

CS6903: Network Security

DNSSEC

Antony Franklin

Based on Slides by D. Choffnes (NEU). Revised by P. Gill Fall 2014. Some content on DNS censorship from N. Weaver.

- ❑ DNS Basics
- ❑ DNS Security
- ❑ DNS and Censorship

Naming in Internet

4

- If you want to...
 - ▣ Call someone, you need to ask for their phone number
 - You can't just dial "M R B O B"
 - ▣ Mail someone, you need to get their address first
- What about the Internet?
 - ▣ If you need to reach Google, you need their IP
 - ▣ Does anyone know Google's IP?
- Problem:
 - ▣ People can't remember IP addresses
 - ▣ Need human readable names that map to IPs

Internet Names and Addresses

5

- ❑ Addresses, e.g. 129.10.117.100
 - ❑ Computer usable labels for machines
 - ❑ Conform to structure of the network
- ❑ Names, e.g. www.iith.ac.in
 - ❑ Human usable labels for machines
 - ❑ Conform to organizational structure
- ❑ How do you map from one to the other?
 - ❑ Domain Name System (DNS)

History

6

- ❑ Before DNS, all mappings were in *hosts.txt*
 - ❑ */etc/hosts* on Linux
 - ❑ *C:\Windows\System32\drivers\etc\hosts* on Windows
- ❑ Centralized, manual system
 - ❑ Changes were submitted to SRI (Stanford Research Institute) via email
 - ❑ Machines periodically FTP new copies of *hosts.txt*
 - ❑ Administrators could pick names at their discretion
 - ❑ Any name was allowed
 - *alans_server_at_sbu_pwns_joo_lol_kthxbye*

Towards DNS

7

- Eventually, the *hosts.txt* system fell apart
 - ▣ Not scalable, SRI couldn't handle the load
 - ▣ Hard to enforce uniqueness of names
 - e.g MIT
 - ▣ Massachusetts Institute of Technology?
 - ▣ Melbourne Institute of Technology?
 - ▣ Many machines had inaccurate copies of *hosts.txt*
- Thus, DNS was born

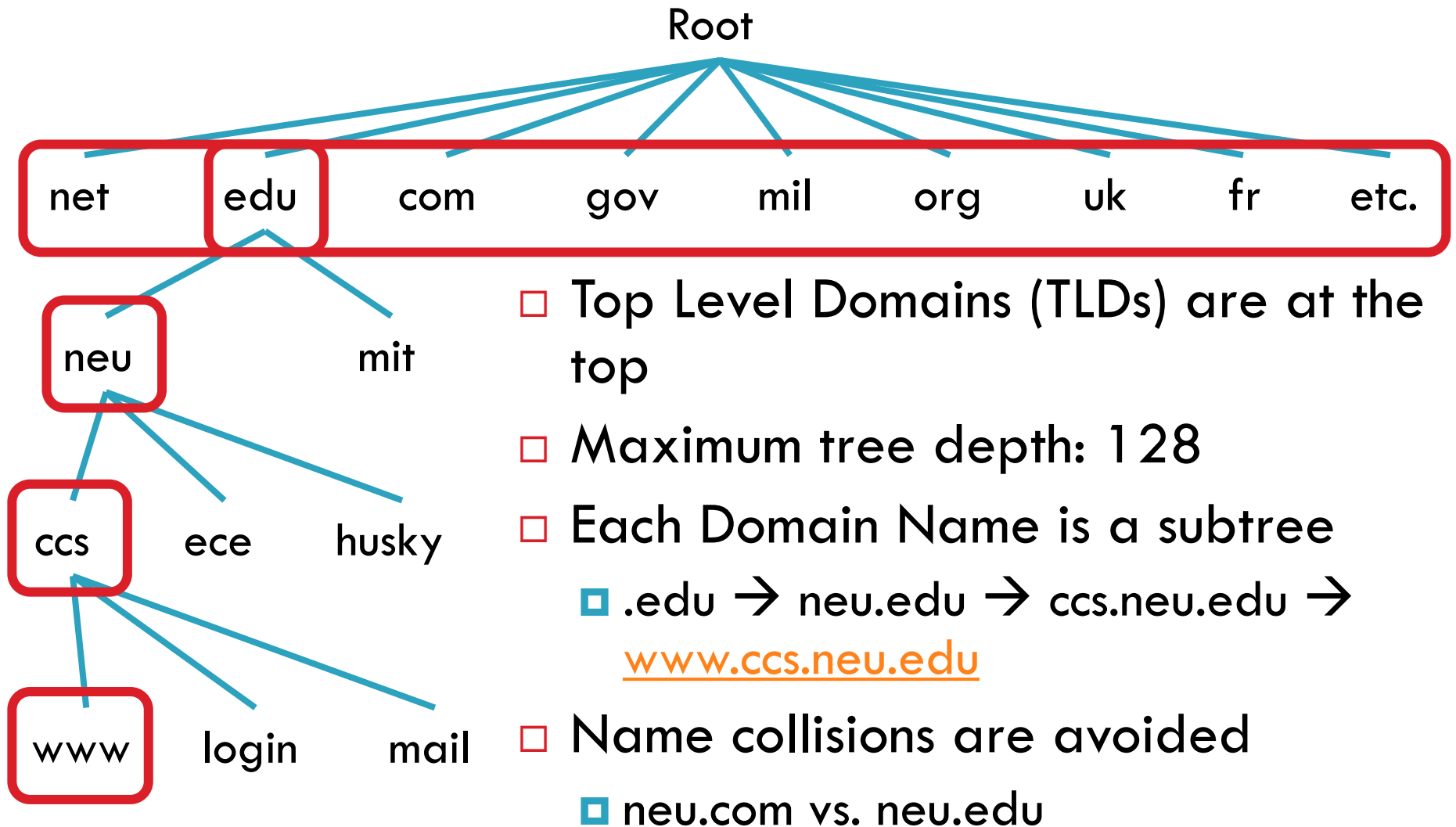
DNS at a High-Level

8

- ❑ Domain Name System
- ❑ Distributed database
 - ▣ No centralization
- ❑ Simple client/server architecture
 - ▣ UDP port 53, some implementations also use TCP
 - ▣ Why?
- ❑ Hierarchical namespace
 - ▣ As opposed to original, flat namespace
 - ▣ e.g. .com → google.com → mail.google.com

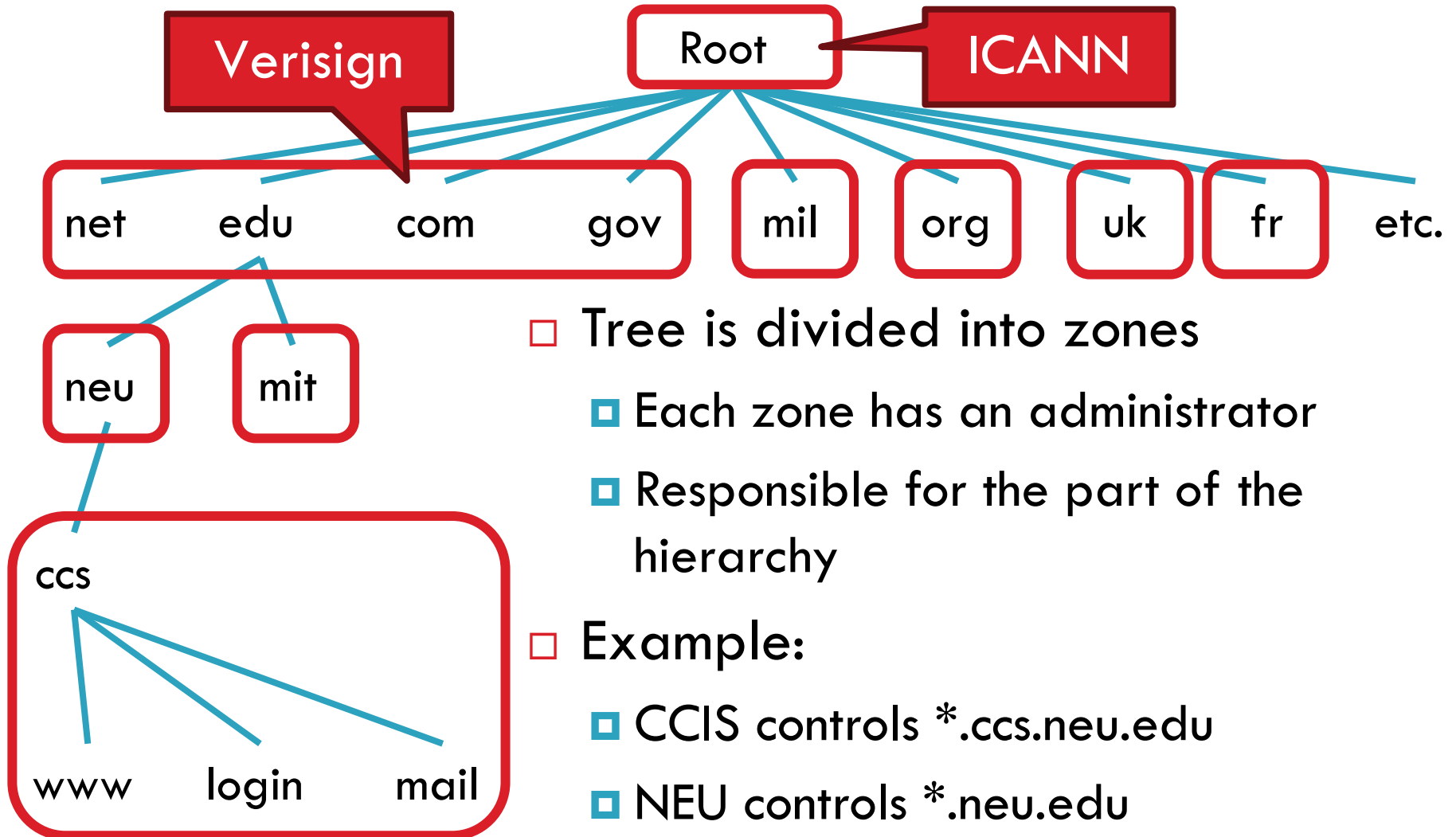
Naming Hierarchy

9



Hierarchical Administration

10



Server Hierarchy

11

- ❑ Functions of each DNS server:
 - ▣ Authority over a portion of the hierarchy
 - No need to store all DNS names
 - ▣ Store all the records for hosts/domains in its zone
 - May be replicated for robustness
 - ▣ Know the addresses of the root servers
 - Resolve queries for unknown names
- ❑ Root servers know about all TLDs
 - ▣ The buck stops at the root servers

Root Name Servers

12

- ❑ Responsible for the Root Zone File

- ▣ Lists the TLDs and who controls them
- ▣ ~272KB in size

com.	172800	IN	NS	a.gtld-servers.net.
com.	172800	IN	NS	b.gtld-servers.net.
com.	172800	IN	NS	c.gtld-servers.net.

- ❑ Administered by ICANN

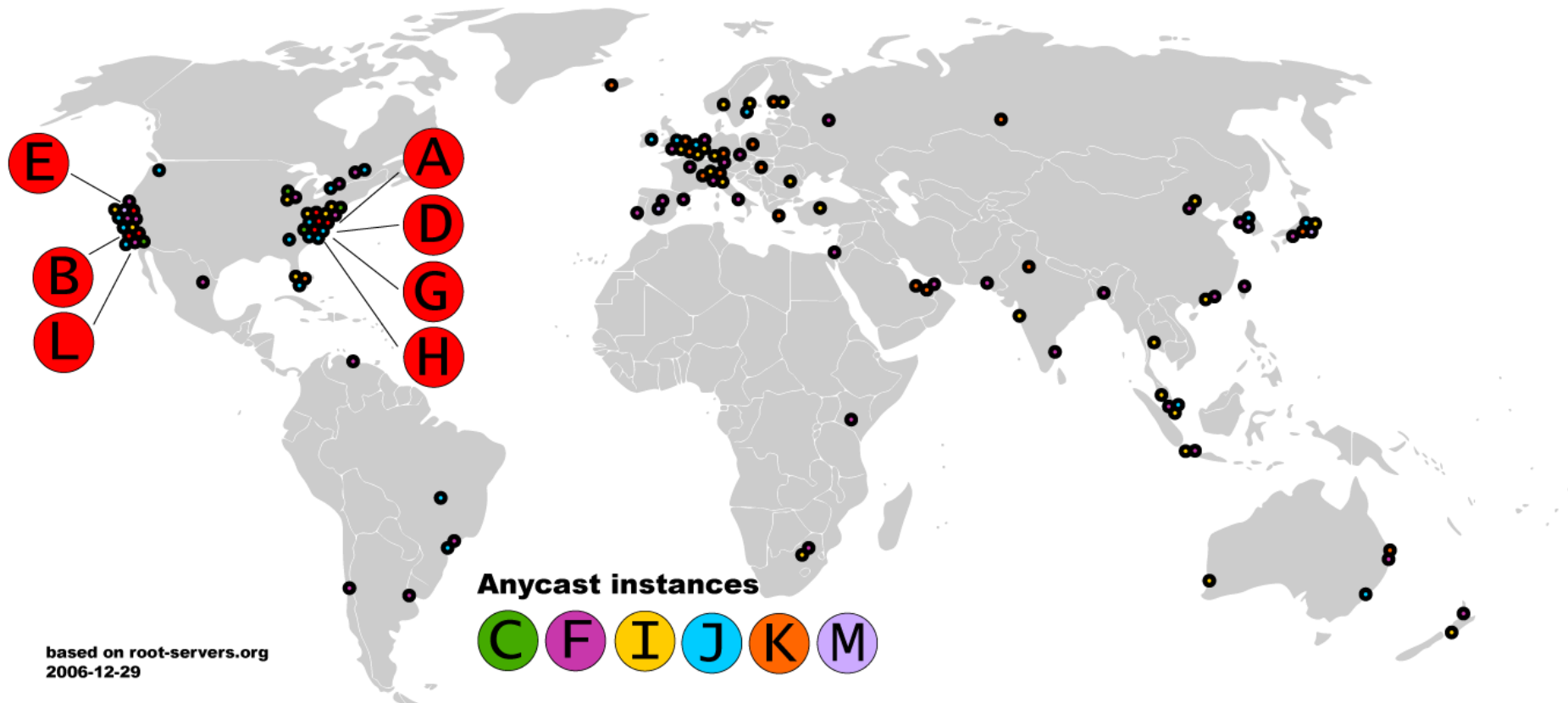
- ▣ 13 root servers, labeled A→M
- ▣ 6 are anycasted, i.e. they are globally replicated

- ❑ Contacted when names cannot be resolved

- ▣ In practice, most systems cache this information

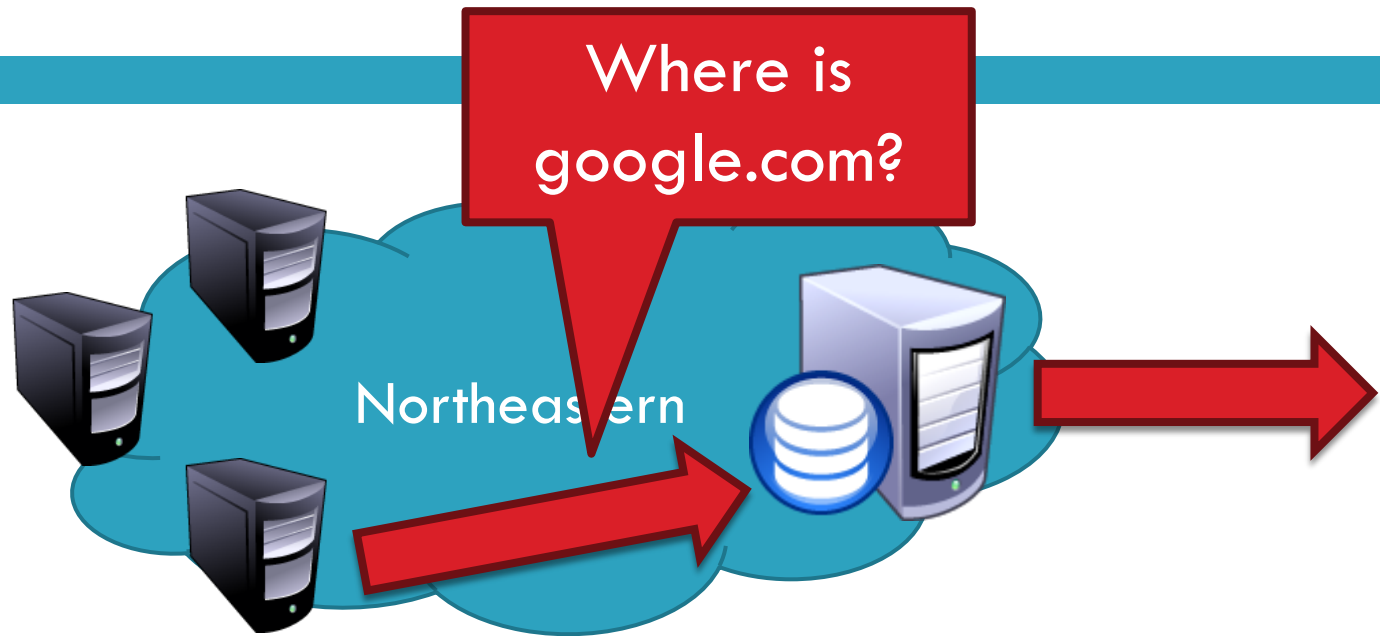
Map of the Roots

13



Local Name Servers

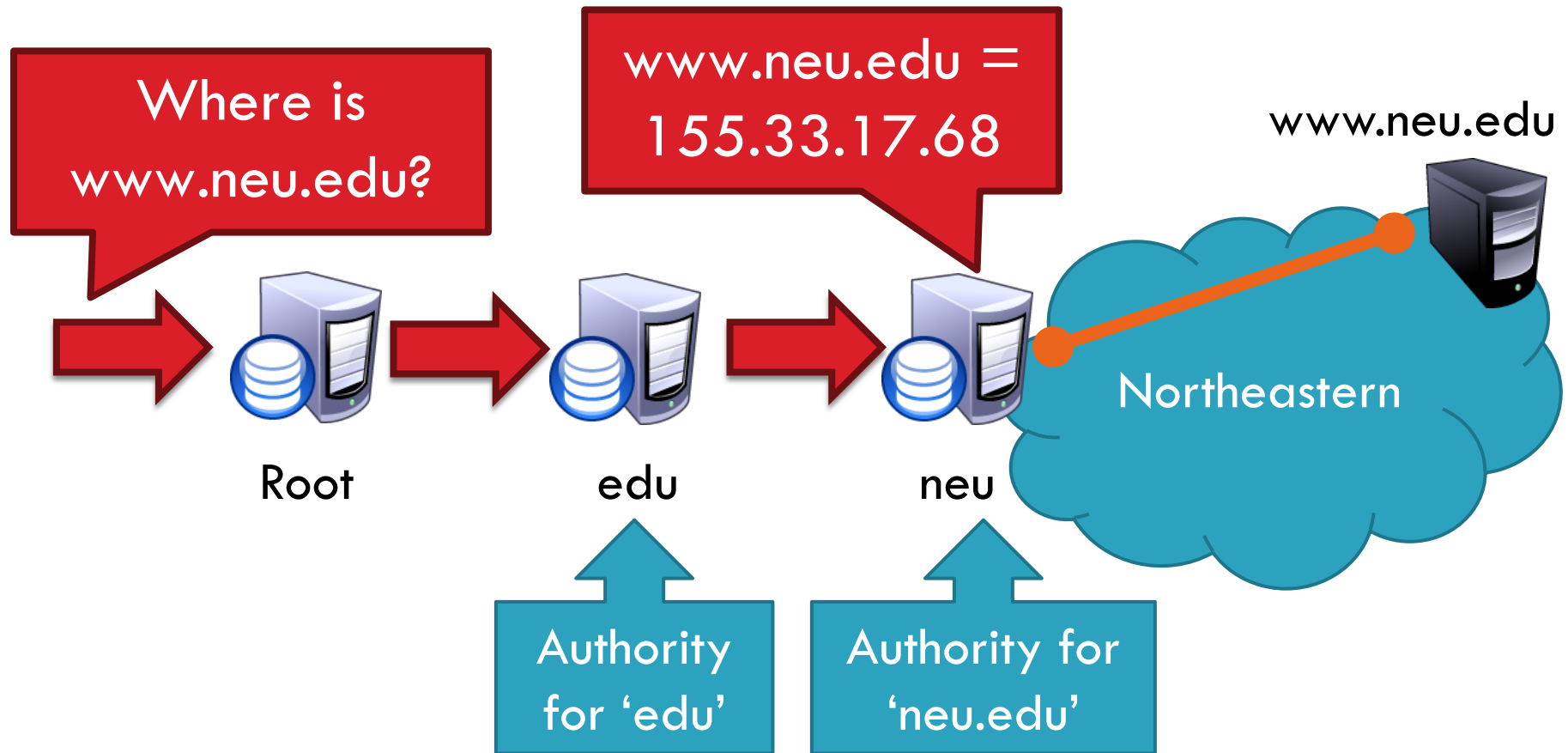
14



- ❑ Each ISP/company has a local, default name server
- ❑ Often configured via DHCP
- ❑ Hosts begin DNS queries by contacting the local name server
- ❑ Frequently cache query results

Authoritative Name Servers

15



- Stores the name \rightarrow IP mapping for a given host

Basic Domain Name Resolution

16

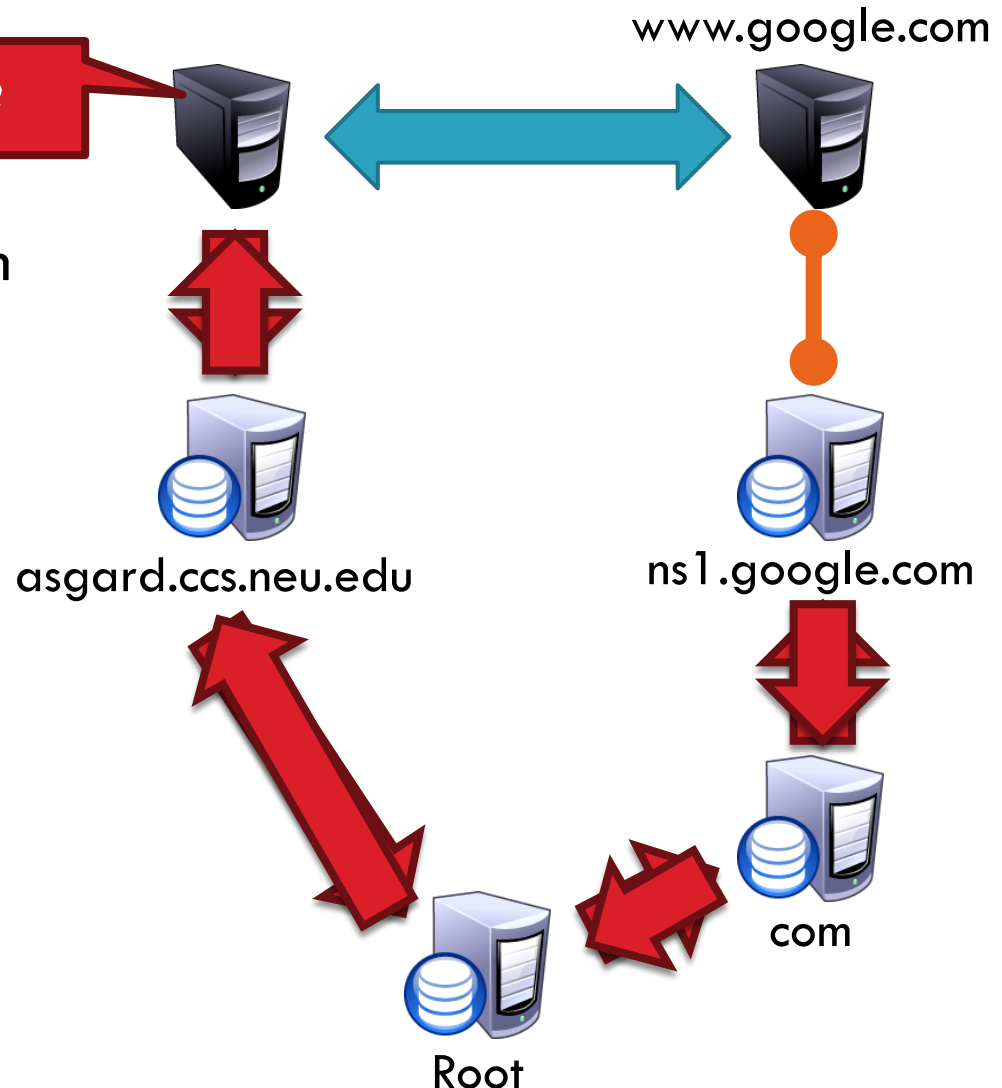
- ❑ Every host knows a local DNS server
 - ▣ Sends all queries to the local DNS server
- ❑ If the local DNS can answer the query, then you're done
 1. Local server is also the authoritative server for that name
 2. Local server has cached the record for that name
- ❑ Otherwise, go down the hierarchy and search for the authoritative name server
 - ▣ Every local DNS server knows the root servers
 - ▣ Use cache to skip steps if possible
 - e.g. skip the root and go directly to .edu if the root file is cached

Recursive DNS Query

17

Where is www.google.com?

- ❑ Puts the burden of resolution on the contacted name server
- ❑ How does asgard know who to forward responses to?
 - ▣ Random IDs embedded in DNS queries

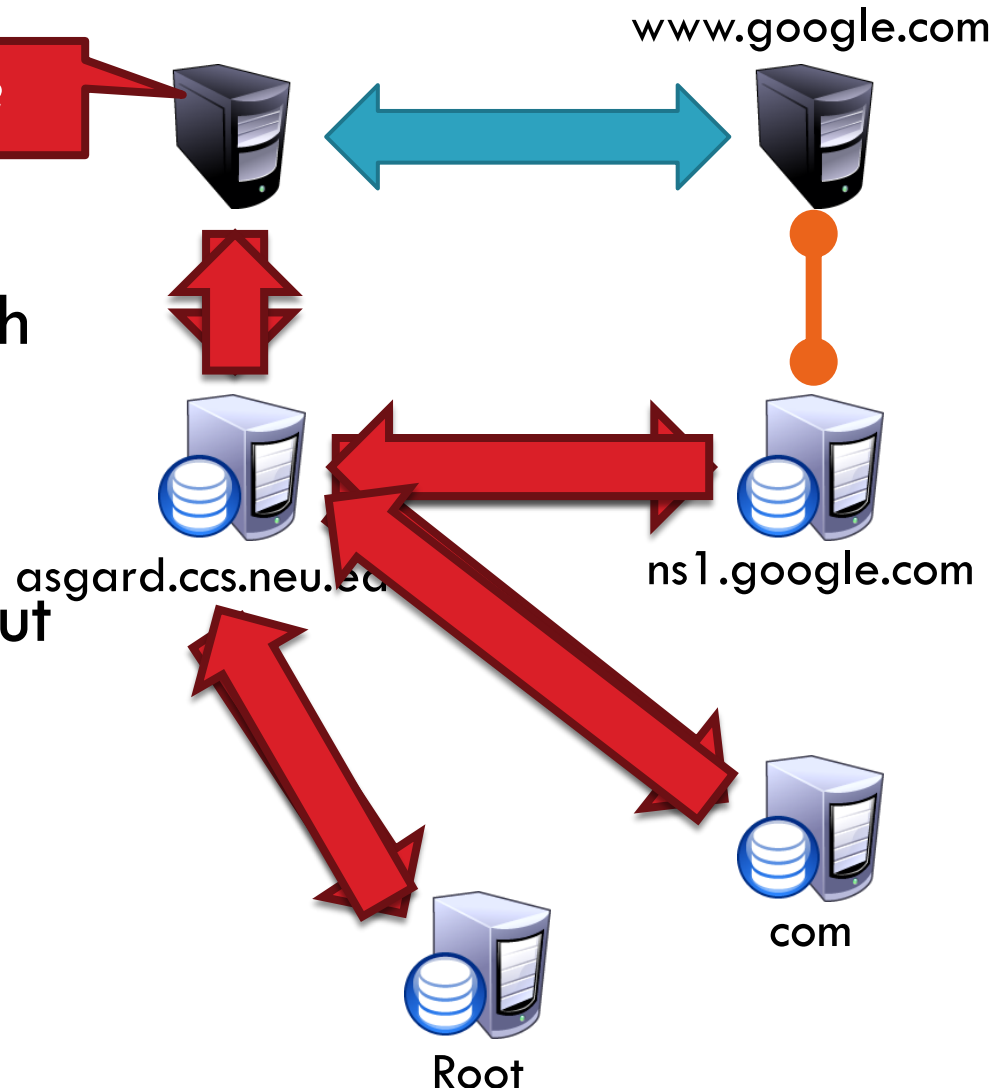


Iterated DNS query

18

Where is www.google.com?

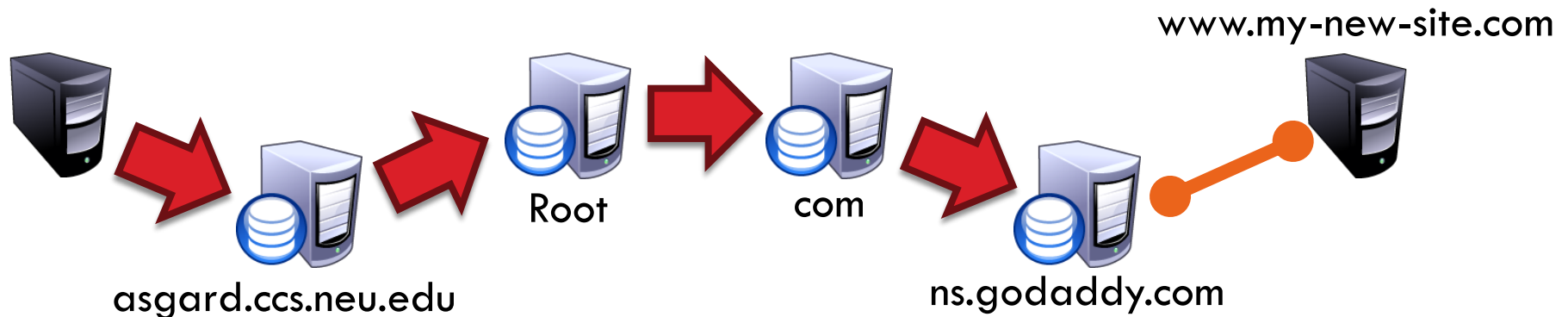
- Contact server replies with the name of the next authority in the hierarchy
- “I don’t know this name, but this other server might”
- This is how DNS works today



DNS Propagation

19

- ❑ How many of you have purchased a domain name?
 - ▣ Did you notice that it took ~72 hours for your name to become accessible?
 - ▣ This delay is called DNS Propagation



- ❑ Why would this process fail for a new DNS name?

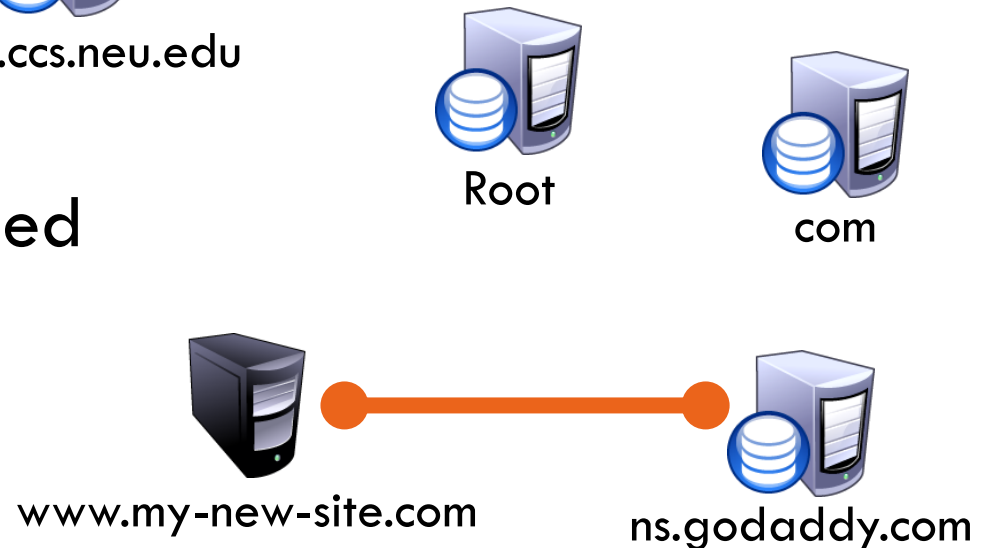
Caching vs. Freshness

20

- DNS Propagation delay is caused by caching



- Zone files may be cached for 1-72 hours



DNS Resource Records

21

- ❑ DNS queries have two fields: **name** and **type**
- ❑ Resource record is the response to a query
 - ❑ Four fields: (**name**, **value**, **type**, TTL)
 - ❑ There may be multiple records returned for one query
- ❑ What do the **name** and **value** mean?
 - ❑ Depends on the **type** of query and response

DNS Types

22

□ Type = A / AAAA

- ▣ Name = domain name
- ▣ Value = IP address
- ▣ A is IPv4, AAAA is IPv6

Query

Name: www.ccs.neu.edu
Type: A

Resp.

Name: www.ccs.neu.edu
Value: 129.10.116.81

□ Type = NS

- ▣ Name = partial domain
- ▣ Value = name of DNS server for this domain
- ▣ “Go send your query to this other server”

Query

Name: ccs.neu.edu
Type: NS

Resp.

Name: ccs.neu.edu
Value: neu.edu

DNS Types, Continued

23

□ Type = CNAME

- ▣ Name = hostname
- ▣ Value = canonical hostname
- ▣ Useful for aliasing
- ▣ CDNs use this

Query

Name: foo.mysite.com
Type: CNAME

Resp.

Name: foo.mysite.com
Value: bar.mysite.com

□ Type = MX

- ▣ Name = domain in email address
- ▣ Value = canonical name of mail server

Query

Name: ccs.neu.edu
Type: MX

Resp.

Name: ccs.neu.edu
Value: amber.ccs.neu.edu

Reverse Lookups

24

- ❑ What about the IP→name mapping?
- ❑ Separate server hierarchy stores reverse mappings
 - ▣ Rooted at in-addr.arpa and ip6.arpa
- ❑ Additional DNS record **type**: PTR
 - ▣ Name = IP address
 - ▣ Value = domain name
- ❑ Not guaranteed to exist for all IPs

Query

Name: 129.10.116.51
Type: PTR

Resp.

Name: 129.10.116.51
Value: ccs.neu.edu

DNS as Indirection Service

25

- ❑ DNS gives us very powerful capabilities
 - ▣ Not only easier for humans to reference machines!

- ❑ Changing the IPs of machines becomes trivial
 - ▣ e.g. you want to move your web server to a new host
 - ▣ Just change the DNS record!

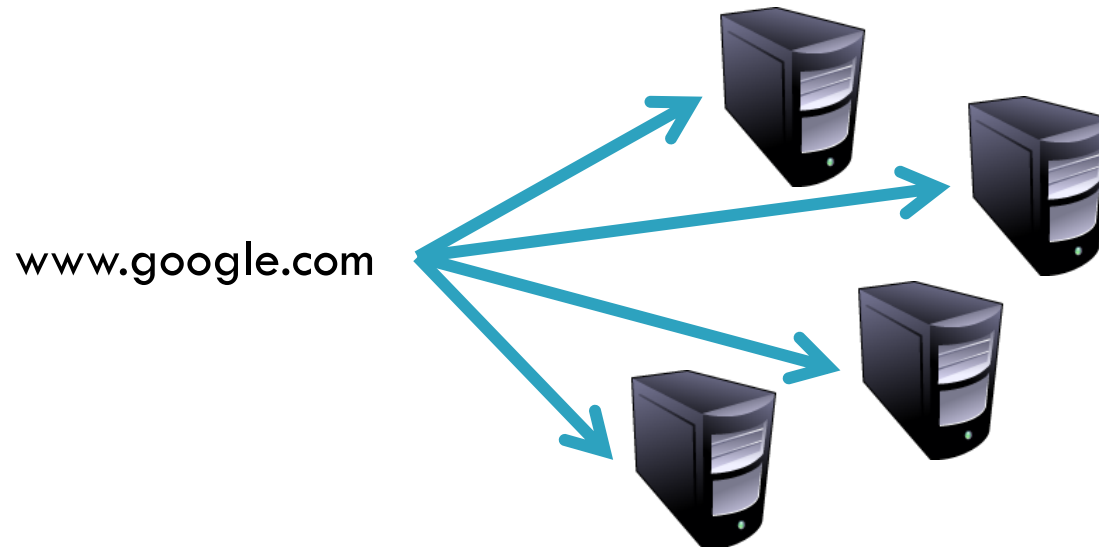
Aliasing and Load Balancing

26

- ❑ One machine can have many aliases

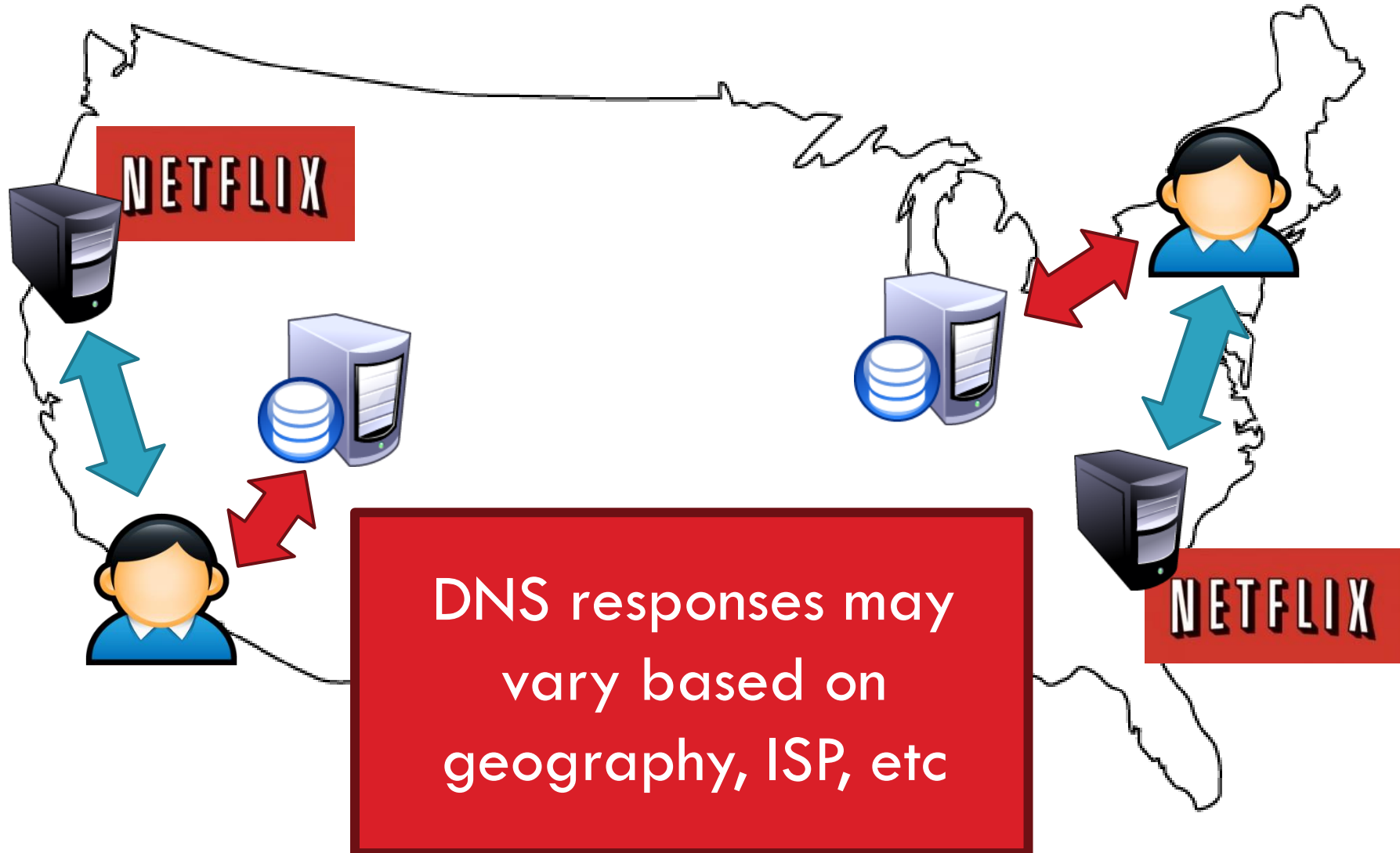


- ❑ One domain can map to multiple machines



Content Delivery Networks

27



- ❑ DNS Basics
- ❑ DNS Security
- ❑ DNS and Censorship

The Importance of DNS

29

- ❑ Without DNS...
 - ▣ How could you get to any websites?
- ❑ You are your mailserver
 - ▣ When you sign up for websites, you use your email address
 - ▣ What if someone hijacks the DNS for your mail server?
- ❑ DNS is the root of trust for the web
 - ▣ When a user types www.bankofamerica.com, they expect to be taken to their bank's website
 - ▣ What if the DNS record is compromised?

Denial Of Service

30

- ❑ Flood DNS servers with requests until they fail
- ❑ October 2002: massive DDoS against the root name servers
 - ▣ What was the effect?
 - ▣ ... users didn't even notice
 - ▣ Root zone file is cached almost everywhere
- ❑ More targeted attacks can be effective
 - ▣ Local DNS server → cannot access DNS
 - ▣ Authoritative server → cannot access domain

DNS Hijacking

31

- ❑ Infect their OS or browser with a virus/trojan
 - ❑ e.g. Many trojans change entries in /etc/hosts
 - ❑ *.bankofamerica.com → evilbank.com
- ❑ Man-in-the-middle



- ❑ Response Spoofing
 - ❑ Eavesdrop on requests
 - ❑ Race the servers response – Useful for censorship

D

Where is
bankofamerica.com?

123.45.67.89

32

How do you know that a given
name → IP mapping is correct?

Where is
bankofamerica.com?

66.66.66.93

123.45.67.89

dns.evil.com

66.66.66.93

Bank of America

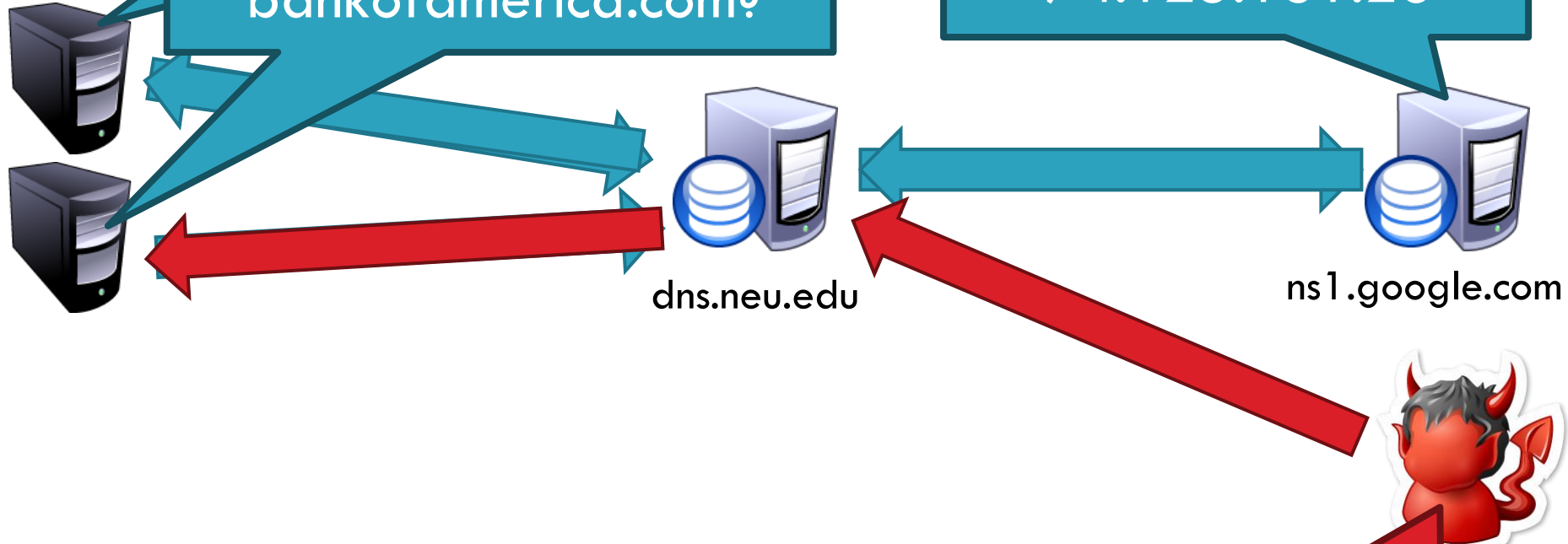


33

Where is

Where is
bankofamerica.com?

www.google.com =
74.125.131.26



- ❑ Until the TTL expires, all queries for BofA to dns.neu.edu will return poisoned results
- ❑ Much worse than spoofing/man-in-the-middle
 - ❑ Whole ISPs can be impacted!

How will the attacker get his entry into the cache? 2 ways

34

- ❑ 1. Tell resolver that NS for victim is at adversary's IP
 - ▣ Issue query: subdomain.attacker.example IN A
 - ▣ Attacker's response:
- ❑ Answer: (no response)
- ❑ Authority Section: attacker.example. 3600 IN NS ns.target.example.
- ❑ Additional Section: ns.target.example. IN A w.x.y.z

Adversary says “authoritative server for my domain is ns.target.example and oh by the way here is the IP for it (adversary's IP)

How will the attacker get his entry into the cache? 2 ways

35

- ❑ 2. Redirect the NS record to the adversary's domain
 - ▣ Issue query: subdomain.attacker.example IN A
 - ▣ Answer: (no response)
 - ▣ Authority section:
 - Target.example. 3600 IN NS ns.attacker.example.
 - ▣ Additional section:
 - Ns.attacker.example. IN A w.x.y.z

The attacker has inserted an unrelated piece of information that will be cached by the server (that target.example.'s ADNS is ns.attacker.example.)

Modern DNS Hijacking (IMC '16)

36

Country	ISP	DNS Servers	Hosts Affected
Argentina	Telefonica de Argentina	14	276
Australia	Dodo Australia	21	1,404
Brazil	Oi Fixo	21	2,558
	CTBC	4	290
Germany	Deutsche Telekom	8	1,385
India	Airtel Broadband	9	735
	BSNL	2	71
	Ntl. Int. Backbone	8	245
Malaysia	TMNet	8	1,676
Spain	Ono	2	71
U.K.	BT Internet	6	479
	Talk Talk	46	3,738
U.S.	AT&T	37	561
	Cable One	4	108
	Cox Communications	63	1,789
	Mediacom Cable	6	219
	Suddenlink	9	98
	Verizon	98	2,102
	WideOpen West	1	39

Basic DNS problems

37

- ❑ DNS is plain text
- ❑ Simple UDP, no sessions
- ❑ Tree structure with delegations
 - ▣ Each entity is responsible for a limited part of it
- ❑ Resolvers are victims of attacks, hijacks and mistakes
- ❑ Trust is needed

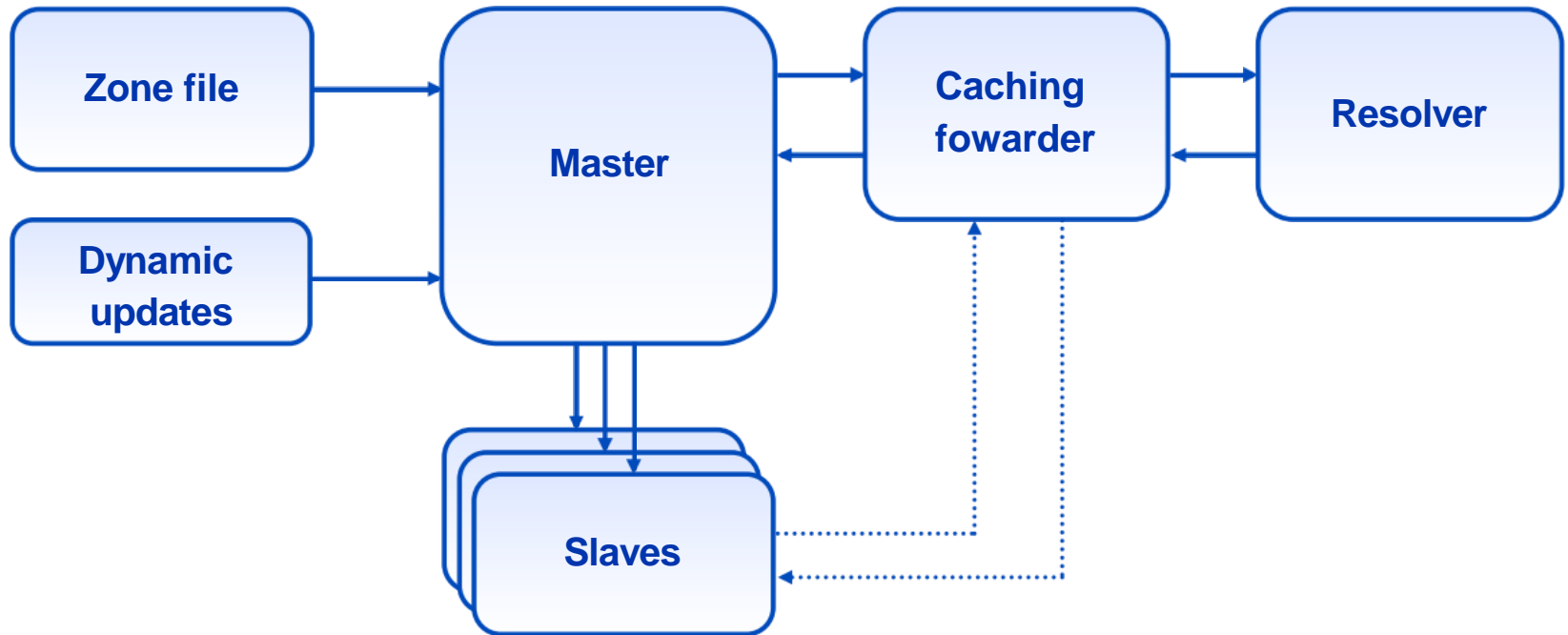
DNSSEC

38

- ❑ DNS Security Extensions
 - ▣ RFC4033
- ❑ Adds layers on top of DNS to make it verifiable
 - ▣ Adds new record types
 - ▣ Adds PKI
- ❑ Chain of trust to validate data

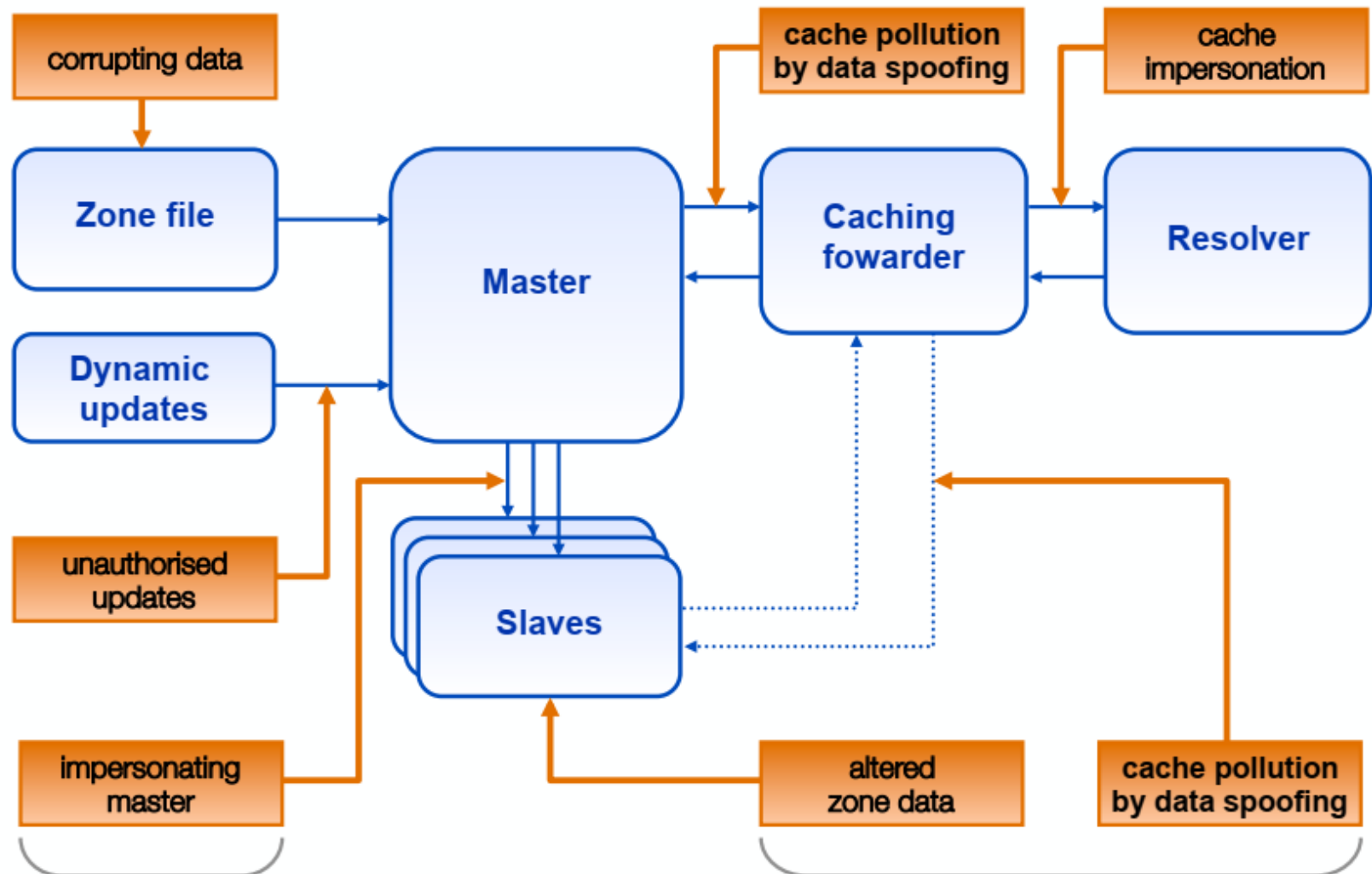
DNS Data Flow

39



DNS Vulnerabilities

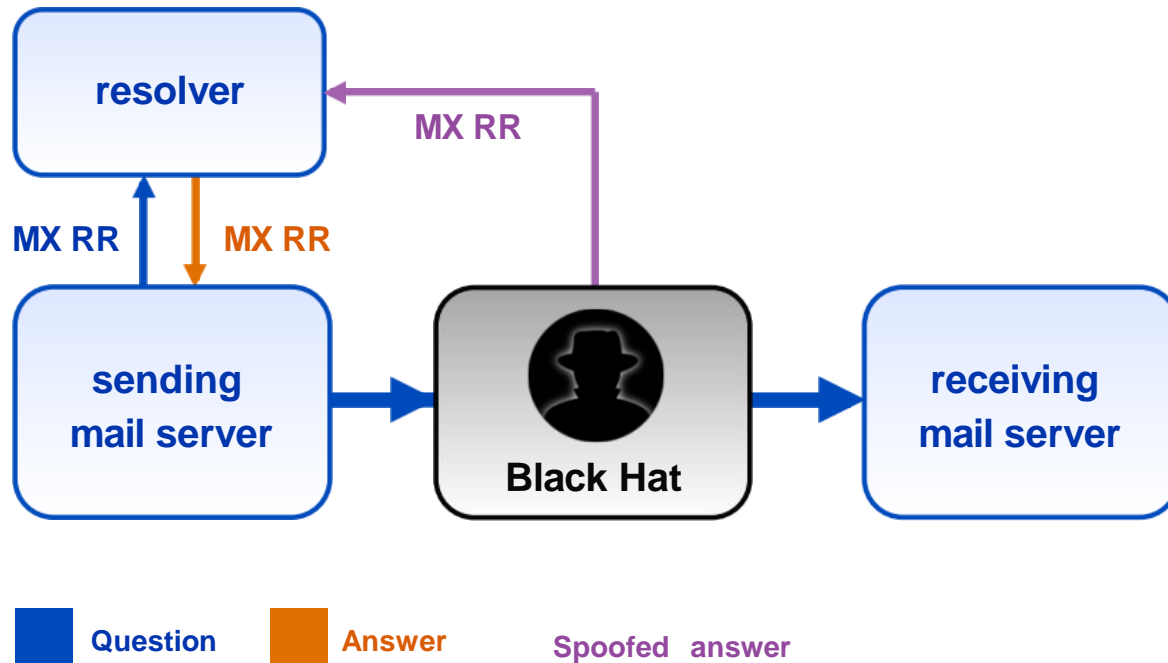
40



DNS Exploit Example

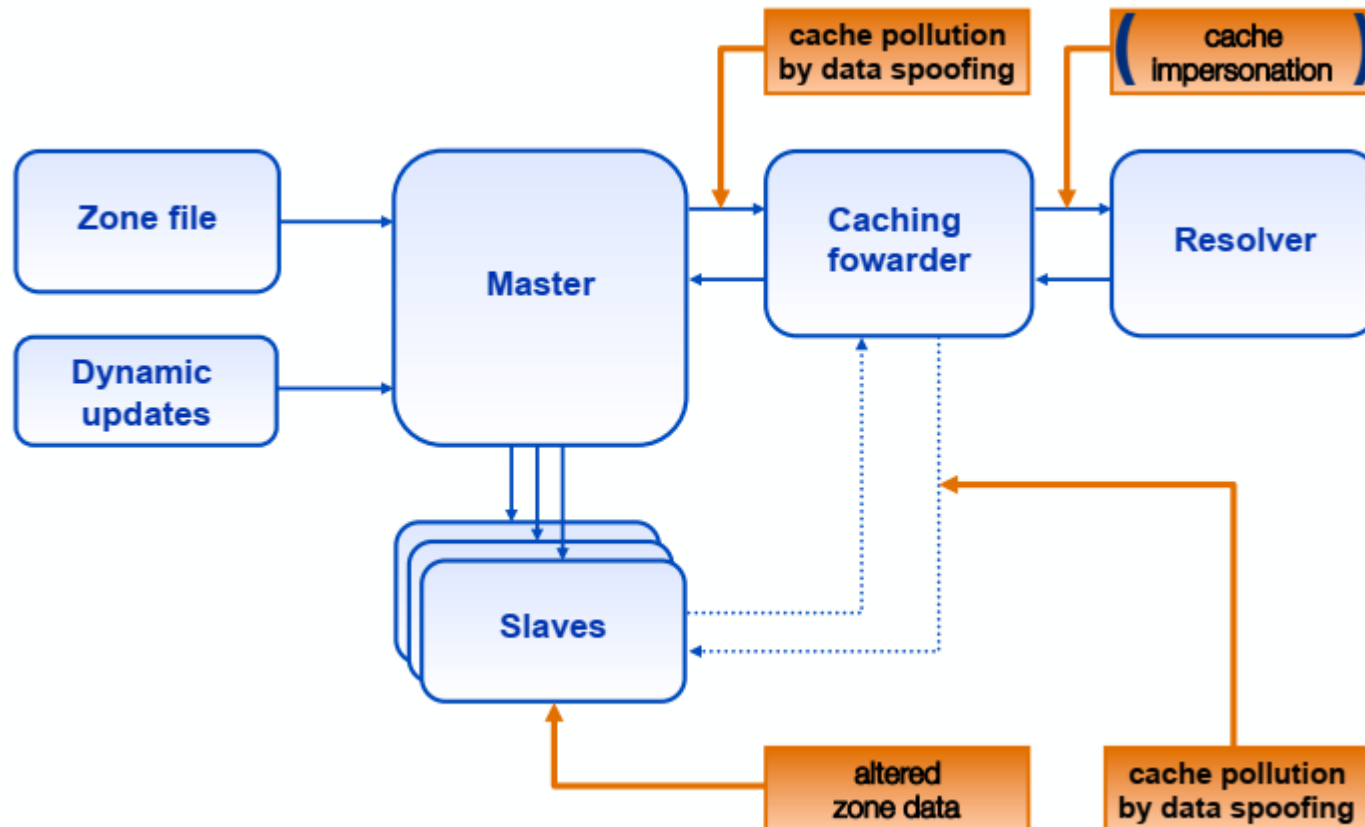
41

- ❑ Mail goes to the server in the MX resource record
- ❑ Path only visible in the email headers



DNSSEC Protected Vulnerabilities

42



The Recursive Resolver's View

43

- ❑ Recursive resolver will query them for records
 - ▣ and for authentication of records
- ❑ DNSSEC happens between server and resolver
 - ▣ Security status of records
 - ▣ Security status determines what client gets to see

Security Status of Data

44

☐ Secure

- ☐ Resolver can build chain of signed DNSKEY and DS RRs from trusted anchor to RRSet

☐ Insecure

- ☐ Resolver knows it has no chain of signed DNSKEY and DS RRs from any trusted starting point to RRset

☐ Bogus

- ☐ Resolver thinks it can build a chain of trust but it is unable to do so
- ☐ May indicate attack or configuration error or data corruption

☐ Indeterminate

- ☐ Resolver cannot determine whether the RRset should be signed

RRs and RRSets

45

❑ Resource Record

name	TTL	class	type	rdata
www.ripe.net.	7200	IN	A	192.168.10.3

❑ RRset: RRs with same name, class and type

www.ripe.net.	7200	IN	A	192.168.10.3
www.ripe.net.	7200	IN	A	10.0.0.3
www.ripe.net.	7200	IN	A	172.25.215.2

❑ RRSets are signed, not the individual RRs

New resource records

46

RRSIG

Signature over RRset

DNSKEY

Public key(s)

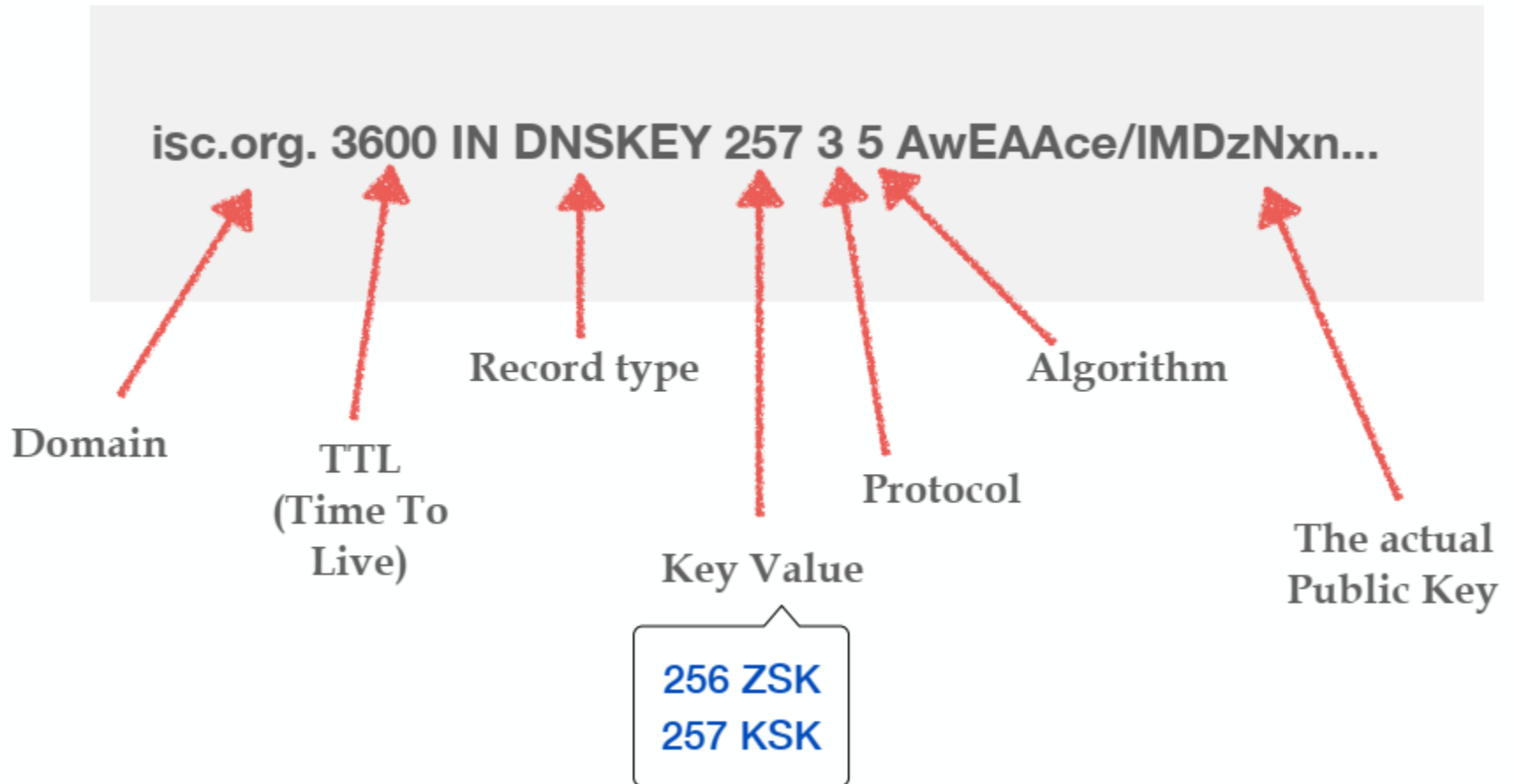
DS

Delegation Signer (hash of DNSKEY)

DNSKEY Record

47

- Contains Zone's public key(s)



DNSKEY Record

48

OWNER TYPE FLAGS PROTOCOL ALGORITHM

MYZONE. 600 DNSKEY 256 3 5 (

AwEAAdevJXb4NxFnDFT0Jg9d/jRhJwzM/YTu
PJqpvjRl14WabhabS6vioBX8Vz6XvnCzh1Ax

...) ; key id = 5538

KEY ID

PUBLIC KEY (BASE64)

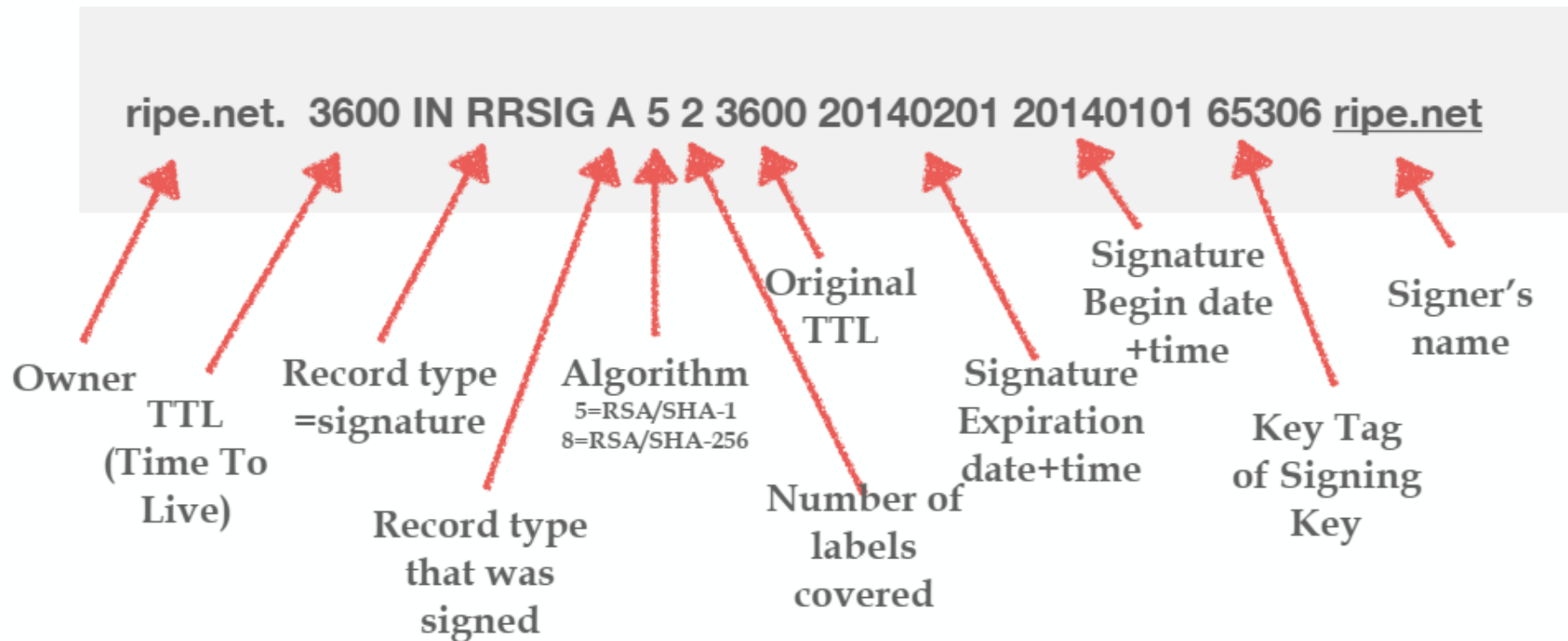
- FLAGS determines the usage of the key (more on this...)
- PROTOCOL is always 3 (DNSSEC)
- ALGORITHM can be:

0 – reserved	5 – RSA/SHA-1 (mandatory in validator)
1 – RSA/MD5 (deprecated)	8 – RSA/SHA-256
2 – Diffie/Hellman	
3 – DSA/SHA-1 (optional)	
4 – reserved	

RRSIG

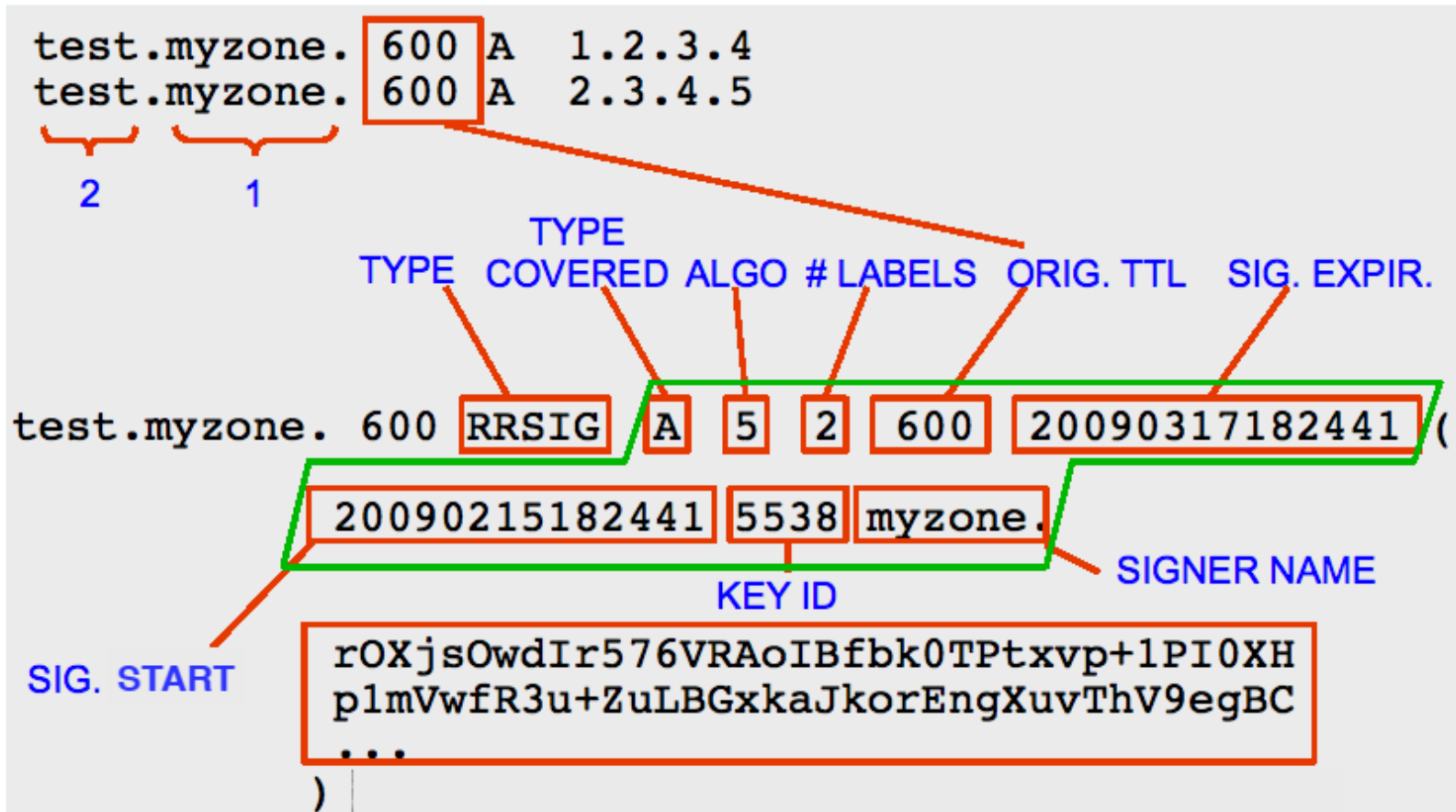
49

- ❑ Resource Record SIgnature
- ❑ Digital signature of a set of records



RRSIG

50



RR set

RRSIG

Delegation Signer Record

51

- ❑ The child's DNSKEY is hashed
- ❑ The hash of the key is signed by the parent's DNSKEY
 - ▣ and included in the parent's zone file
- ❑ Repeat for grandchild
- ❑ Chain of trust

Delegation Signer (DS)

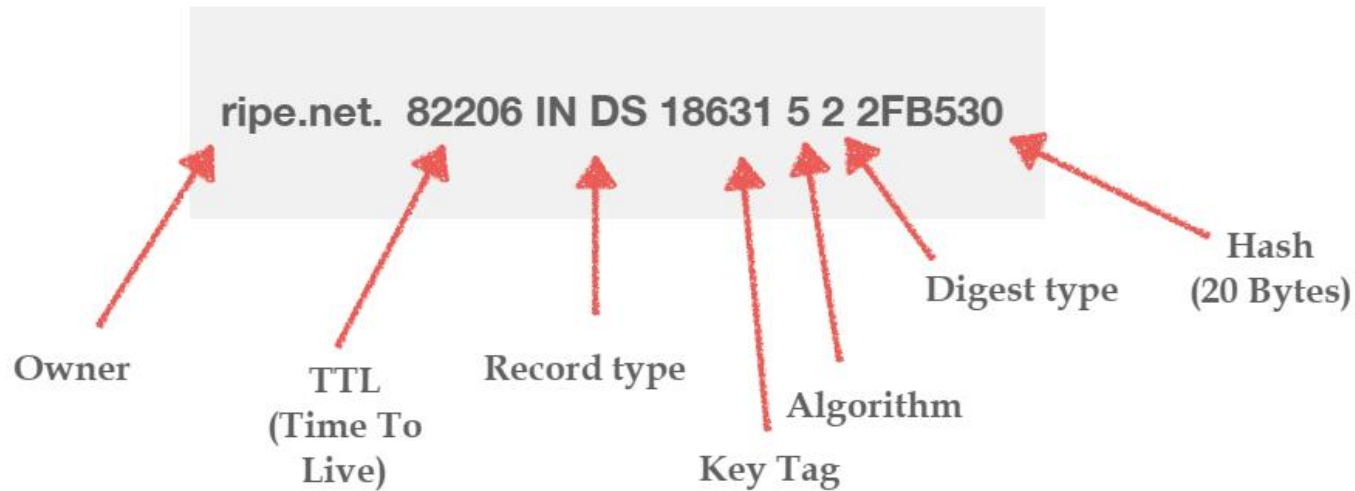
52

- ❑ Delegation Signer (DS) RR shows that:
 - ▣ child's zone is digitally signed
 - ▣ hashed key is used for the child's zone
- ❑ Parent is authoritative for the DS of the child's zone
 - ▣ DS should be in the parent's , not the child's zone

DS

53

- ❑ Delegation Signer
- ❑ Contains hash of the DNSKEY
- ❑ To be published in the parent zone of DNS chain



NSEC Record

54


- ❑ “Next SECure” record
- ❑ Authenticates non-existence of data
- ❑ Side Effect: allows discovery of zone contents

NSEC Example 1

55

ZONE FILE

```
ant.ripe.net NSEC baby.ripe.net A AAAA NSEC RRSIG
baby.ripe.net NSEC cat.ripe.net A NSEC RRSIG
cat.ripe.net NSEC dodo.ripe.net A AAAA NSEC RRSIG
dodo.ripe.net NSEC mouse.ripe.net A NSEC RRSIG
mouse.ripe.net NSEC ripe.net A AAAA NSEC RRSIG
ripe.net NSEC www.ripe.net A AAAA MX NSEC RRSIG
www.ripe.net NSEC ant.ripe.net A AAA NSEC RRSIG
```



Q: A for fruit.ripe.net ?

Doesn't exist! There is nothing between **dodo** and **mouse** !

A: **dodo.ripe.net** NSEC mouse.ripe net A NSEC RRSIG

RRSIG over NSEC

NSEC Example 2

56

ZONE FILE

```
ant.ripe.net NSEC baby.ripe.net A AAAA NSEC RRSIG
→ baby.ripe.net NSEC cat.ripe.net A NSEC RRSIG
cat.ripe.net NSEC dodo.ripe.net A AAAA NSEC RRSIG
dodo.ripe.net NSEC mouse.ripe.net A NSEC RRSIG
mouse.ripe.net NSEC ripe.net A AAAA NSEC RRSIG
ripe.net NSEC www.ripe.net A AAAA MX NSEC RRSIG
www.ripe.net NSEC ant.ripe.net A AAA NSEC RRSIG
```

Q: AAAA for baby.ripe.net ?

Doesn't exist! Its not in the list in the NSEC record

A: baby.ripe.net NSEC cat.ripe.net A NSEC RRSIG

RRSIG over NSEC

NSEC Record

57

- Points to the next domain name in the zone
 - ▣ also lists what are all the existing RRs for “owner”
 - ▣ NSEC record for last name “wraps around” to first name in zone
- Used for authenticated denial-of-existence of data
- authenticated non-existence of TYPEs and labels

www.ripe.net. 3600 IN NSEC ant.ripe.net. A RRSIG NSEC

“owner”
↓

next owner in zone file
↓

Existing Resource Record types for www.ripe.net
↓

Problem: NSEC Walk

58

- ❑ NSEC records allow for zone “re-construction”
- ❑ Causes privacy issues
- ❑ It's a deployment barrier

Solution: NSEC3 Record

59

- ❑ Same as NSEC
- ❑ But hashes all names to avoid zone discovery
- ❑ Hashed names are ordered

```
DRVR6JA3E4VO5UIPOFAO5OEEVV2U4T1K.dnssec-course.net. 3600 IN  
NSEC3 1 0 10 03F92714 GJPS66MS4J1N6TIIJ4CL58TS9GQ2KRJ0 A RRSIG
```

New Resource Records

60

- ❑ Three Public key crypto related RRs
 - ▣ RRSIG Signature over RRset using private key
 - ▣ DNSKEY Public key, needed for verifying an RRSIG
 - ▣ DS Delegation Signer; 'Pointer' for building chains of authentication
- ❑ One RR for internal consistency
 - ▣ NSEC shows which name is the next one in the zone and which types exist for the name queried
 - ▣ authenticated non-existence of data

What if There Was No DS?

61

- Without delegating signing authority (DS) the
 - ▣ resolver would need to store millions of public keys
- But with DS only one key is needed: the root key

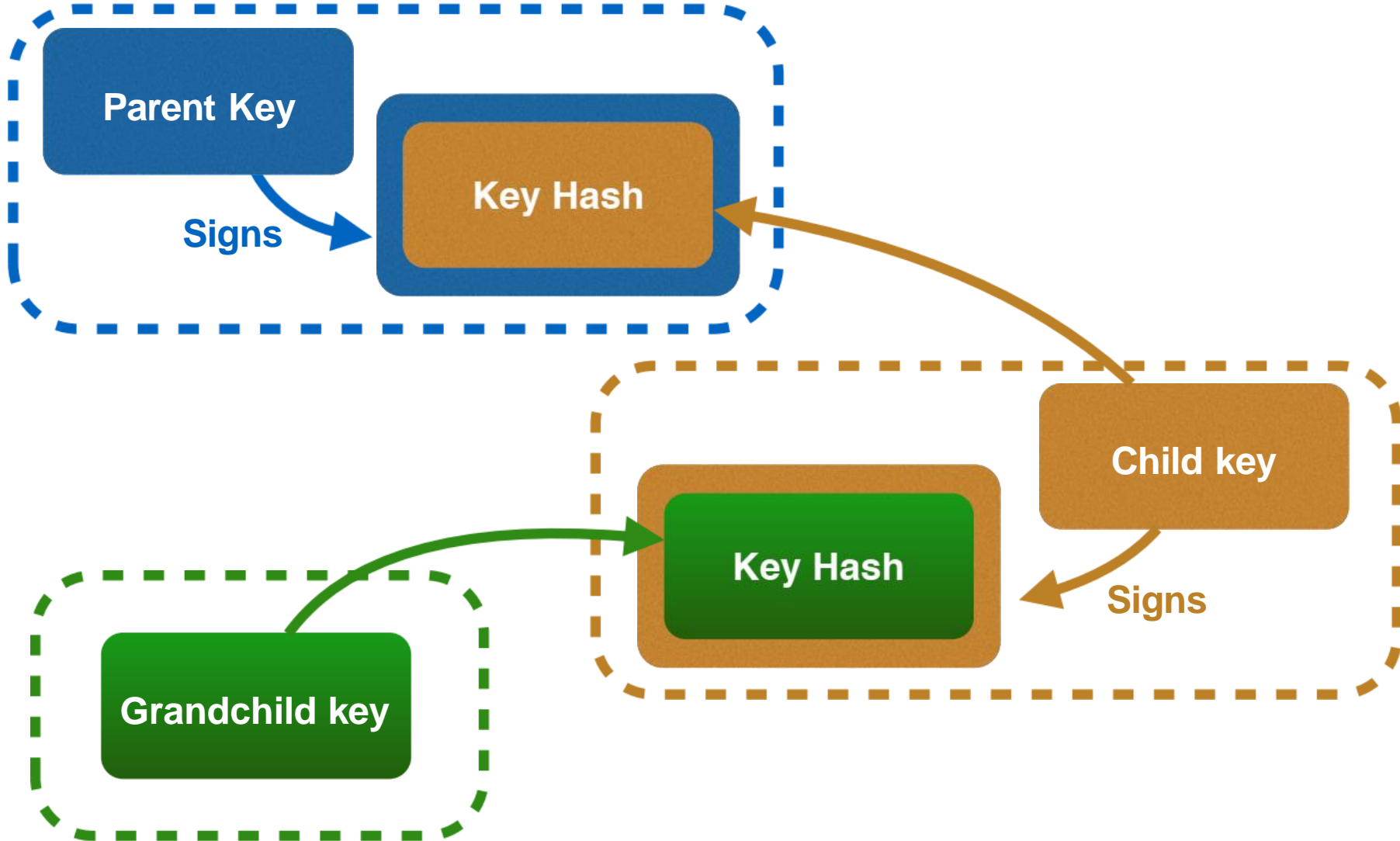
DNS and Keys

62

- ❑ DNS is made of islands of trust, with delegations
- ❑ A parent needs to have pointers to child keys
 - ▣ in order to sign/verify them
 - ▣ DS Records are used for this
- ❑ You want to keep interaction between parent and children at a minimum

DNSSEC Made simple

63



Key Problem

64

- ❑ Interaction with parent administratively expensive
 - ▣ Should only be done when needed
 - ▣ Bigger keys are better
- ❑ Signing zones should be fast
 - ▣ Memory restrictions
 - ▣ Space and time concerns
 - ▣ Smaller keys with short lifetimes are better

Key Functions

65

- ❑ Large keys are more secure
 - ▣ Can be used longer
 - ▣ Large signatures \Rightarrow large zone files ✕
 - ▣ Signing and verifying computationally expensive ✕
- ❑ Small keys are fast
 - ▣ Small signatures
 - ▣ Signing and verifying less expensive
 - ▣ Short lifetime ✕

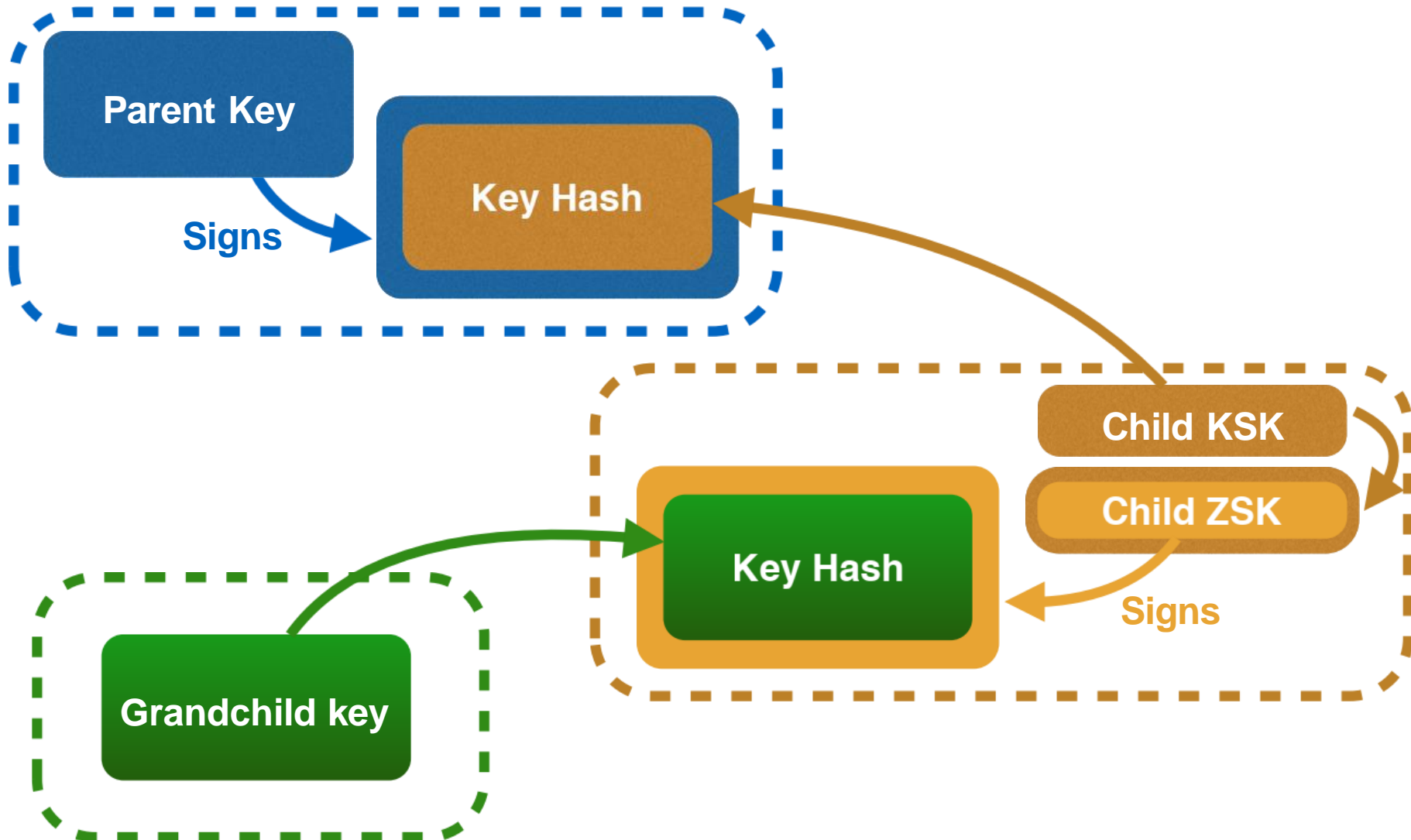
Key Solution: More Than One Key

66

- ❑ Key Signing Key (KSK) only signs DNSKEY RRset
- ❑ Zone Signing Key (ZSK) signs all RRsets in zone
- ❑ RRsets are signed, not RRs
- ❑ DS points to child's KSK
 - ▣ Parent's ZSK signs DS
 - ▣ Signature transfers trust from parent key to child key

Key split - ZSK and KSK

67



Zone Signing Key - ZSK

68

- ❑ Used to sign a zone
- ❑ Can be lower strength than the KSK
- ❑ No need to coordinate with parent zone if you want to change it

Key Signing Key - KSK

69

- ❑ Only signs the Resource Record Set containing DNSKEYs for a zone
- ❑ Used as the trust anchor
- ❑ Needs to be specified in the parent zone using DS (Delegation Signature) records

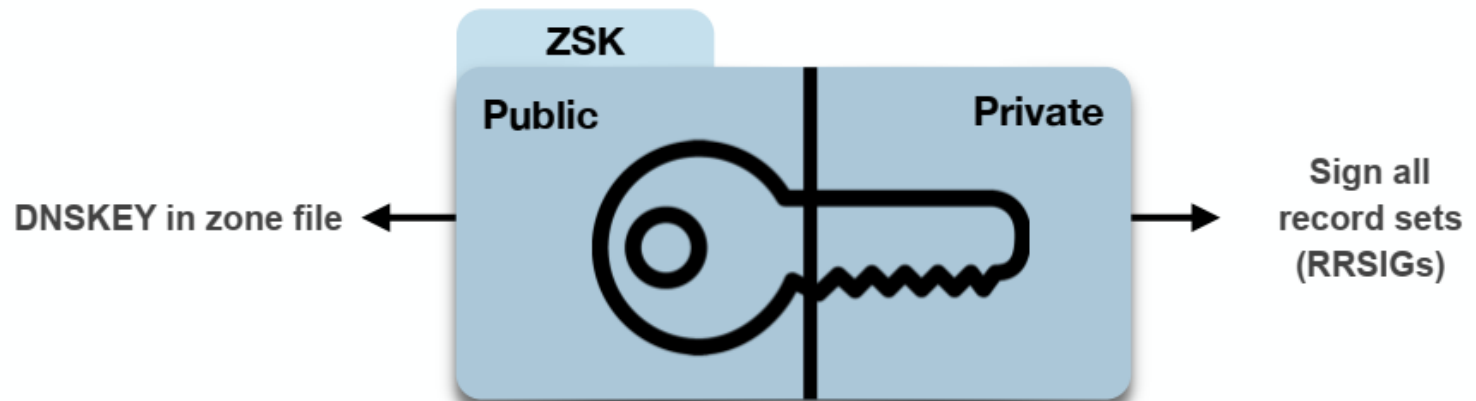
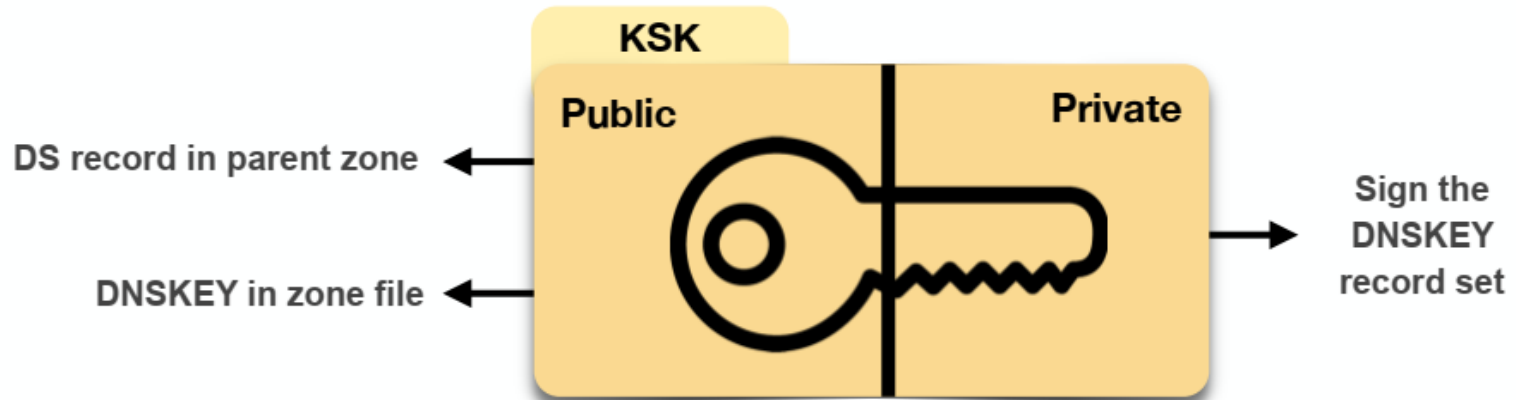
Initial Key Exchange

70

- Child needs to:
 - ▣ Send key signing keyset to parent
- Parent needs to:
 - ▣ Check childs zone
 - for DNSKEY & RRSIGs
 - ▣ Verify if key can be trusted
 - ▣ Generate DS RR

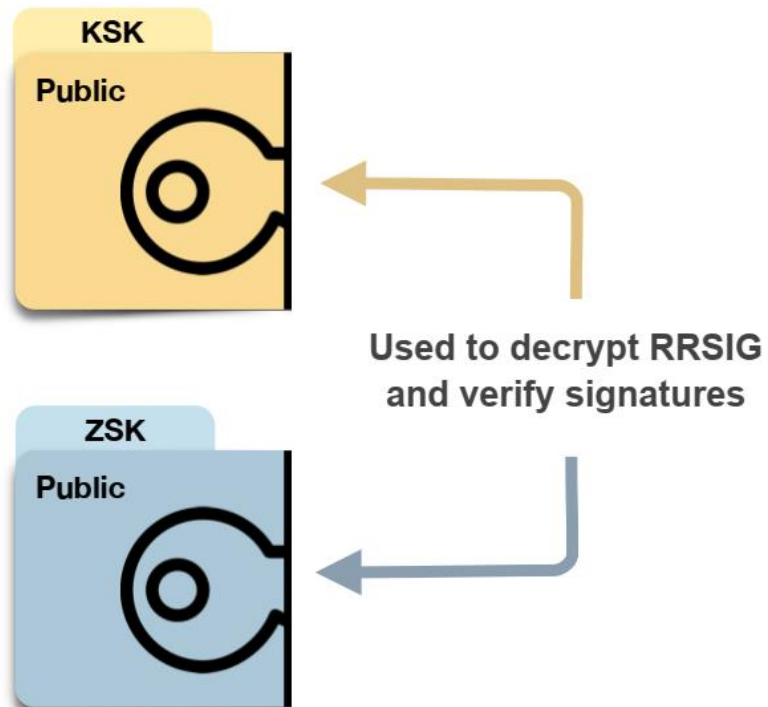
Keys

71



Keys

72



PARENT

DNSKEY (KSK)

DNSKEY (ZSK)

DS

RRSIG DS

← hash of child's (public) KSK

← signed by Parent's (private) ZSK

CHILD

MX
MX
MX

Record Set

RRSIG MX

← signed by (private) ZSK

A
A
A

Record Set

RRSIG A

← signed by (private) ZSK

DNSKEY (KSK)

DNSKEY (ZSK)

← (public) KSK

← (public) ZSK

RRSIG DNSKEY

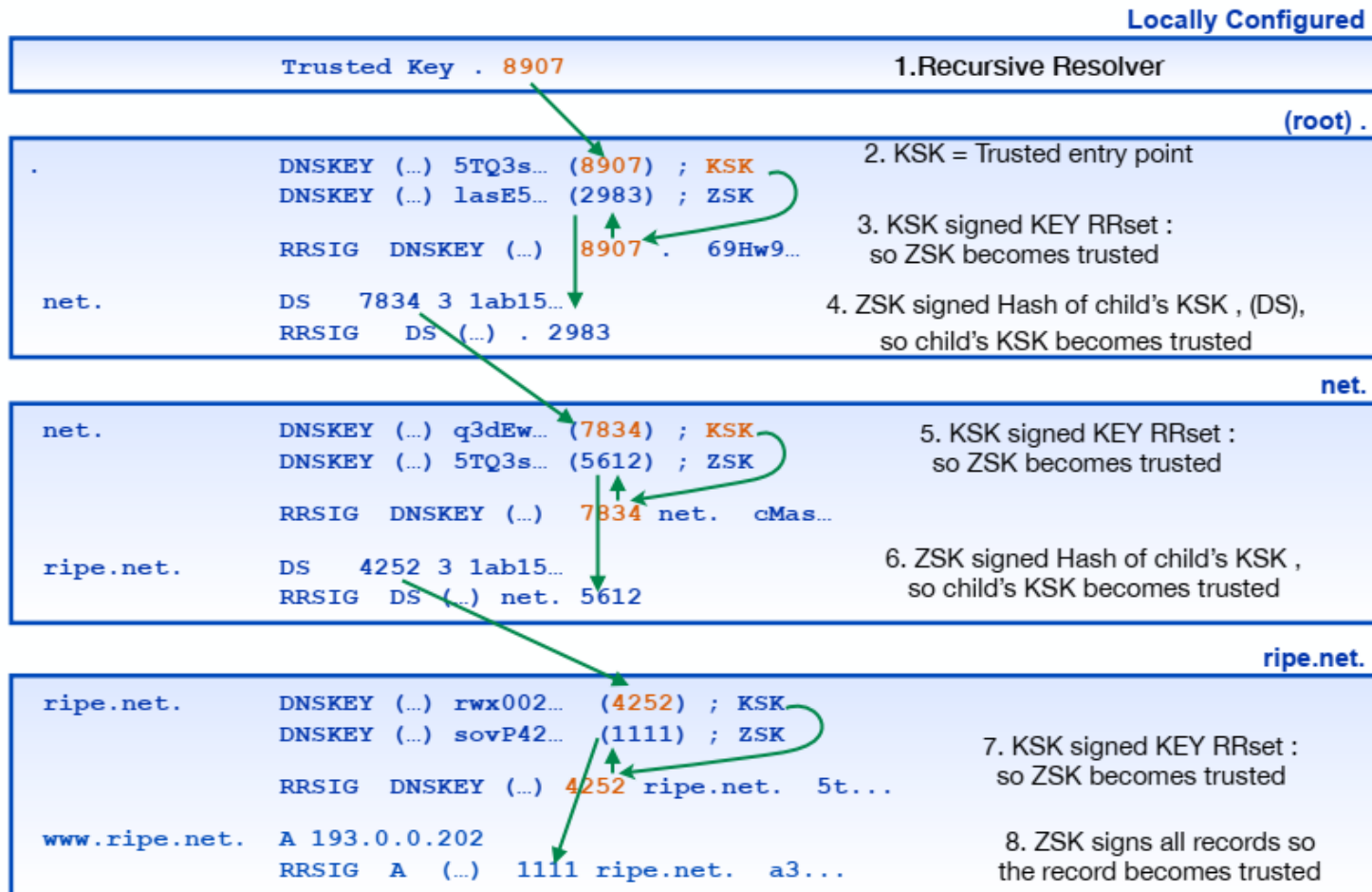
← signed by (private) ZSK

RRSIG DNSKEY

← signed by (private) KSK

Walking the Chain of Trust

74



Key Rollovers: Keys need to be changed

75

- ❑ Keys become old quickly
 - ▣ New exploits are discovered every day
 - ▣ Brute force becomes less and less expensive
- ❑ Your keys could be stolen or compromised
- ❑ You need to have a plan

Key rollover methods

76

- ❑ Pre-publish
- ❑ Double signature
- ❑ Both for ZSK and KSK
 - ▣ Rolling a KSK means changing parent DS records
- ❑ Rollover times depend on TTL and method

Pre-publishing Method

77

- ❑ A new DNSKEY record is introduced with new key
 - ▣ Not used for signing, yet
- ❑ After TTL expires, new RRSIGs are created with new DNSKEY
 - ▣ Old DNSKEY remains published
- ❑ After TTL expires again, old DNSKEY is removed

Double signature Method

78

- A new DNSKEY is introduced, and immediately used to sign the records
- We have two RRSIGs for every record, with signatures from both DNSKEYs
- After TTL expires, old DNSKEY is removed, and records are again signed only once

Do We Have to Remember to Rollover?

79

- ❑ No, we can automate it
 - ▣ in the configuration
 - ▣ including the schedule
- ❑ just provide ahead of time enough DNSSEC keys for the next few rollovers

Recommendations

80

- ❑ Use pre-publishing for ZSK
 - ▣ Especially for large zones
- ❑ Use double signature for KSK
 - ▣ KSK double-signs the DNSKEY, not the zone
- ❑ For KSK rollovers, update DS records

Does DNSSEC Solve all our problems?

81

- ❑ No.
- ❑ DNS still vulnerable to reflection attacks + injected responses

DNS Reflection

82

❑ Very big incident in 2012

- ❑ (<http://blog.cloudflare.com/65gbps-ddos-no-problem/>)
- ❑ 65 Gbps DDoS
- ❑ Would need to compromise 65,000 machines each with 1 Mbps uplink
 - How was this attack possible?

❑ Use DNS reflection to amplify a Botnet attack.

❑ Key weak link: Open DNS resolvers will answer queries for anyone <http://openresolverproject.org/>

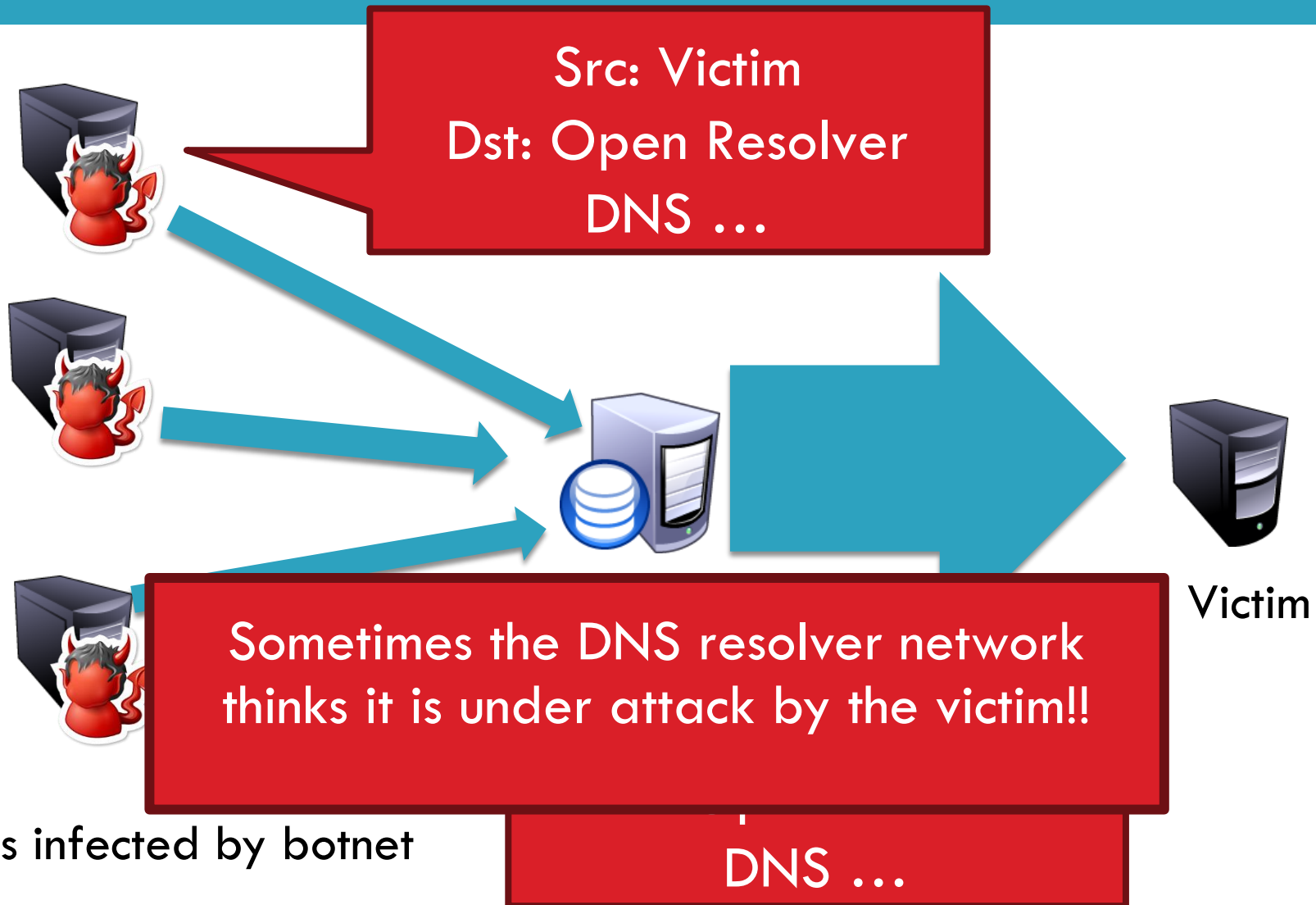
So how does this work?

83

- ❑ Remember: DNS is UDP
- ❑ No handshaking between endpoints
- ❑ One can send a DNS query with a forged IP address and the response will go to that IP address
 - ▣ **Secret sauce:** a small request that can elicit a large response
 - ▣ E.g., query for zone files, or DNSSEC records (both large record types).
- ❑ Botnet hosts spoof DNS queries with victim's IP address as source
 - ▣ Resolver responds by sending massive volumes of data to the victim

DNS amplification illustrated

84



Amplification not unique to DNS

85

- ❑ NTP is the latest protocol to be used in this way:
- ❑ <http://www.prolexic.com/news-events-pr-threat-advisory-ddos-ntp-amplification.html>
- ❑ (Exploiting NTP Monlist command which returns a list of 600 most recent hosts to connect to the NTP server)

How well is DNSSEC managed?

86

- ❑ Looked at 147M domains, 60K+ resolvers over 21 months
- ❑ TLDs, ccTLDs broadly deploy
 - ▣ But only 1-2% of second-level domains use it
- ❑ Pervasive record mismanagement
 - ▣ Nearly 1/3 of DNSSEC-enabled domains can't be validated
- ❑ Resolvers not validating properly
 - ▣ 80+% of resolvers ask for DNSSEC records
 - ▣ Only 12% actually bother to check the result!

- ❑ DNS Basics
- ❑ DNS Security
- ❑ DNS and Censorship

DNS and Censorship



- ❑ DNS is a popular protocol for targeting by Internet censors
- ❑ A few things to keep in mind ...
- ❑ No cryptographic integrity of DNS messages
 - ▣ DNSSEC proposed but not widely implemented
- ❑ Caching of replies means leakage of bad DNS data can persist

Blocking DNS Names

- Can the censor pressure the registrar?

Name blocked, forever

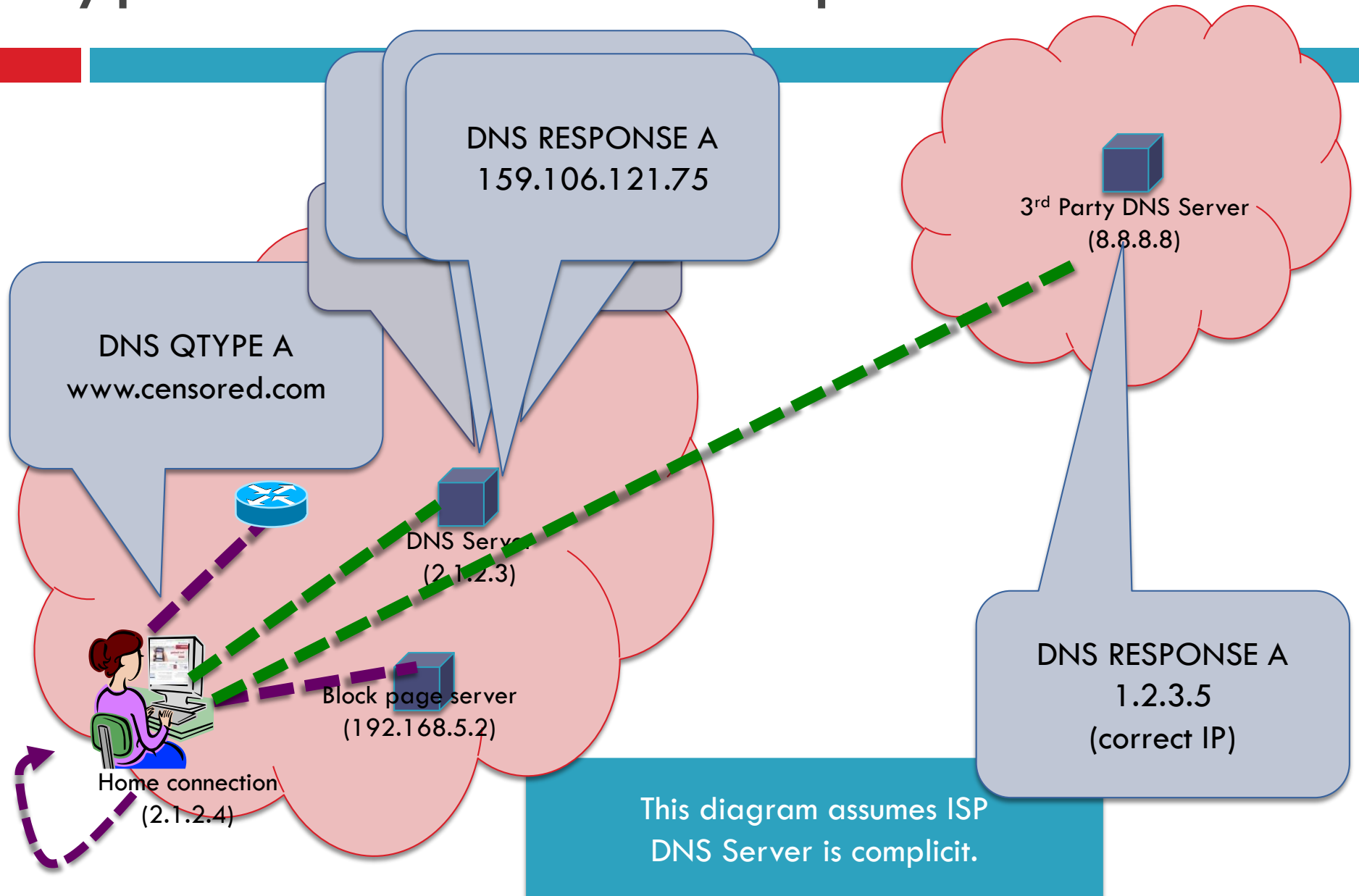


Blocking DNS Names



- ❑ Can the censor pressure the ISPs?
 - ▣ Just force an entry in the recursive resolver to poison results for a given domain
- ❑ Clients can trivially evade this using alternate DNS services
 - ▣ E.g., Google's 8.8.8.8
 - ▣ ...but this does require client changes
 - ▣ Also, ISPs must not block third party DNS queries for this to work
- ❑ Initially used by ISPs in the UK to block the Pirate Bay

Types of false DNS responses



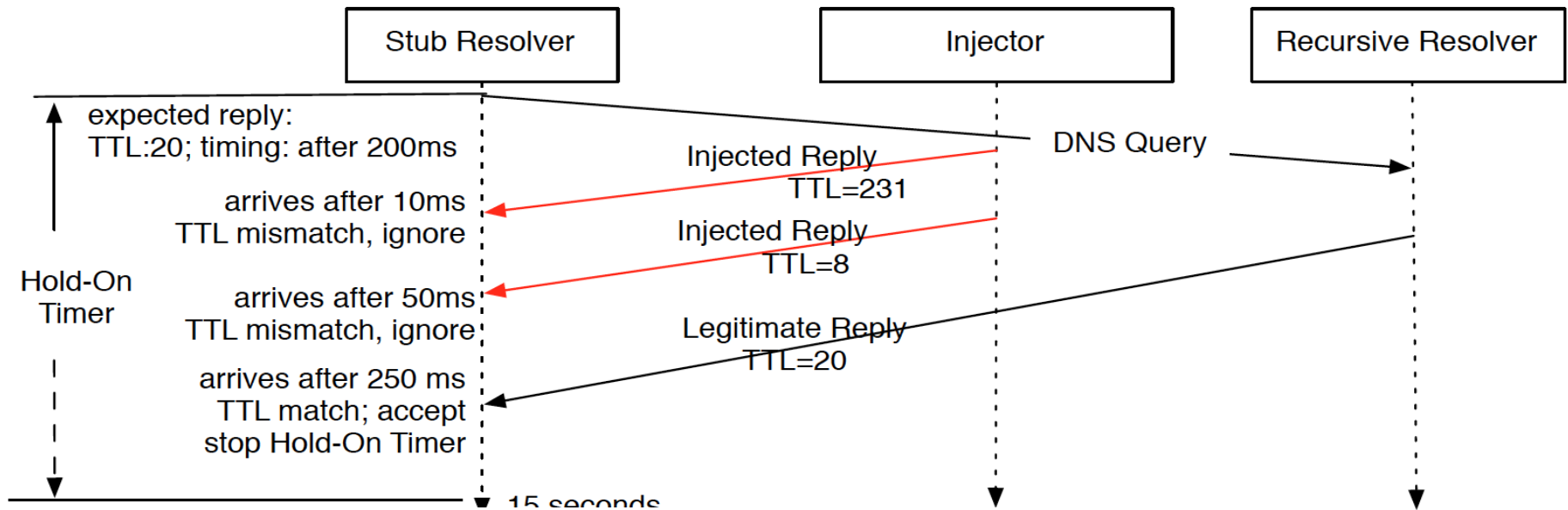
Blocking DNS names

- ❑ Option A: get ISP resolver on board
 - ▣ (Previous slide)
- ❑ Option B: On-path packet injection
 - ▣ Censor injects a DNS response that **races** the legitimate reply
 - ▣ Can be mostly countered with DNS-hold-open:
 - Don't take the first answer but instead wait for up to a second
 - ▣ Generally reliable when using an out of country recursive resolve (e.g., 8.8.8.8, censor packet should win the race)
 - ▣ Can be **completely** countered by DNS-hold-open + DNSSEC
 - Accept the first DNS reply **which validates**

Reading from Web ...

- ❑ **Hold-On: Protecting Against On-Path DNS Poisoning**, H. Duan, N. Weaver, Z. Zhao, M. Hu, J. Liang, J. Jiang, K. Li, and V. Paxson.
- ❑ **Idea:** Once you receive a DNS packet, wait for a predefined “hold-on” period before accepting the result.
 - ▣ DNSSEC is still vulnerable to these injected packets and does not make hold-on unnecessary
 - ▣ Censor can just inject a reply with an invalid signature: client will reject (denial of service)
- ❑ **Method:** Use active measurements to determine the expected TTL and RTT to the server.

Hold-on in action



Much More to DNS

95

- ❑ Caching: when, where, how much, etc.
- ❑ Other uses for DNS (i.e. DNS hacks)
 - ▣ Content Delivery Networks (CDNs)
 - ▣ Different types of DNS load balancing
 - ▣ Dynamic DNS (e.g. for mobile hosts)
- ❑ DNS and botnets
- ❑ Politics and growth of the DNS system
 - ▣ Governance
 - ▣ New TLDs (.xxx, .biz), eliminating TLDs altogether
 - ▣ Copyright, arbitration, squatting, typo-squatting

DNSSEC Summary

96

- ❑ Data authenticity and integrity by signing the
 - ❑ Resource Records Sets with **private DNSKEY**
 - ❑ You need **Public DNSKEYs** to verify the RRSIGs
- ❑ Children sign their zones with their **private key**
 - ❑ Parent guarantees authenticity of child's key by signing the hash of it (DS)
- ❑ Repeat for parent ...
 - ❑ ...and grandparent
- ❑ Ideal case: one **public DNSKEY** distributed

DNSSEC Summary

97

ripe.net.

www.ripe.net	IN A	193.0.0.214
www.ripe.net	IN RRSIG	A ... 26523 ripe.net.
ripe.net	IN DNSKEY	256 26523 ... ripe.net.
ripe.net	IN RRSIG	DNSKEY 32987 ... ripe.net.
ripe.net	IN DNSKEY	257 32987 ... ripe.net.

net.

ripe.net	IN DS	26523 8 1 ...
ripe.net	IN RRSIG	DS ... 43249 net.
net	IN DNSKEY	256 43249 ... net.

DNSSEC Summary

98

- ❑ Cryptographically sign critical resource records
 - ▣ Resolver can verify the cryptographic signature
- ❑ Two new resource **types**
 - ▣ Type = DNSKEY
 - Name = Zone domain name
 - Value = Public key for the zone
 - ▣ Type = RRSIG
 - Name = (type, name) tuple, i.e. the query itself
 - Value = Cryptographic signature of the query results
- ❑ Deployment
 - ▣ On the roots since July 2010
 - ▣ Verisign enabled it on .com and .net in January 2011
 - ▣ Comcast is the first major ISP to support it (January 2012)



Creates a hierarchy of
trust within each zone
and spoofing

DNSSEC Hierarchy of Trust

99

