CS 5450

DNS

Vitaly Shmatikov

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

March 16, 2014

Reply Retweet Favorite More

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

RETWEETS 805 FAVORITES 156

Turkey (2014)


Engin Onder
Follow
#twitter blocked in #turkey tonight. folks are painting #google dns numbers onto the posters of the governing party.
pic.twitter.com/9vQ7NTgotO


DNS: 8.8.8.8
Alternatif: 8.8.4.4

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

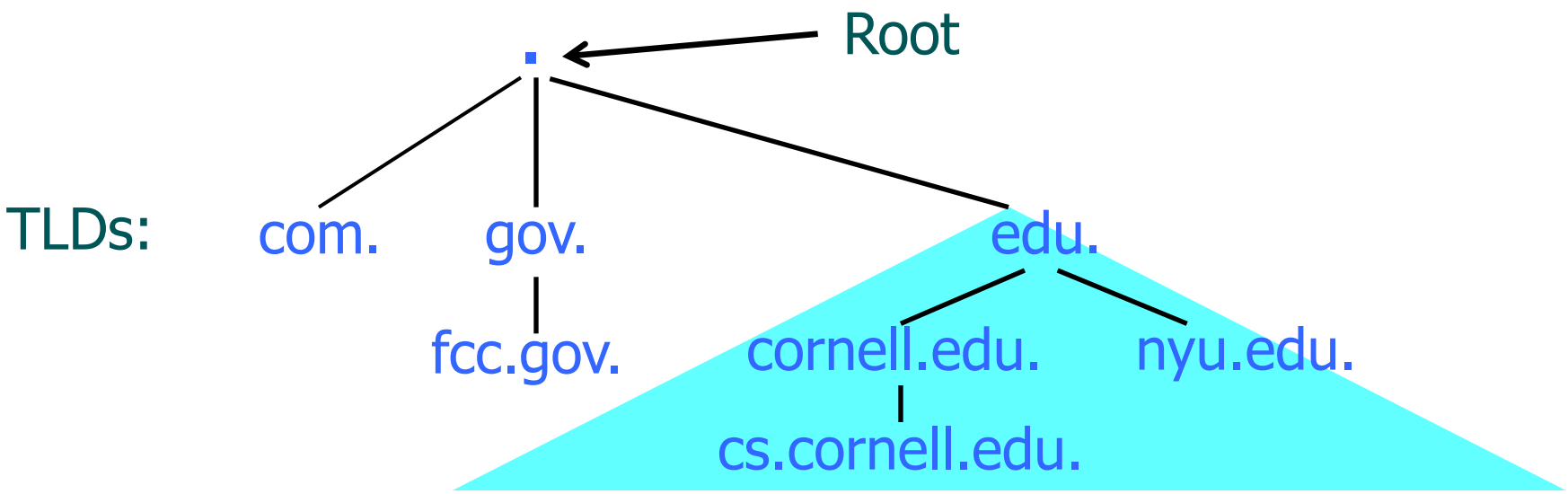
Goals of DNS

- 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 not 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



- 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

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 = NS (name server)

- name = domain (e.g. cornell.edu)
- Value = hostname of authoritative name server for this domain

Type = CNAME

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

Type = MX (mail exchange)

- name = domain
- value = name of mail server for that domain

DNS in Operation

Most queries and responses are UDP datagrams

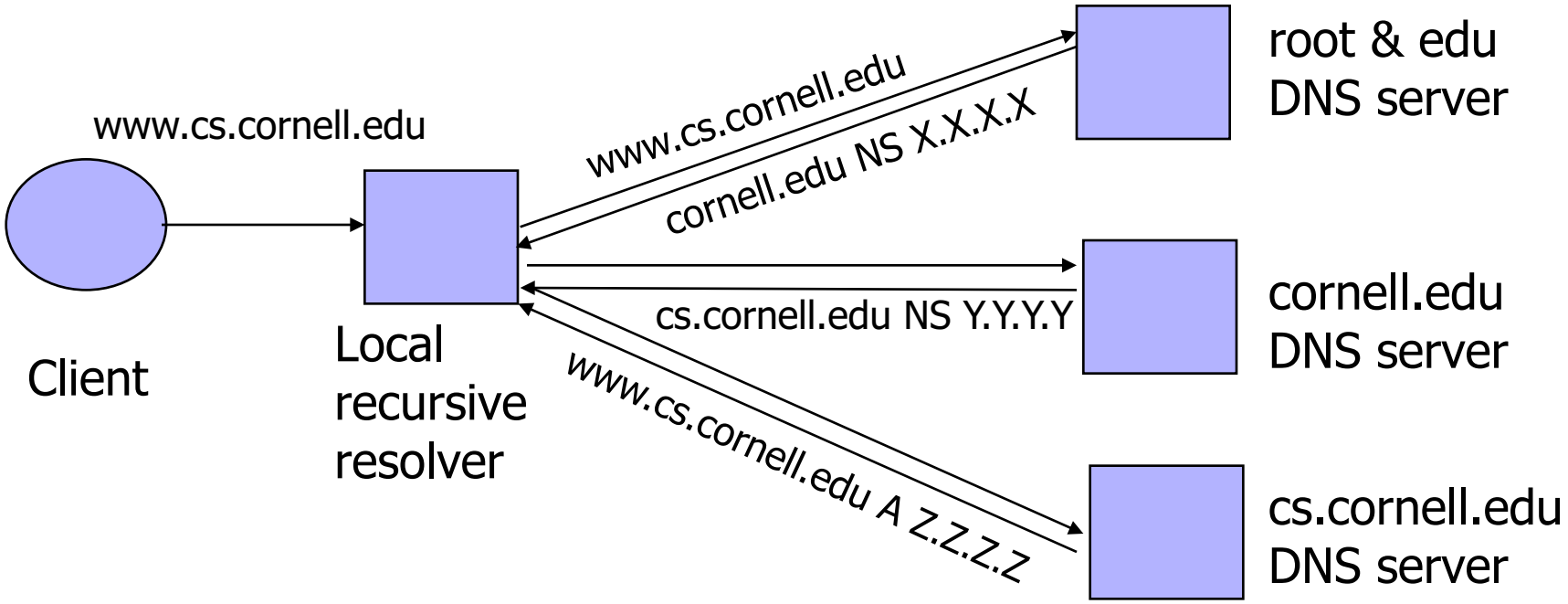
- Recursive
  - Nameserver responds with answer or error

Answer: www.cornell.edu A 132.236.207.20

- Iterative
  - Nameserver may respond with a referral

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 nameserve

```
$ 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

;; 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.
```

[Output edited for clarity]

(authoritative for freebsd.org.)

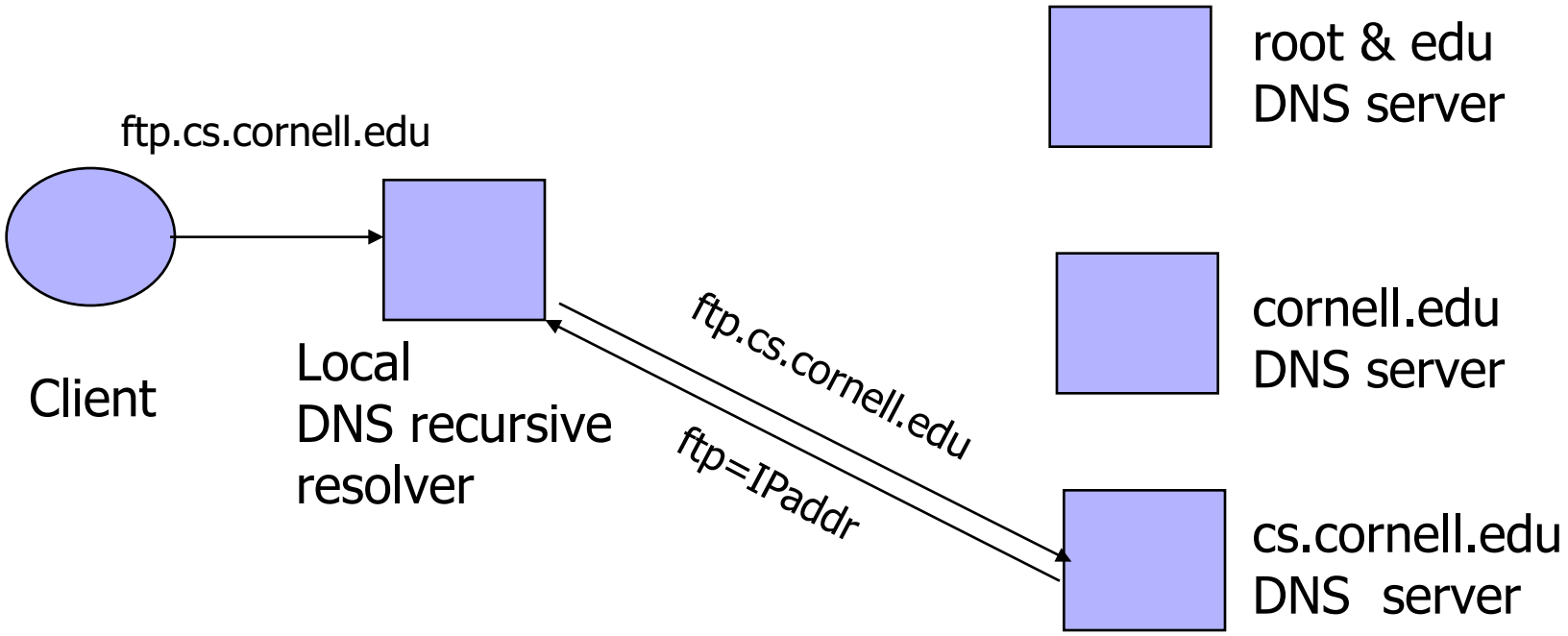
```
$ dig @ns1.isc-sns.net www.freebsd.org +norecurse
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 17037
;; QUERY: 1, ANSWER: 1, AUTHORITY: 3, ADDITIONAL: 3

;; 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
  - … before 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



What if DNS is Subverted?

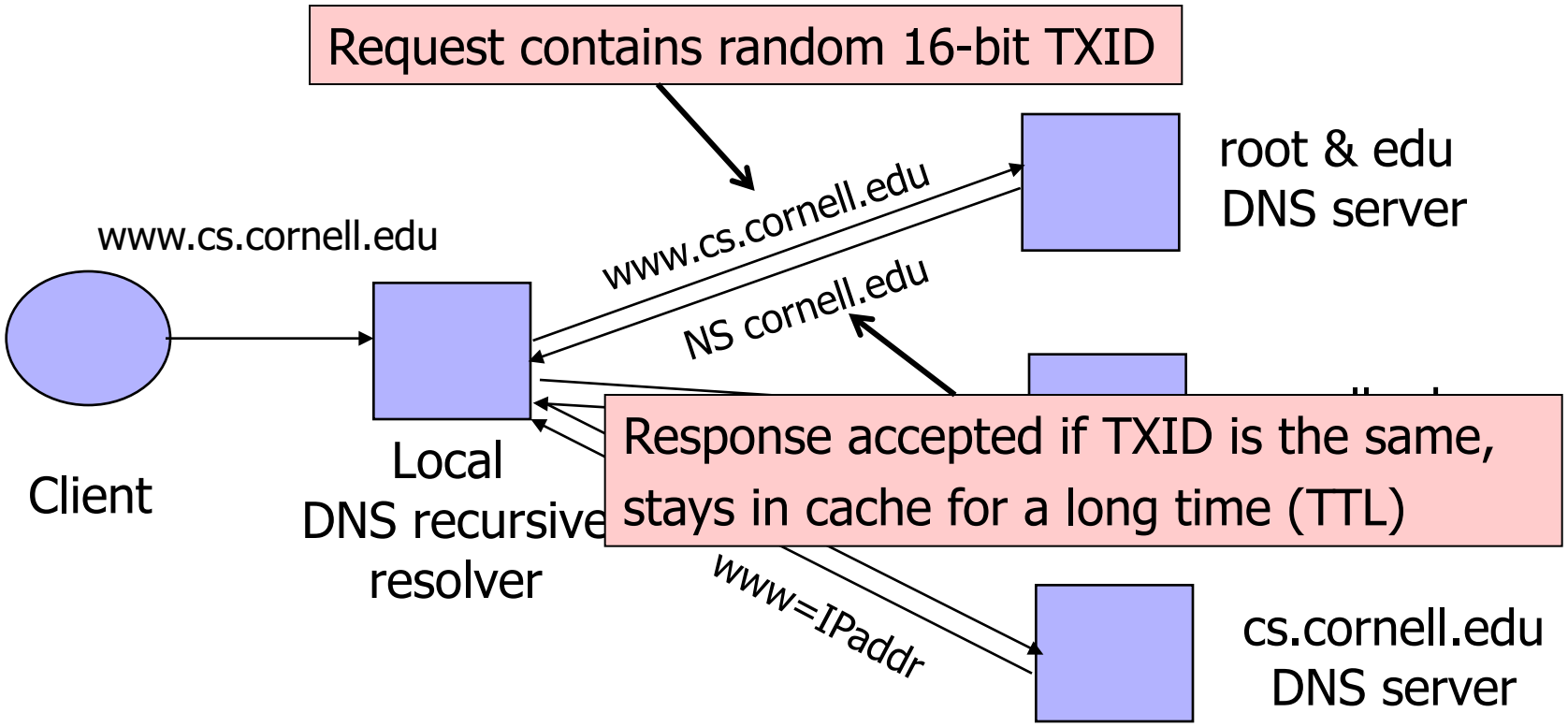
- Redirect victim's web traffic to rogue servers
- Redirect victim's email to rogue email servers (MX records in DNS)
- Does Secure Sockets Layer (SSL) provide protection?
  - Yes—user will get "wrong certificate" if SSL enabled
  - No—SSL not enabled or user ignores warnings
  - No—how is SSL trust established? Often, by email!

Problem #1: Coffee Shop

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

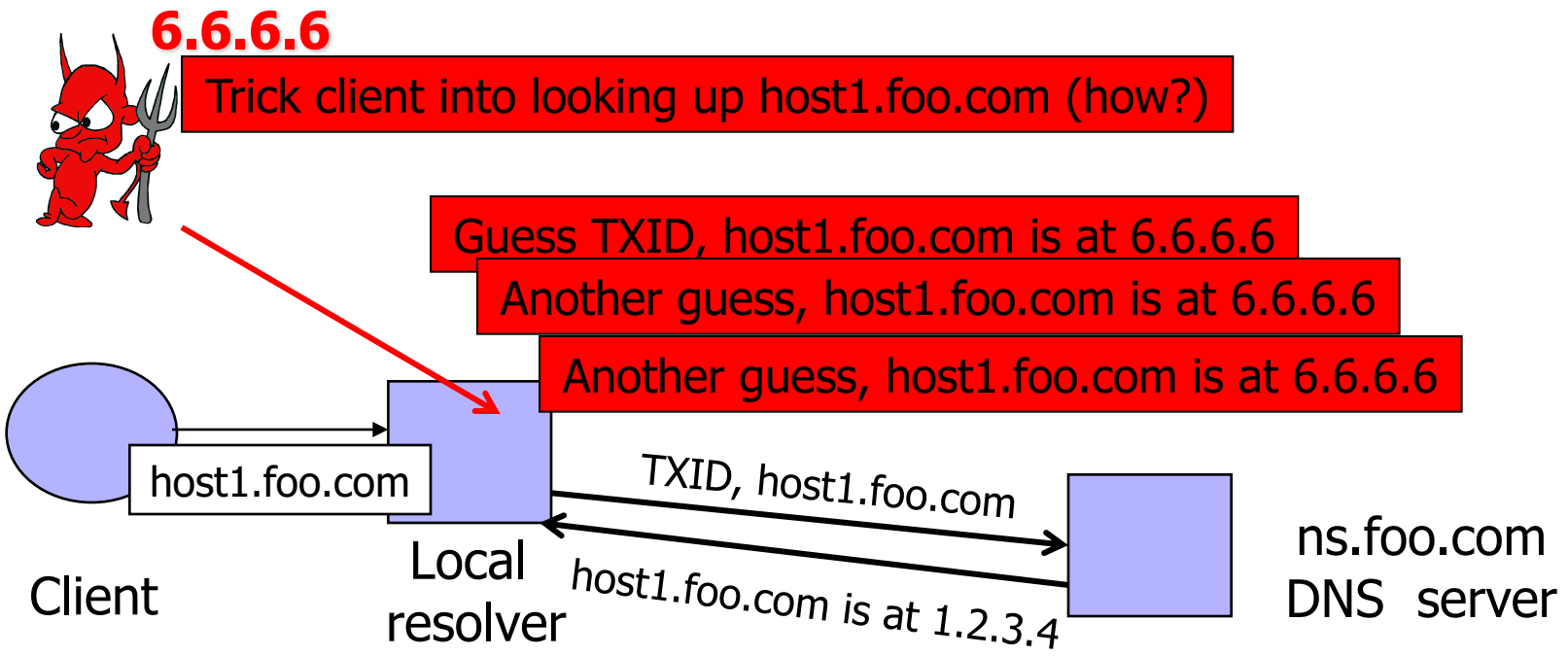
- Answer: it asks the local DNS nameserver
  - … which is run by the coffee shop or their contractor
  - … and can return to you any answer they please
- How can you know you're getting correct data?

Problem #2: DNS "Authentication"



How is this different from coffee-shop scenario?

DNS Spoofing by Off-Path Attacker



Several opportunities to win the race.

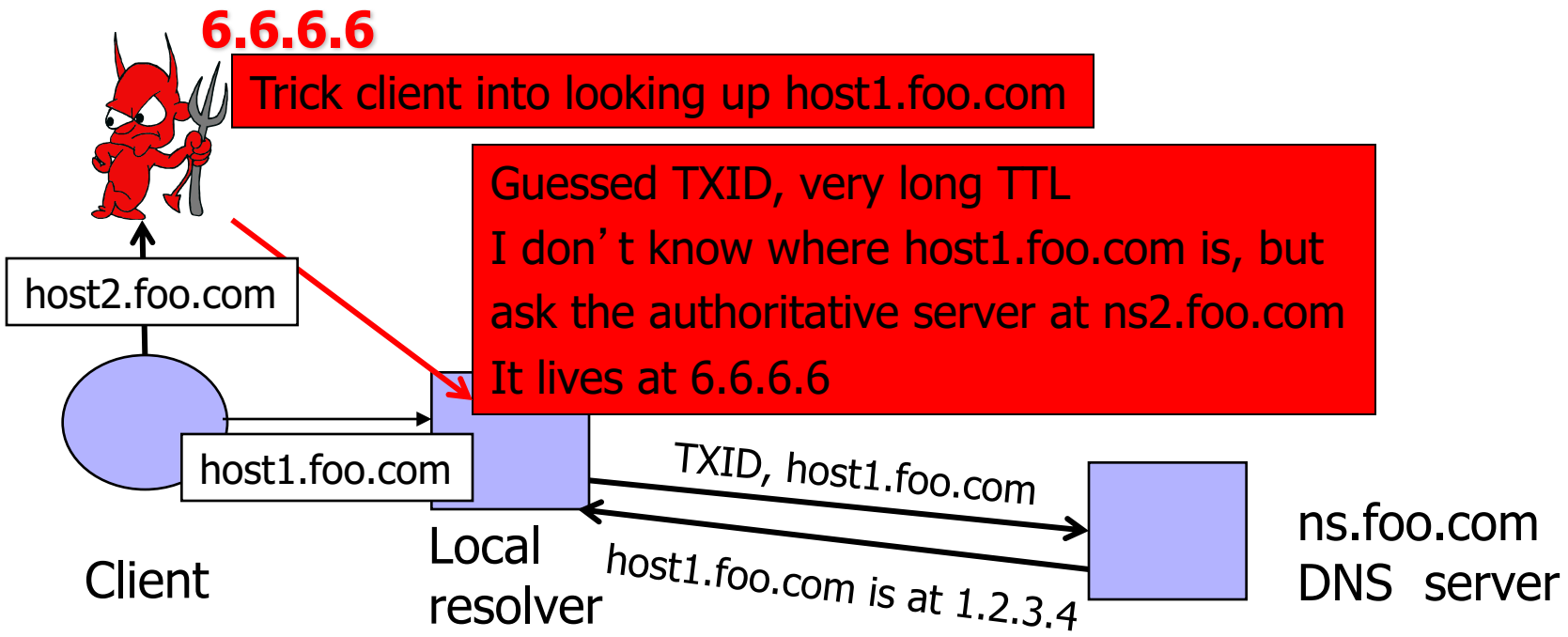
If attacker loses, has to wait until TTL expires…

… 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]



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

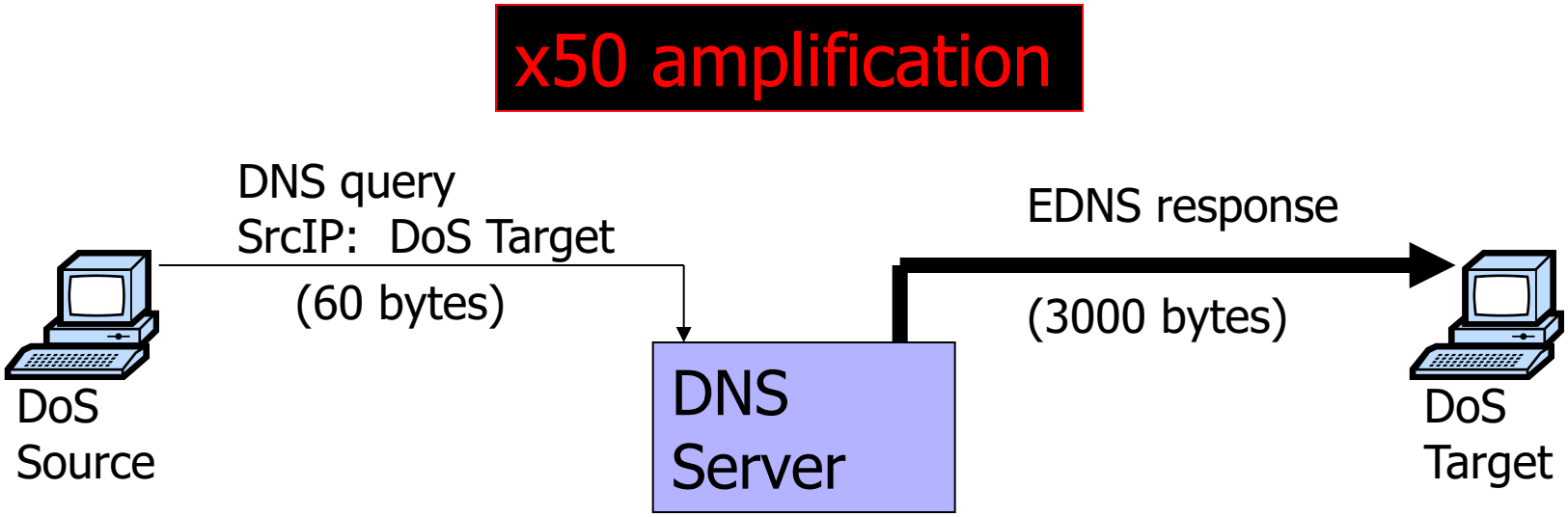
Reverse DNS Spoofing

- 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

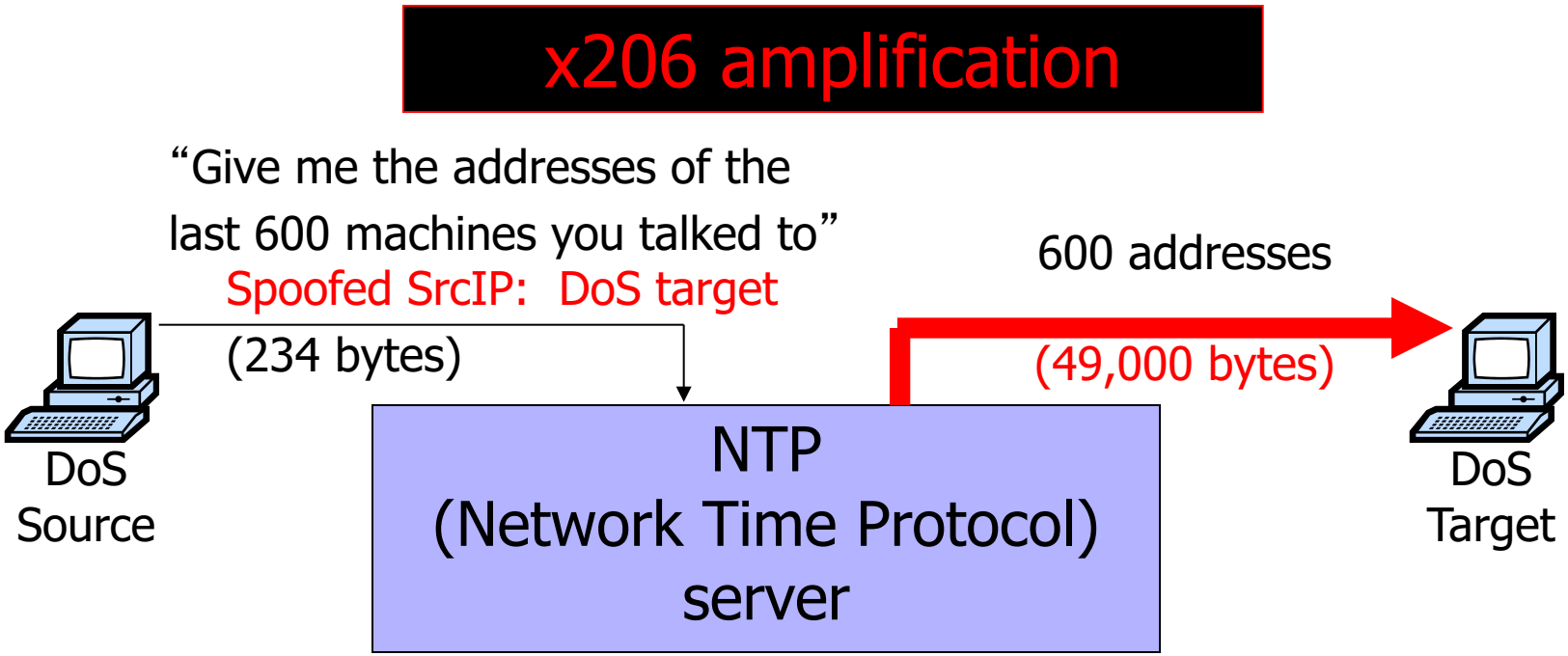


2006: 0.58M open resolvers on Internet (Kaminsky-Shiffman)

2013: 21.7M open resolvers (openresolverproject.org)

March 2013: 300 Gbps DDoS attack on Spamhaus

(Not Just DNS)



December 2013 – February 2014:

400 Gbps DDoS attacks involving 4,529 NTP servers

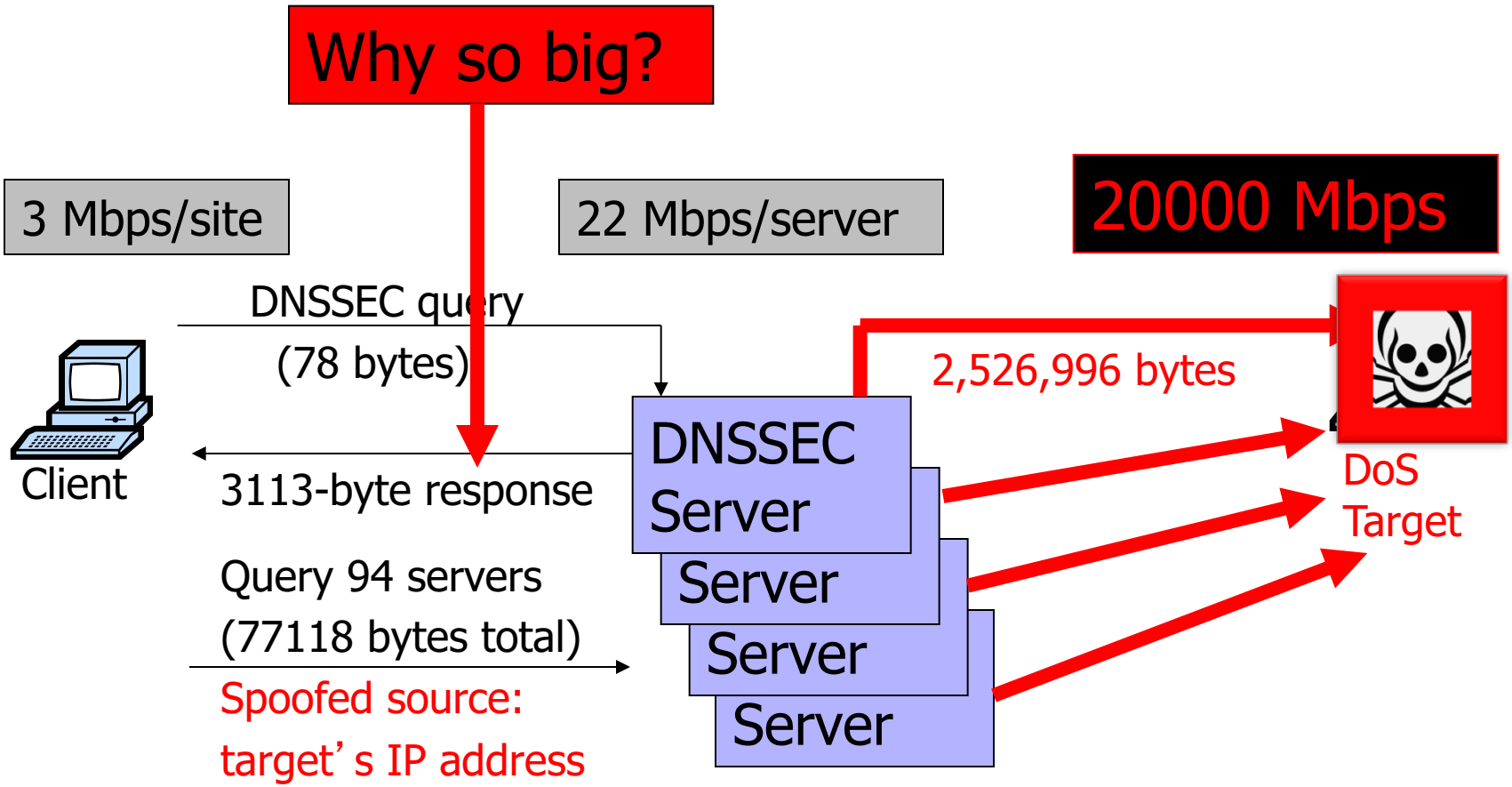
7 million unsecured NTP servers on the Internet (Arbor)

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]



5 times per second, from 200 sites

Using DNSSEC for DDoS

[Bernstein]

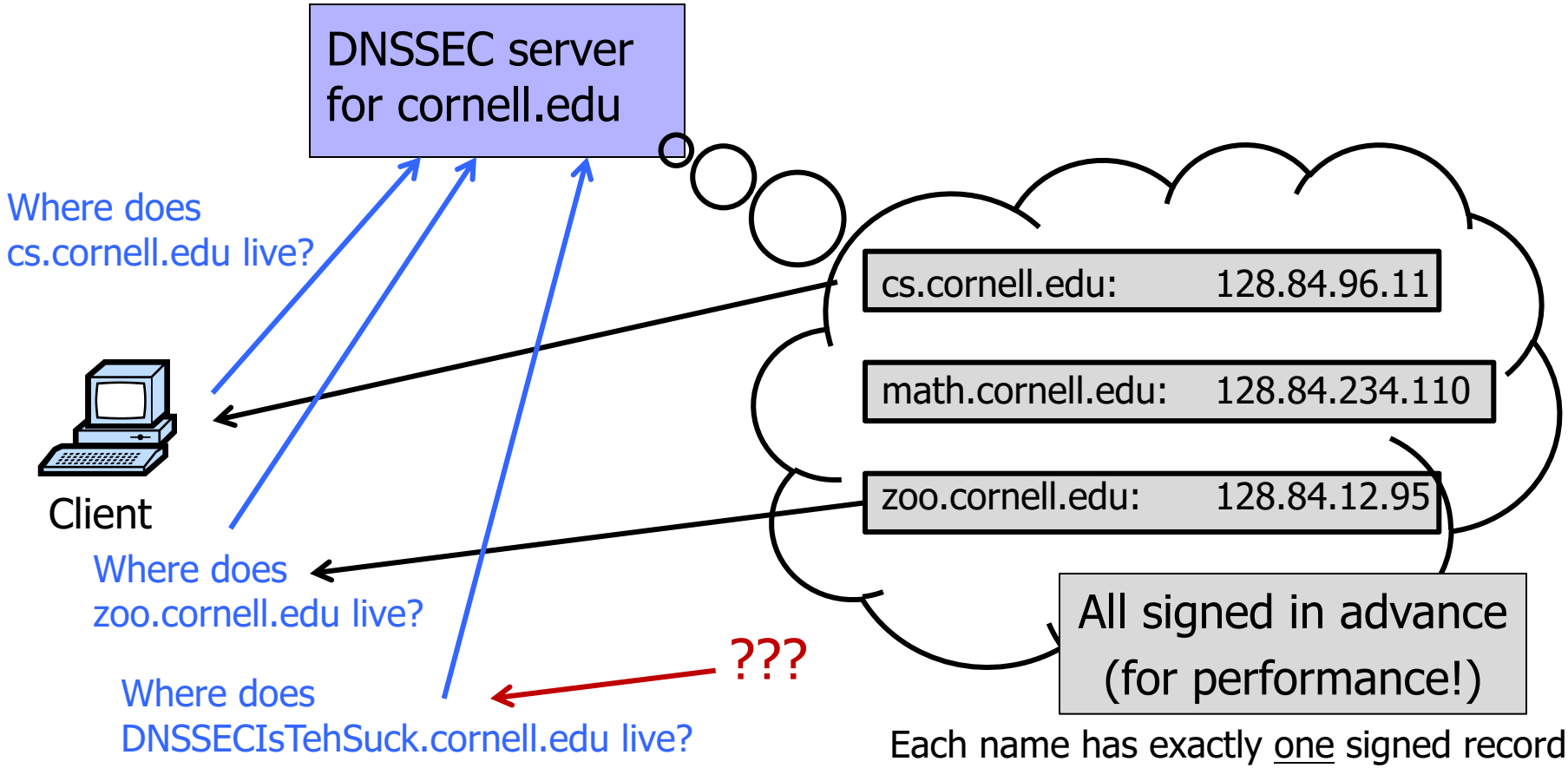
- RFC 4033 says:

"DNSSEC provides no protection against denial of service attacks"

- RFC 4033 doesn't say:

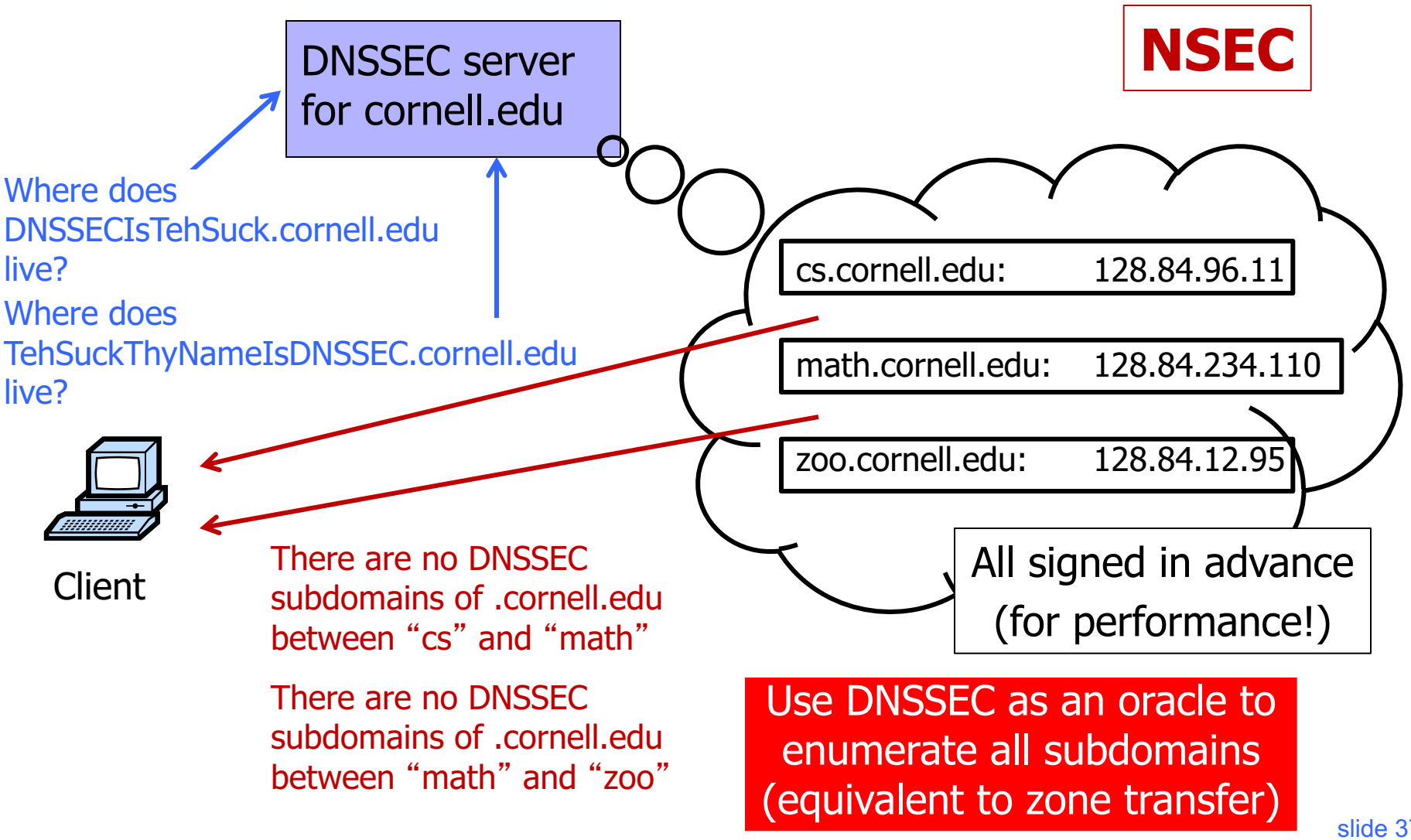
"DNSSEC is a remote-controlled double-barreled shotgun, the worst DDoS amplifier on the Internet"

DNSSEC In Action



Why can't the resolver simply send an empty record when queried for a domain that does not exist?

Authenticated Denial of Existence



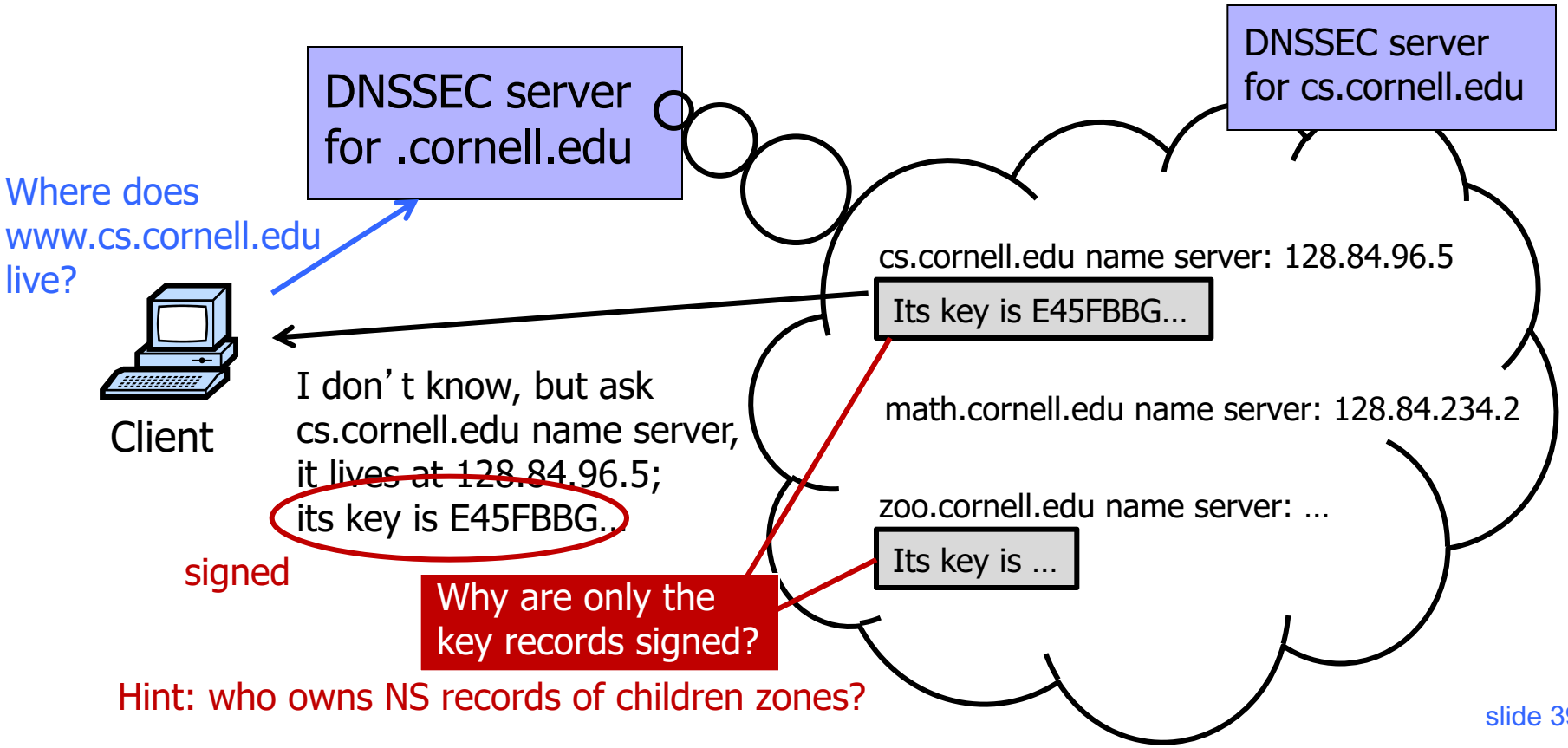
NSEC3

[Bernstein]

- 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

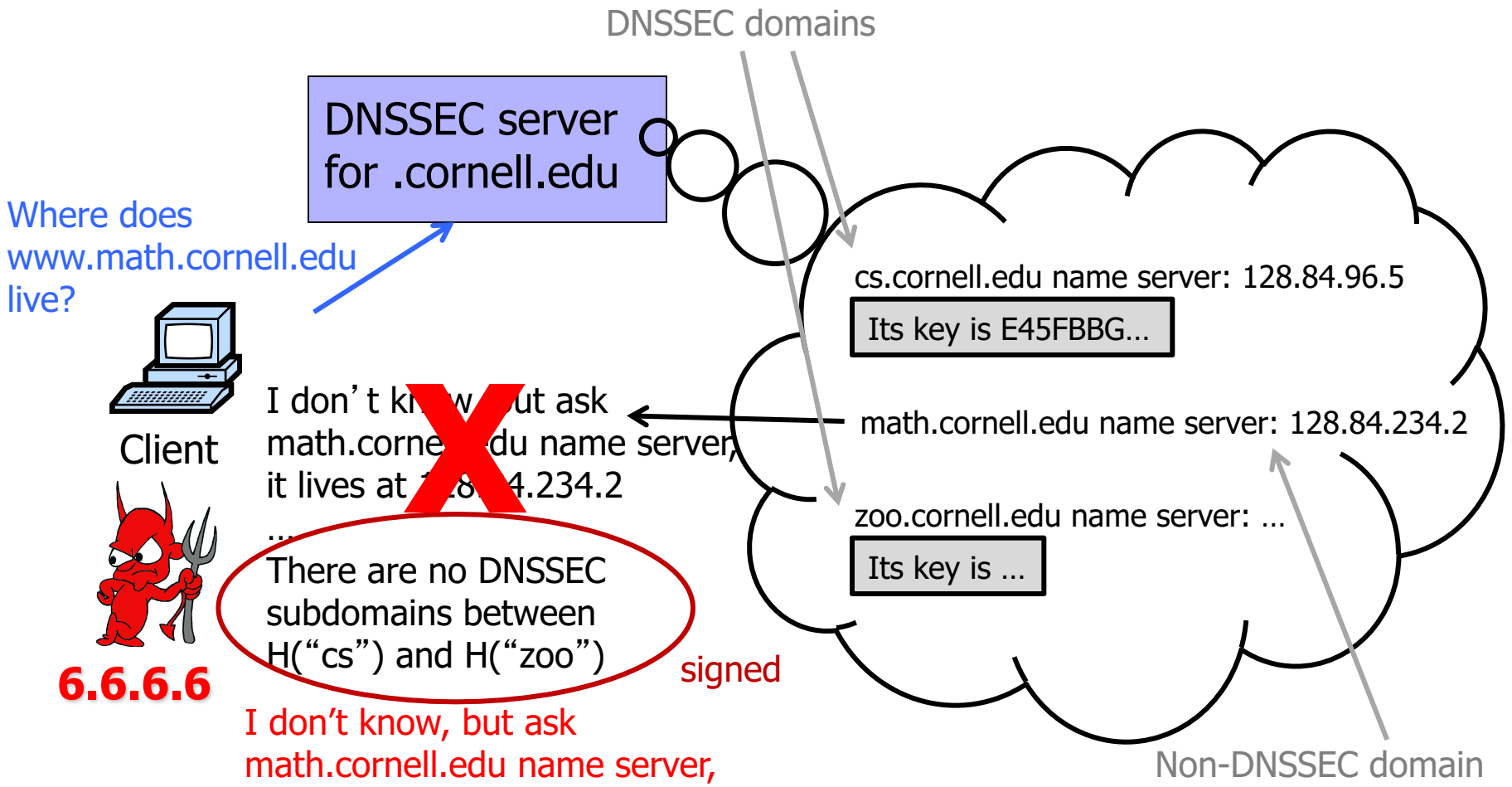
Delegation in DNSSEC

- Delegation is essential for scalability
  - For example, there are 100,000,000 .com domains



Forging Delegation Responses

[Bernstein]



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"

[Bernstein]

- 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