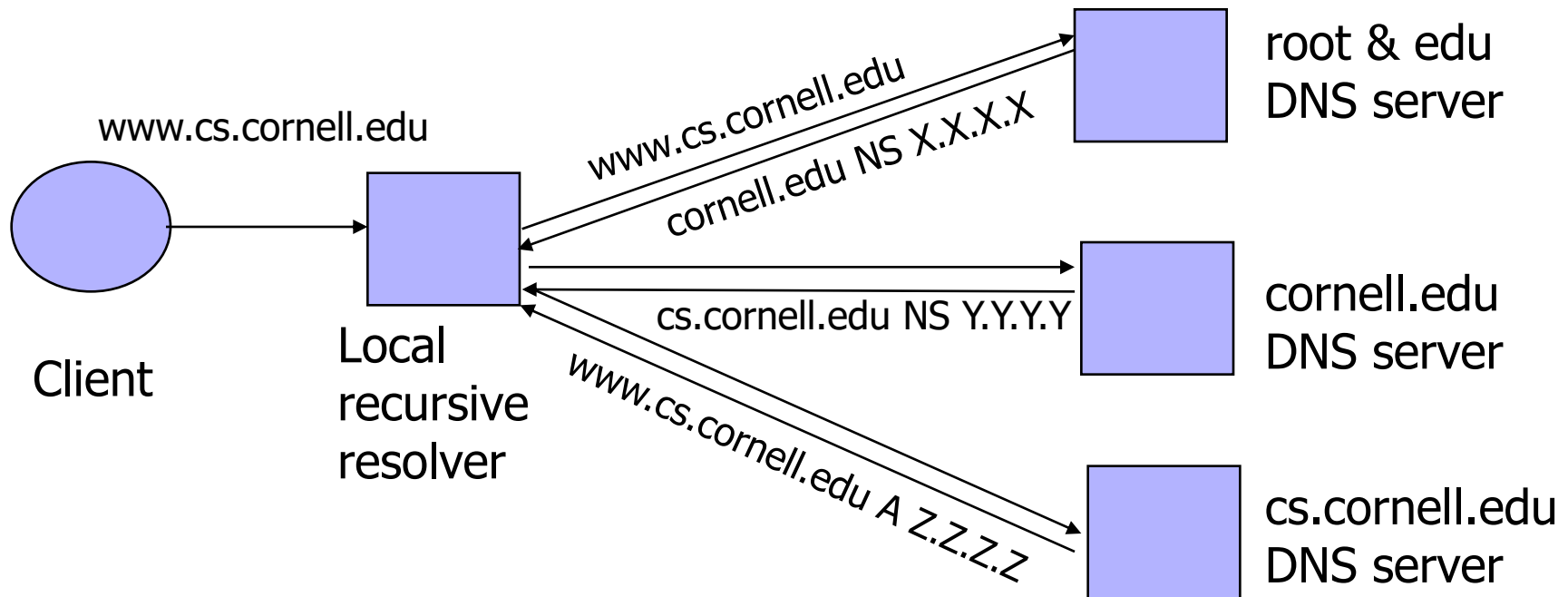Client   www.cornell.edu?   Nameserver

Answer: www.cornell.edu A 132.236.207.20

◆ **Iterative**

- Nameserver may respond with a referral

Client   www.cornell.edu?   Nameserver

Referral: edu NS a.edu-servers.net
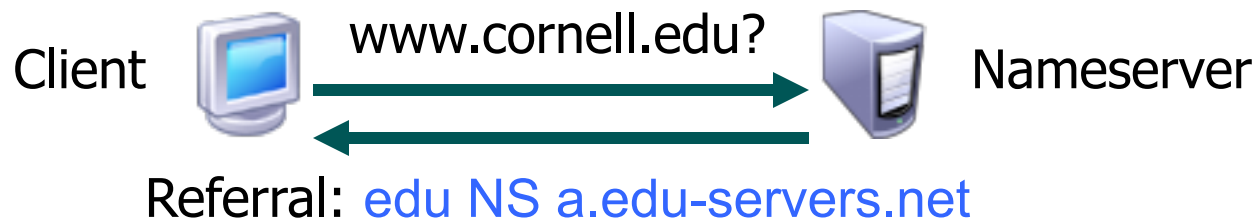
Recursive DNS Resolution

Recursive vs. Iterative Queries

Recursive

◆ Less burden query initiator

◆ More burden on nameserver
- Has to return an answer

◆ Most root and TLD servers won't answer (shed load)

◆ Local name server answers recursive query

Iterative

◆ More burden on query initiator

◆ Less burden on nameserver
- Refers query to another nameserver

```
$ dig @a.root-servers.net www.freebsd.org +norecurse
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 57494
;; QUERY: 1, ANSWER: 0, AUTHORITY: 2, ADDITIONAL: 2

;; QUESTION SECTION:
;www.freebsd.org.               IN      A
```

Time to live in seconds

```
;; AUTHORITY SECTION:
org.                    172800 IN      NS      b0.org.afilias-nst.org.
org.                    172800 IN      NS      d0.org.afilias-nst.org.

;; ADDITIONAL SECTION:
b0.org.afilias-nst.org.         172800 IN      A       199.19.54.1
d0.org.afilias-nst.org.         172800 IN      A       199.19.57.1
```

Glue records

[Output edited for clarity]

(authoritative for org.)

```
$ dig @199.19.54.1 www.freebsd.org +norecurse
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 39912
;; QUERY: 1, ANSWER: 0, AUTHORITY: 3, ADDITIONAL: 0

;; QUESTION SECTION:
;www.freebsd.org.               IN      A


;; AUTHORITY SECTION:
freebsd.org.            86400  IN      NS      ns1.isc-sns.net.
freebsd.org.            86400  IN      NS      ns2.isc-sns.com.
freebsd.org.            86400  IN      NS      ns3.isc-sns.info.
```

(authoritative for freebsd.org.)

```
$ dig @ns1.isc-sns.net www.freebsd.org +norecurse
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 17037
;; QUERY: 1, ANSWER: 1, AUTHORITY: 3, ADDITIONAL: 3

;; QUESTION SECTION:
;www.freebsd.org.              IN      A

;; ANSWER SECTION:
www.freebsd.org.    3600  IN      A       69.147.83.33

;; AUTHORITY SECTION:
freebsd.org.        3600  IN      NS      ns2.isc-sns.com.
freebsd.org.        3600  IN      NS      ns1.isc-sns.net.
freebsd.org.        3600  IN      NS      ns3.isc-sns.info.

;; ADDITIONAL SECTION:
ns1.isc-sns.net.    3600  IN      A       72.52.71.1
ns2.isc-sns.com.    3600  IN      A       38.103.2.1
ns3.isc-sns.info.   3600  IN      A       63.243.194.1
```

[Output edited for clarity]

DNS Caching

◆Performing all these queries takes time

- … <u>before</u> actual communication takes place

◆Caching can greatly reduce overhead

- Top-level servers very rarely change
- Popular sites visited often

◆How DNS caching works

- All DNS servers cache responses to queries
  – Including negative responses (e.g., misspellings)
- Responses include a time-to-live (TTL) field
- Server deletes cached entry after TTL expires

Cached Lookup Example

ftp.cs.cornell.edu

Client

Local
DNS recursive
resolver

root & edu
DNS server

ftp.cs.cornell.edu

ftp=IPaddr

cornell.edu
DNS server

cs.cornell.edu
DNS server
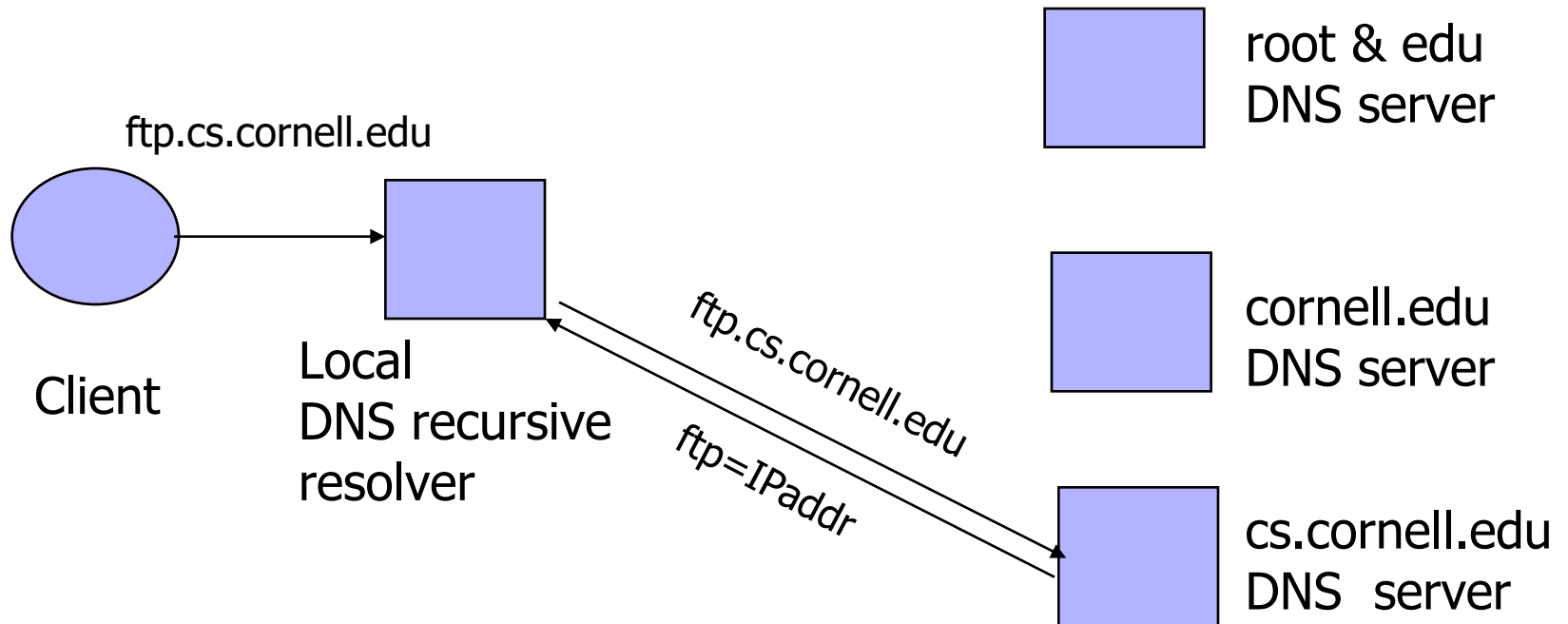
What if DNS is Subverted?

◆ Redirect victim's web traffic to rogue servers

◆ Redirect victim's email to rogue email servers (MX records in DNS)

◆ Does Secure Sockets Layer (SSL) provide protection?

   ◆ Yes—user will get "wrong certificate" if SSL enabled

   ◆ No—SSL not enabled or user ignores warnings

   ◆ No—how is SSL trust established? Often, by email!

Problem #1: Coffee Shop

◆ As you sip your latte and surf the Web, how does your laptop find google.com?

◆ Answer: it asks the local DNS nameserver

… which is run by the coffee shop or their contractor

… and can return to you any answer they please

◆ How can you know you're getting correct data?

Problem #2: DNS "Authentication"

Request contains random 16-bit TXID

Response accepted if TXID is the same, stays in cache for a long time (TTL)

www.cs.cornell.edu

Client

Local
DNS recursive
resolver

root & edu
DNS server

www.cs.cornell.edu

NS cornell.edu

www=IPaddr

cs.cornell.edu
DNS server
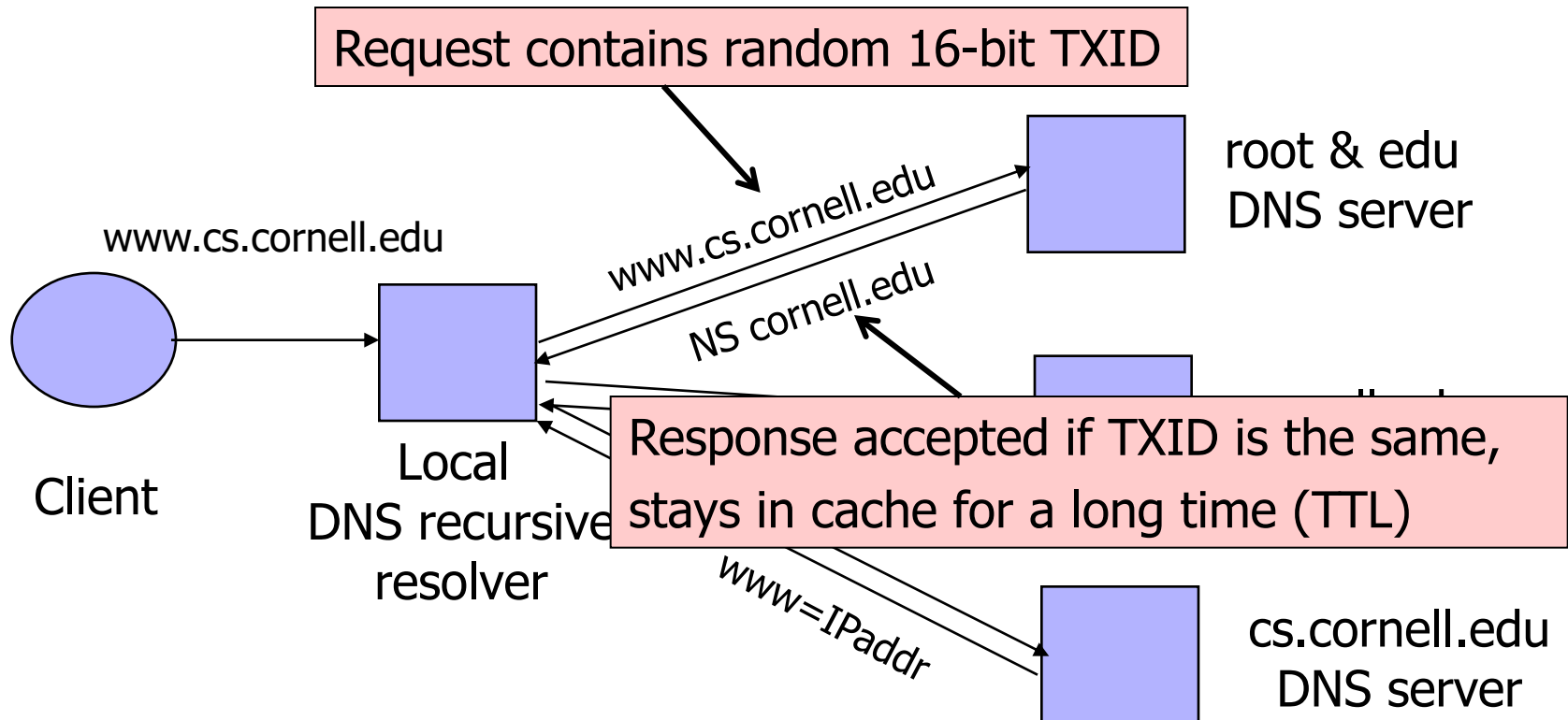
DNS Spoofing by Off-Path Attacker

6.6.6.6

Trick client into looking up host1.foo.com (how?)

Guess TXID, host1.foo.com is at 6.6.6.6

Another guess, host1.foo.com is at 6.6.6.6

Another guess, host1.foo.com is at 6.6.6.6

Client

host1.foo.com

Local resolver

TXID, host1.foo.com

host1.foo.com is at 1.2.3.4

ns.foo.com
DNS server
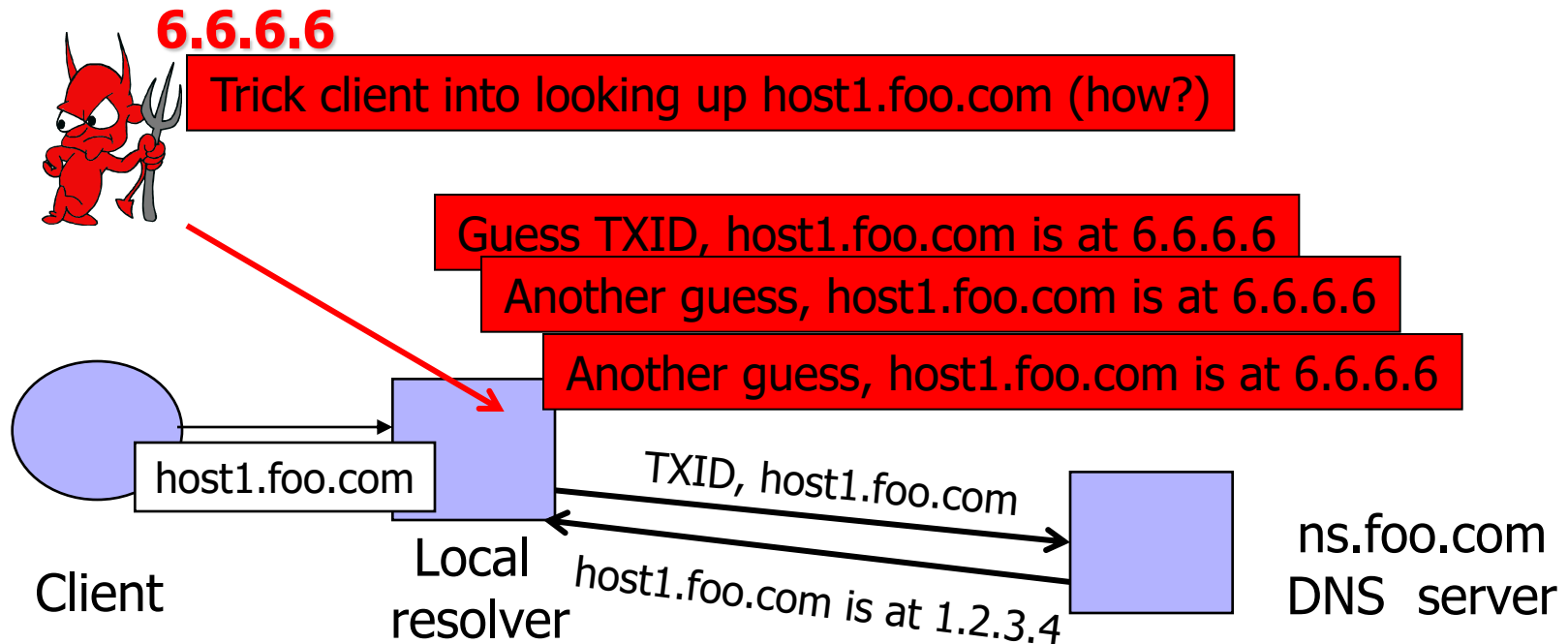
Several opportunities to win the race.

If attacker loses, has to wait until TTL expires…

… but can try again with host2.foo.com, host3.foo.com, etc.

… but what's the point of hijacking host3.foo.com?

Exploiting Recursive Resolving

[Kaminsky]

6.6.6.6

Trick client into looking up host1.foo.com

Guessed TXID, very long TTL
I don't know where host1.foo.com is, but
ask the authoritative server at ns2.foo.com
It lives at 6.6.6.6

host2.foo.com

host1.foo.com

TXID, host1.foo.com

host1.foo.com is at 1.2.3.4

Client

Local resolver

ns.foo.com DNS server
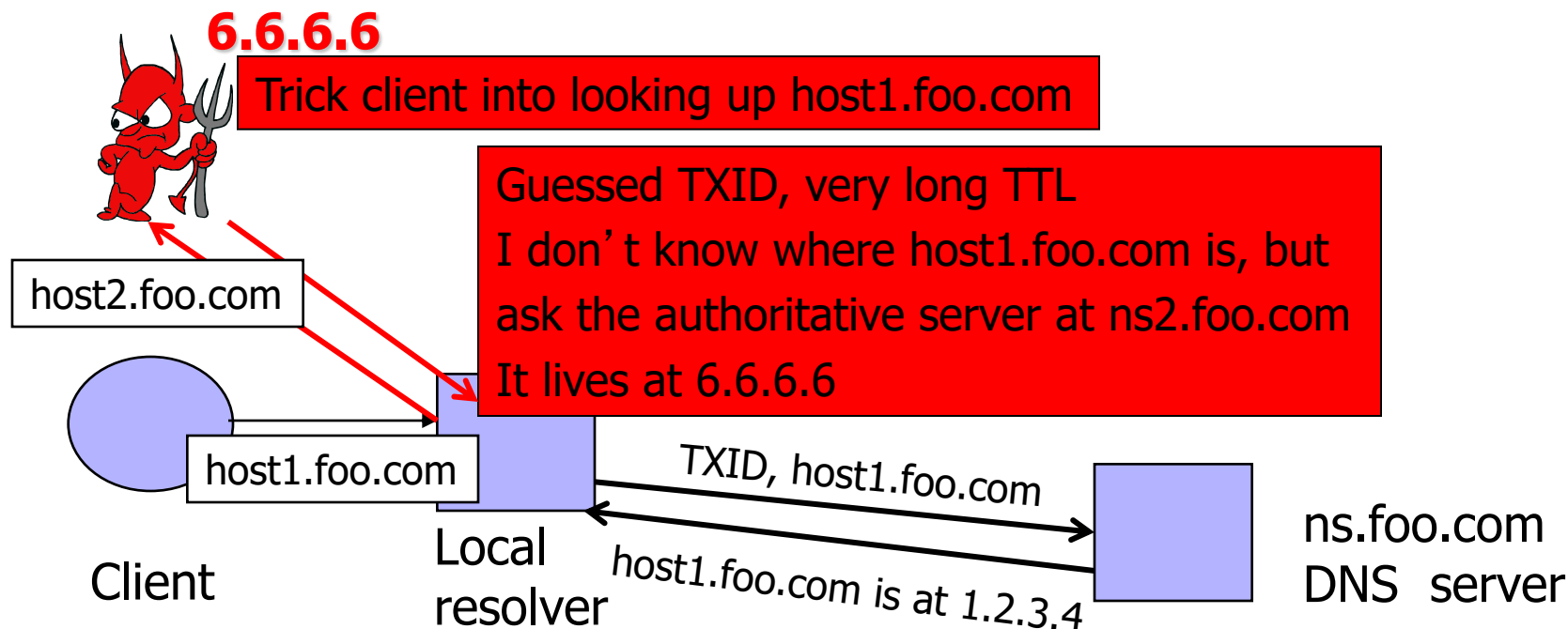
If win the race, any request for XXX.foo.com will go to 6.6.6.6
The cache is poisoned… for a very long time!
No need to win future races!
If lose, try again with <ANYTHING>.foo.com

Triggering a Race

◆ Any link, any image, any ad, anything can cause a DNS lookup

- No JavaScript required, though it helps

◆ Mail servers will look up what bad guy wants

- On first greeting: HELO
- On first learning who they're talking to: MAIL FROM
- On spam check (oops!)
- When trying to deliver a bounce
- When trying to deliver a newsletter
- When trying to deliver an actual response from an actual employee

Reverse DNS Spoofing

◆ Trusted access is often based on host names

- Example: permit all hosts in .rhosts to run remote shell

◆ Network requests such as rsh or rlogin arrive from numeric source addresses

- System performs reverse DNS lookup to determine requester's host name and checks if it's in .rhosts

◆ If attacker can spoof the answer to reverse DNS query, he can fool target machine into thinking that request comes from an authorized host

- No authentication for DNS responses and typically no double-checking (numeric → symbolic → numeric)

Solving the DNS Spoofing Problem

◆ Long TTL for legitimate responses

- Does it really help?

◆ Randomize port in addition to TXID

- 32 bits of randomness, makes it harder for attacker to guess TXID+port

◆ DNSSEC

- Cryptographic authentication of host-address mappings

DNS Amplification Attack
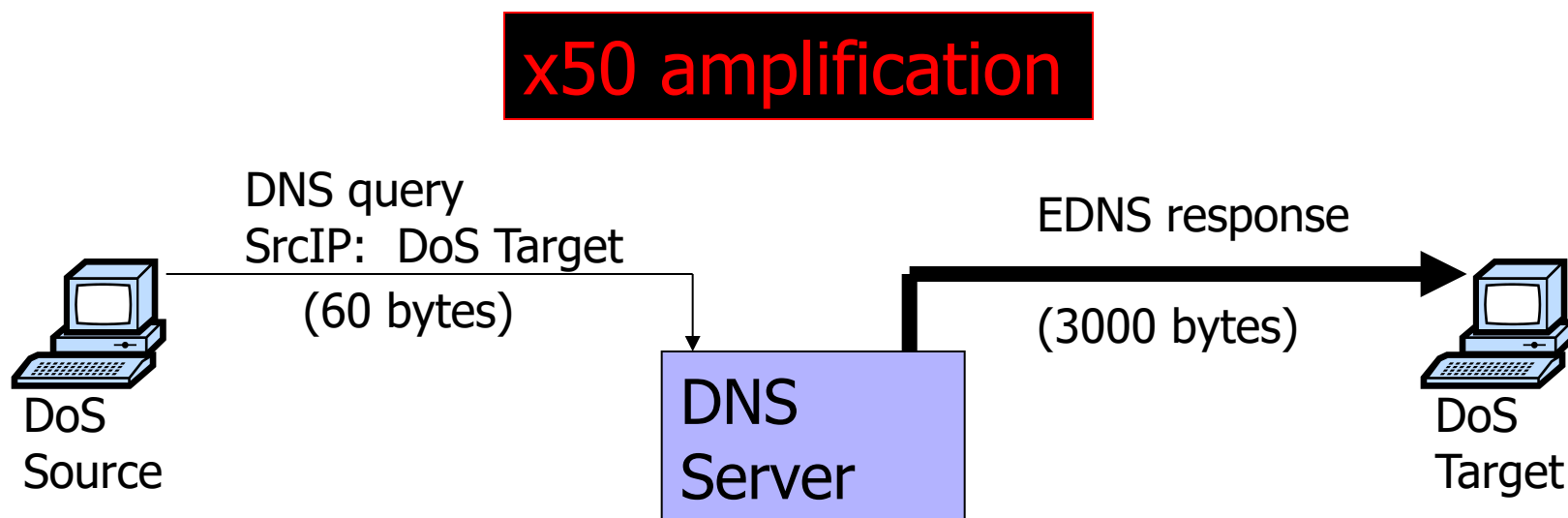
2006:   0.58M open resolvers on Internet (Kaminsky-Shiffman)

2013:   21.7M open resolvers (openresolverproject.org)

March 2013: 300 Gbps DDoS attack on Spamhaus

(Not Just DNS)

x206 amplification

"Give me the addresses of the
last 600 machines you talked to"
Spoofed SrcIP: DoS target
(234 bytes)

600 addresses

(49,000 bytes)

DoS
Source

NTP
(Network Time Protocol)
server

DoS
Target
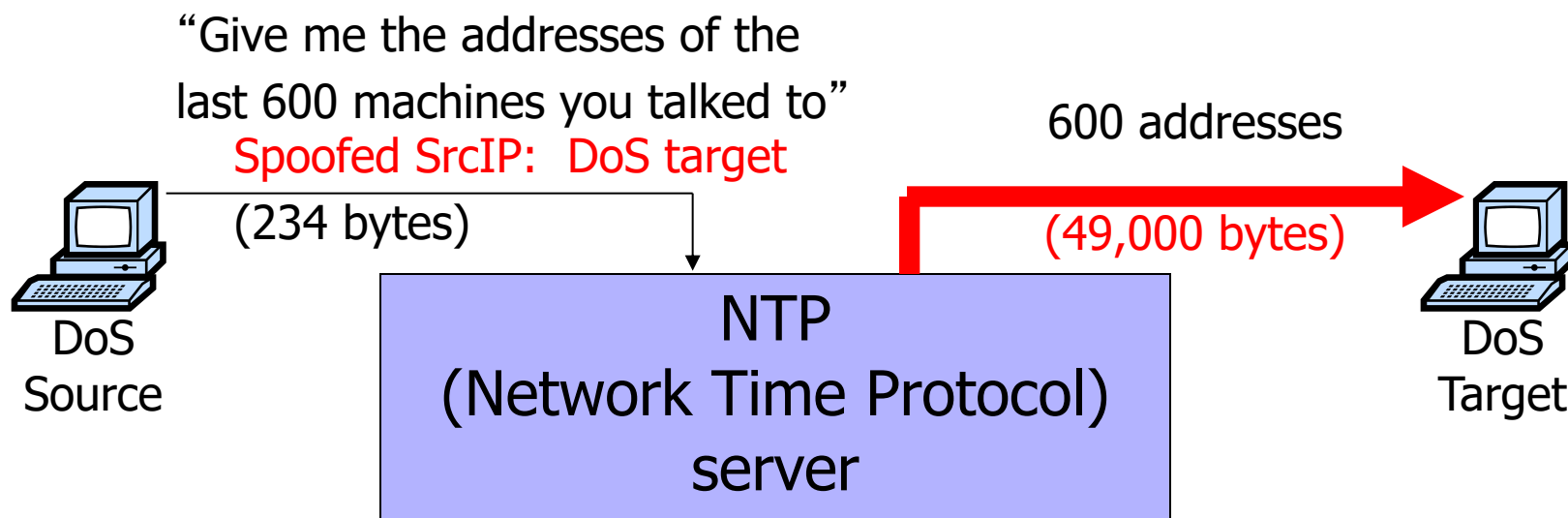
December 2013 – February 2014:

400 Gbps DDoS attacks involving 4,529 NTP servers

7 million unsecured NTP servers on the Internet (Arbor)

DNSSEC

◆ Goals: authentication and integrity of DNS requests and responses

◆ PK-DNSSEC (public key)

- DNS server signs its data – done in advance
- How do other servers learn the public key?

◆ SK-DNSSEC (symmetric key)

- Encryption and MAC: $E_k(m, MAC(m))$
- Each message contains a nonce to avoid replay
- Each DNS node shares a symmetric key with its parent
- Zone root server has a public key (hybrid approach)

Querying DNSSEC Servers

[Bernstein]

Why so big?

`3 Mbps/site`   `22 Mbps/server`   **20000 Mbps**

DNSSEC query
(78 bytes)

Client

3113-byte response

Query 94 servers
(77118 bytes total)

Spoofed source:
target's IP address

DNSSEC
Server

Server

Server

Server

2,526,996 bytes

DoS
Target
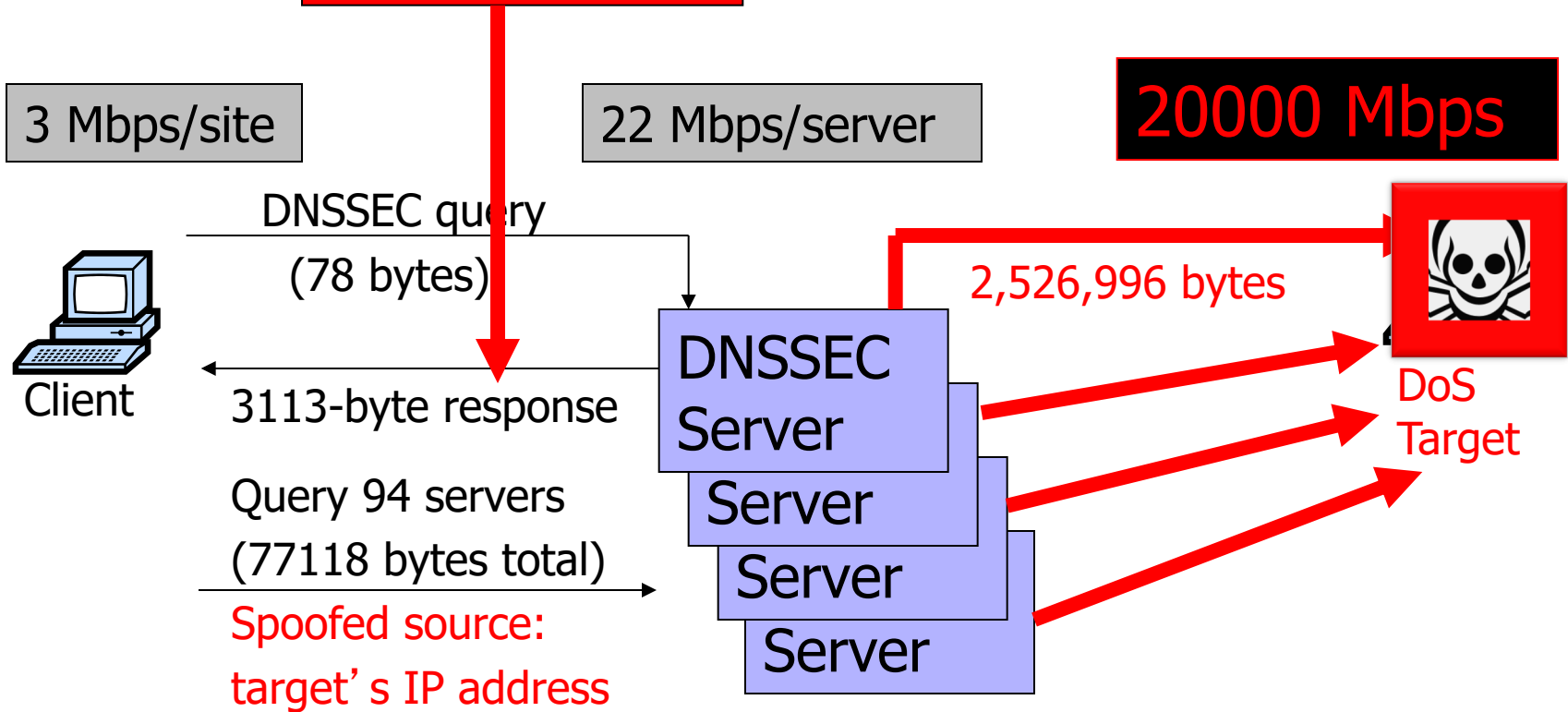
5 times per second, from 200 sites

Using DNSSEC for DDoS

◆ RFC 4033 says:

"DNSSEC provides no protection against denial of service attacks"

◆ RFC 4033 doesn't say:

"DNSSEC is a remote-controlled double-barreled shotgun, the worst DDoS amplifier on the Internet"

DNSSEC In Action

DNSSEC server
for cornell.edu

Where does
cs.cornell.edu live?

Client

Where does
zoo.cornell.edu live?

Where does
DNSSECIsTehSuck.cornell.edu live?

???

cs.cornell.edu: 128.84.96.11

math.cornell.edu: 128.84.234.110

zoo.cornell.edu: 128.84.12.95

All signed in advance
(for performance!)

Each name has exactly <u>one</u> signed record
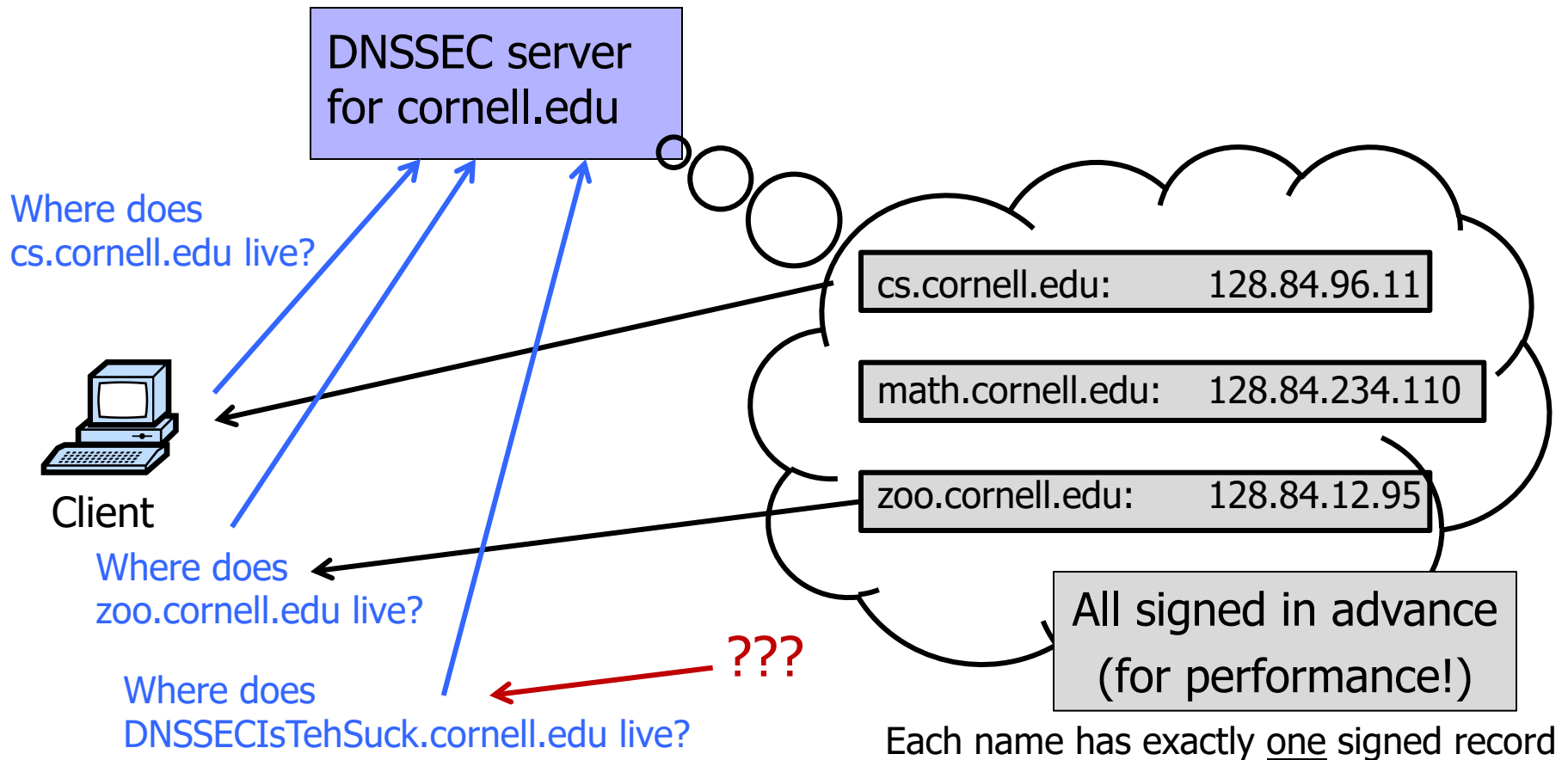
Why can't the resolver simply send an empty record
when queried for a domain that does not exist?

Authenticated Denial of Existence

NSEC

DNSSEC server for cornell.edu

Where does DNSSECIsTehSuck.cornell.edu live?

Where does TehSuckThyNameIsDNSSEC.cornell.edu live?

Client

cs.cornell.edu: 128.84.96.11

math.cornell.edu: 128.84.234.110

zoo.cornell.edu: 128.84.12.95

There are no DNSSEC subdomains of .cornell.edu between "cs" and "math"

There are no DNSSEC subdomains of .cornell.edu between "math" and "zoo"

All signed in advance (for performance!)

Use DNSSEC as an oracle to enumerate all subdomains (equivalent to zone transfer)

NSEC3

◆ Domain names hashed, hashes sorted in lexicographic order

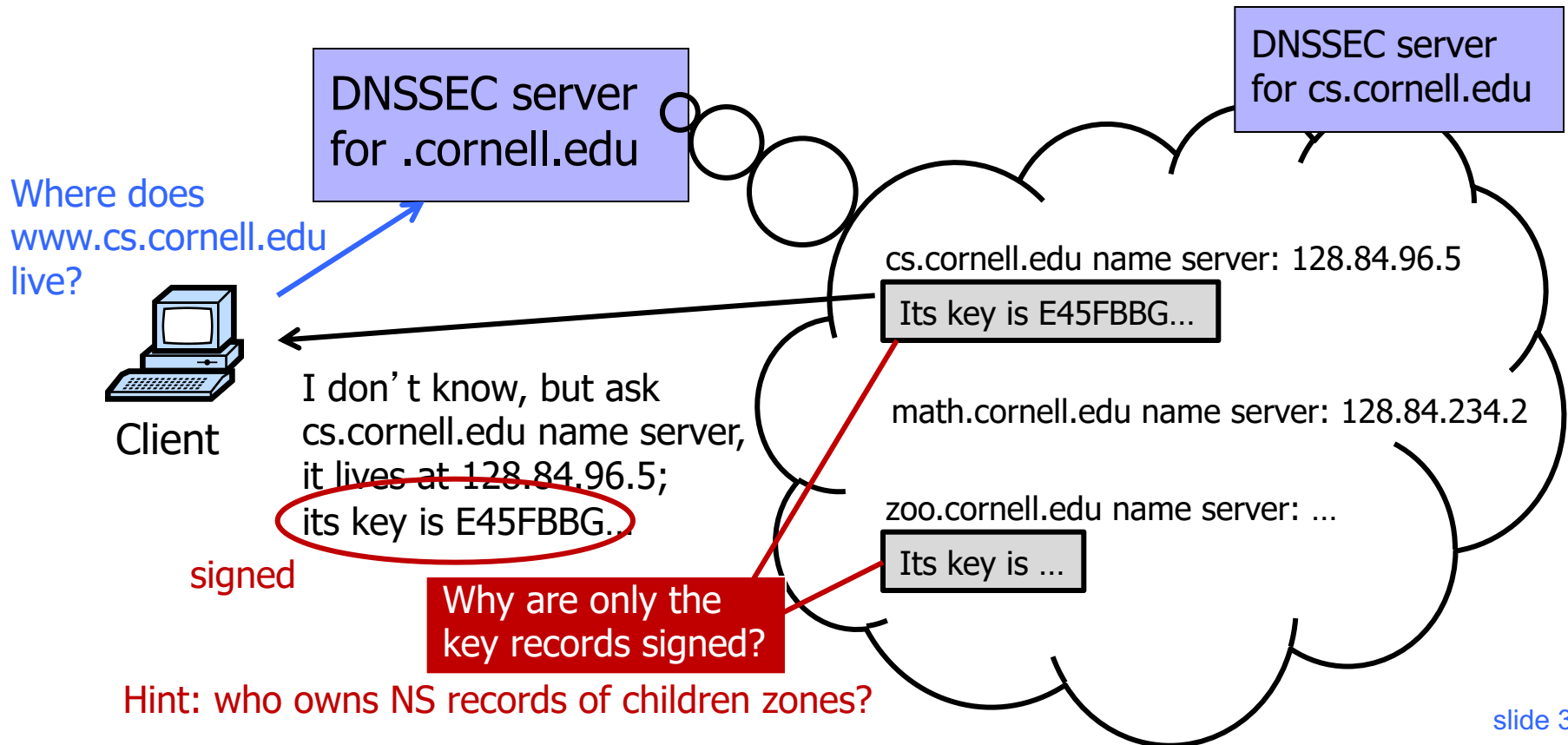◆ Denials of existence certify that there are no DNSSEC domains whose hash values fall into a certain interval

- As opposed to actual domain names

◆ Are domain names random?

◆ Vulnerable to brute-force guessing attacks

Delegation in DNSSEC

◆ Delegation is essential for scalability

- For example, there are 100,000,000 .com domains

DNSSEC server for .cornell.edu

DNSSEC server for cs.cornell.edu

Where does www.cs.cornell.edu live?

I don't know, but ask cs.cornell.edu name server, it lives at 128.84.96.5; its key is E45FBBG...

signed

Client

cs.cornell.edu name server: 128.84.96.5
Its key is E45FBBG...

math.cornell.edu name server: 128.84.234.2

zoo.cornell.edu name server: ...
Its key is ...

Why are only the key records signed?

Hint: who owns NS records of children zones?

Forging Delegation Responses

[Bernstein]

DNSSEC domains

DNSSEC server for .cornell.edu

Where does www.math.cornell.edu live?

Client

6.6.6.6

I don't know, but ask math.cornell.edu name server, it lives at 128.84.234.2

There are no DNSSEC subdomains between H("cs") and H("zoo")

signed

I don't know, but ask math.cornell.edu name server, it lives at 6.6.6.6

cs.cornell.edu name server: 128.84.96.5

Its key is E45FBBG...

math.cornell.edu name server: 128.84.234.2

zoo.cornell.edu name server: ...

Its key is ...

Non-DNSSEC domain

Signed DNSSEC response yet NS record has been forged... what happened??!!

Delegating to Secure Zones

◆ Q: When does verification of signatures on DNSSEC records actually happen?

◆ A: At the very end, when the resolver has the complete chain

◆ But the delegation record is not signed… what if it has been forged?

◆ Current DNSSEC deployments are only "secure" down to the ISP's resolver

- Stub resolvers on users' machines only get an unsigned flag saying that the response is "secure"

DNSSEC "Features"

◆ Does nothing to improve DNS availability

◆ Allows astonishing levels of DDoS amplication, damaging Internet availability

- Also CPU exhaustion attacks

◆ Does nothing to improve DNS confidentiality, leaks private DNS data (even with NSEC3)

◆ Does not prevent forgery of delegation records

◆ Does not protect the "last mile"

◆ Implementations suffered from buffer overflows