DNSSEC Tutorial

Auckland, New Zealand

06 August 2018

As part of:



Issue Date:

Revision:

APNIC

Outline

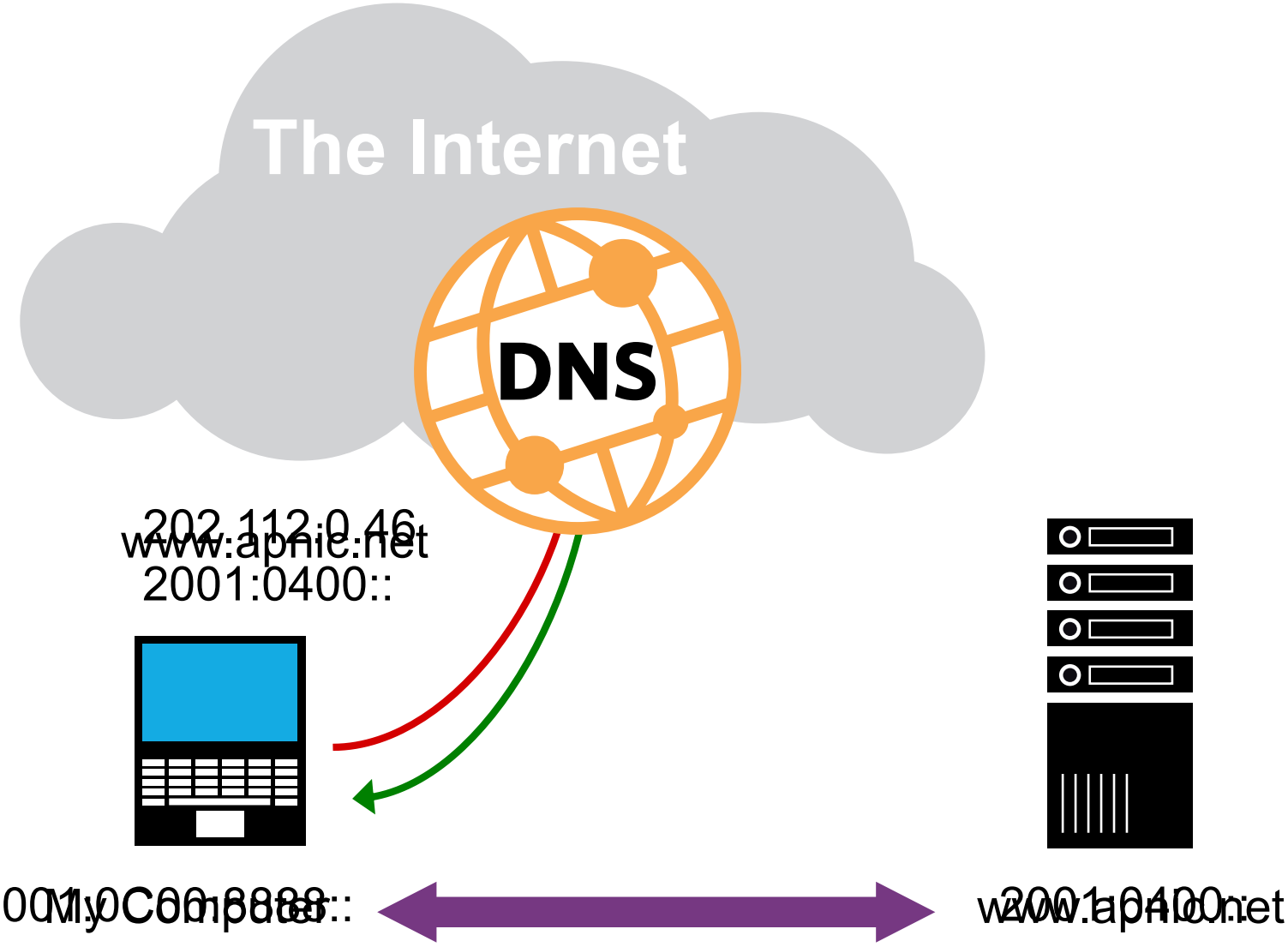
- DNS Operations
- DNS Security Concepts
- DNSSec Primer
- Implementing DNSSec

DNS Operations

Domain Name System

- A lookup mechanism for translating objects into other objects
  - Mapping names to numbers and vice versa
- A critical piece of the Internet infrastructure
- Three components
  - A “name space”
  - Servers making that name space available
  - Resolvers (clients) query the servers about the name space

IP Addresses vs Domain Names



Old Solution: hosts.txt

- A centrally-maintained file, distributed to all hosts on the Internet
- Issues with having just one file
  - Becomes huge after some time
  - Needs frequent copying to ALL hosts
  - Consistency
  - Always out-of-date
  - Name uniqueness
  - Single point of administration

```
// hosts.txt
SERVER1         128.4.13.9
WEBMAIL         4.98.133.7
FTPHOST         200.10.194.33
```

This feature still exists:
[Unix] /etc/hosts
[Windows] c:\windows\hosts

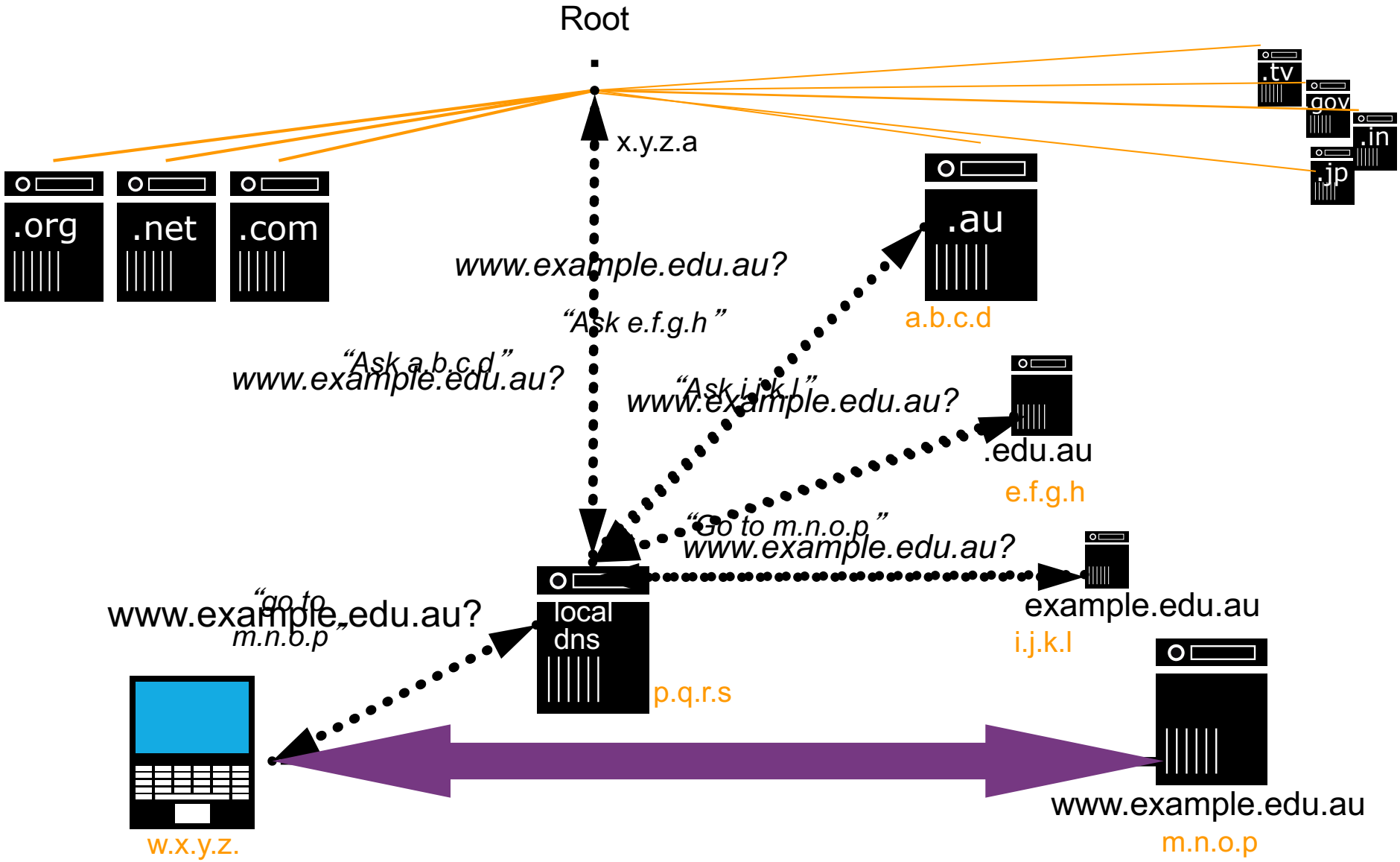
DNS Features

- Global distribution
  - Shares the load and administration
- Loose Coherency
  - Geographically distributed, but still coherent
- Scalability
  - can add DNS servers without affecting the entire DNS
- Reliability
- Dynamicity
  - Modify and update data dynamically

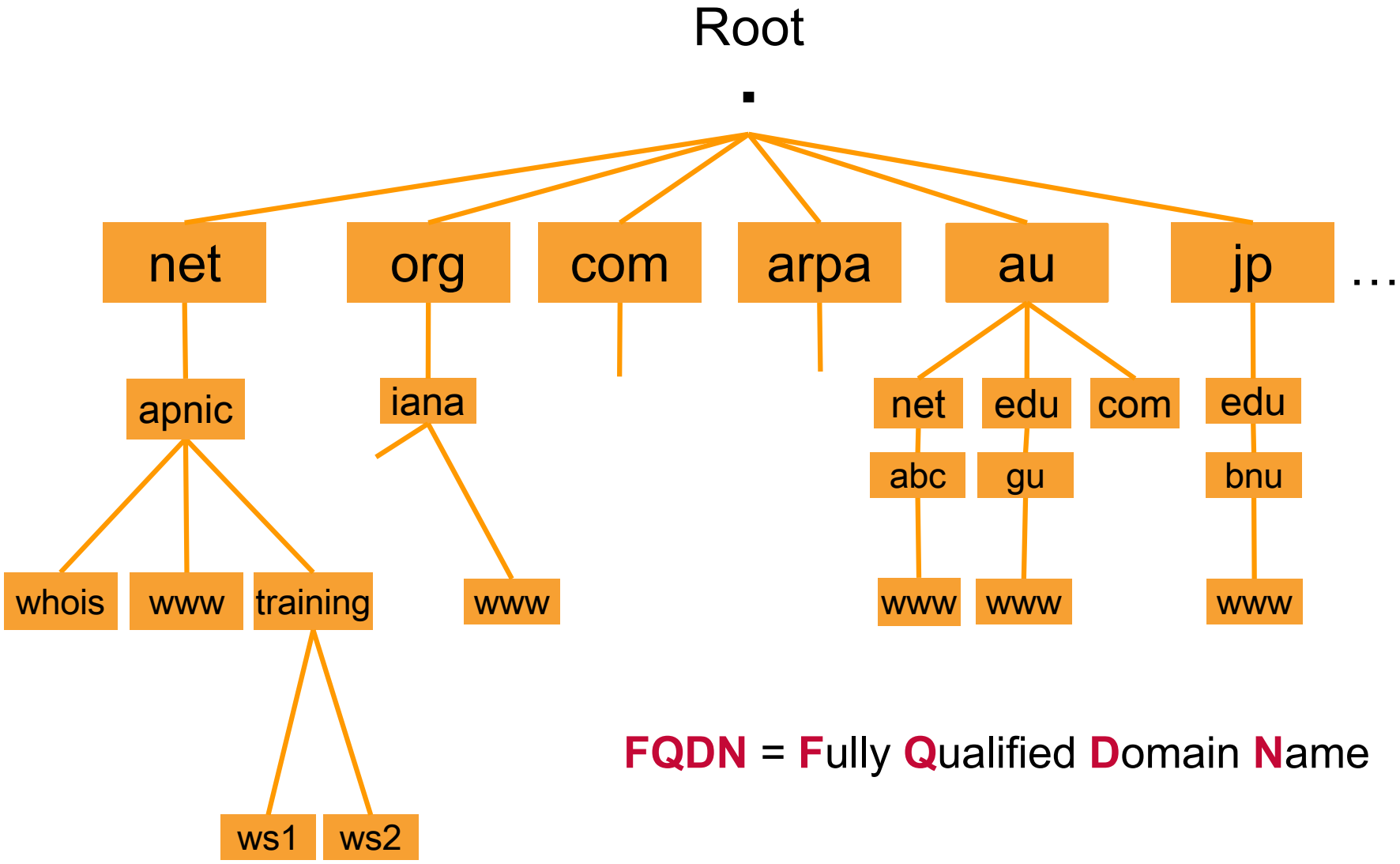
DNS Features

- DNS is a client-server application
- Requests and responses are normally sent in UDP packets, port 53
- Occasionally uses TCP, port 53
  - for very large requests, e.g. zone transfer from primary to secondary

Querying the DNS – It's all about IP!



The DNS Tree Hierarchy

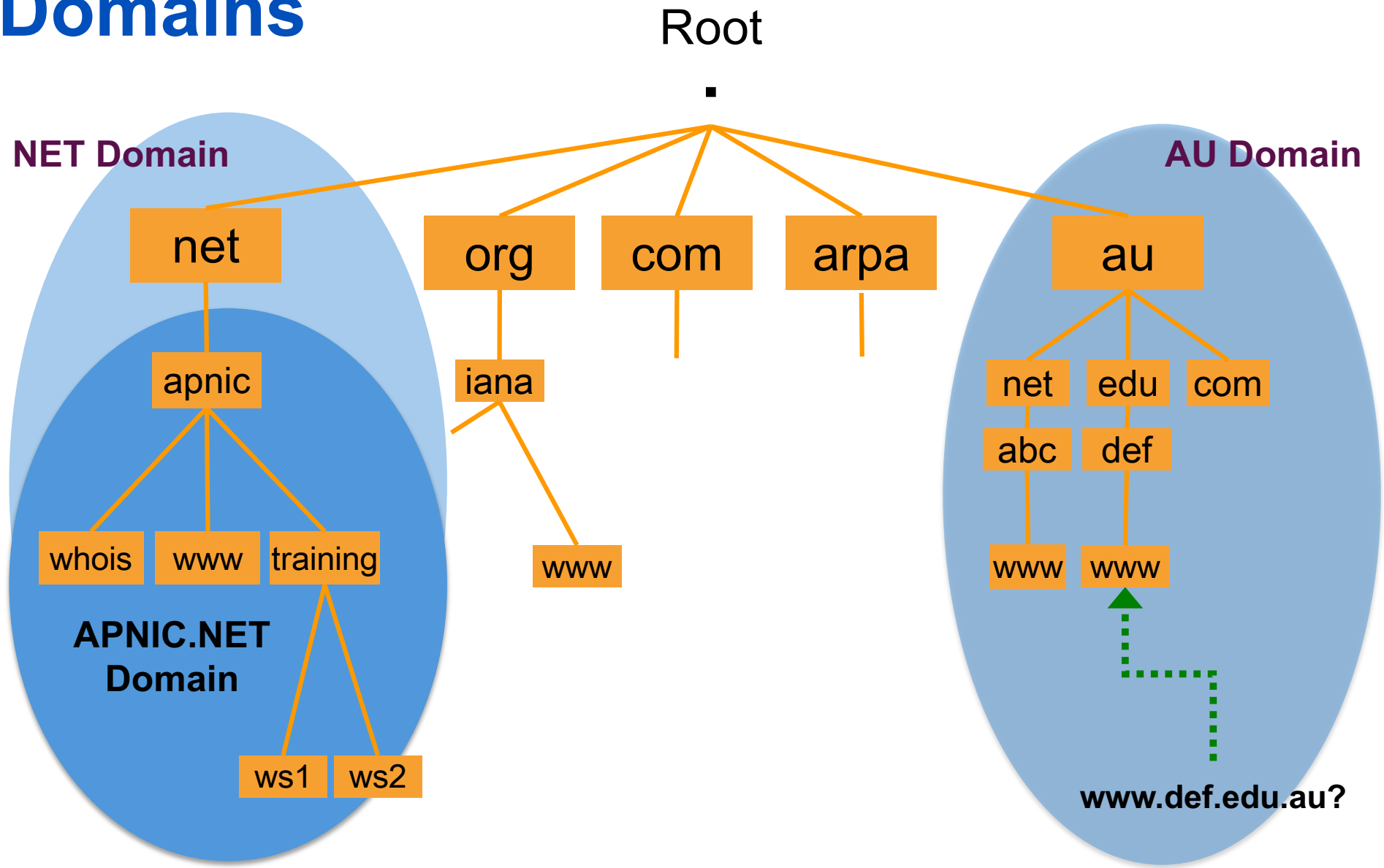


FQDN = **F**ully **Q**ualified **D**omain **N**ame

Domains

- Domains are “namespaces”
- Everything below .com is in the com domain
- Everything below apnic.net is in the apnic.net domain and in the net domain

Domains


Root
.
NET Domain
AU Domain
net
org
com
arpa
au
apnic
iana
net
edu
com
abc
def
whois
www
training
www
www
www
APNIC.NET
Domain
ws1
ws2
www.def.edu.au?

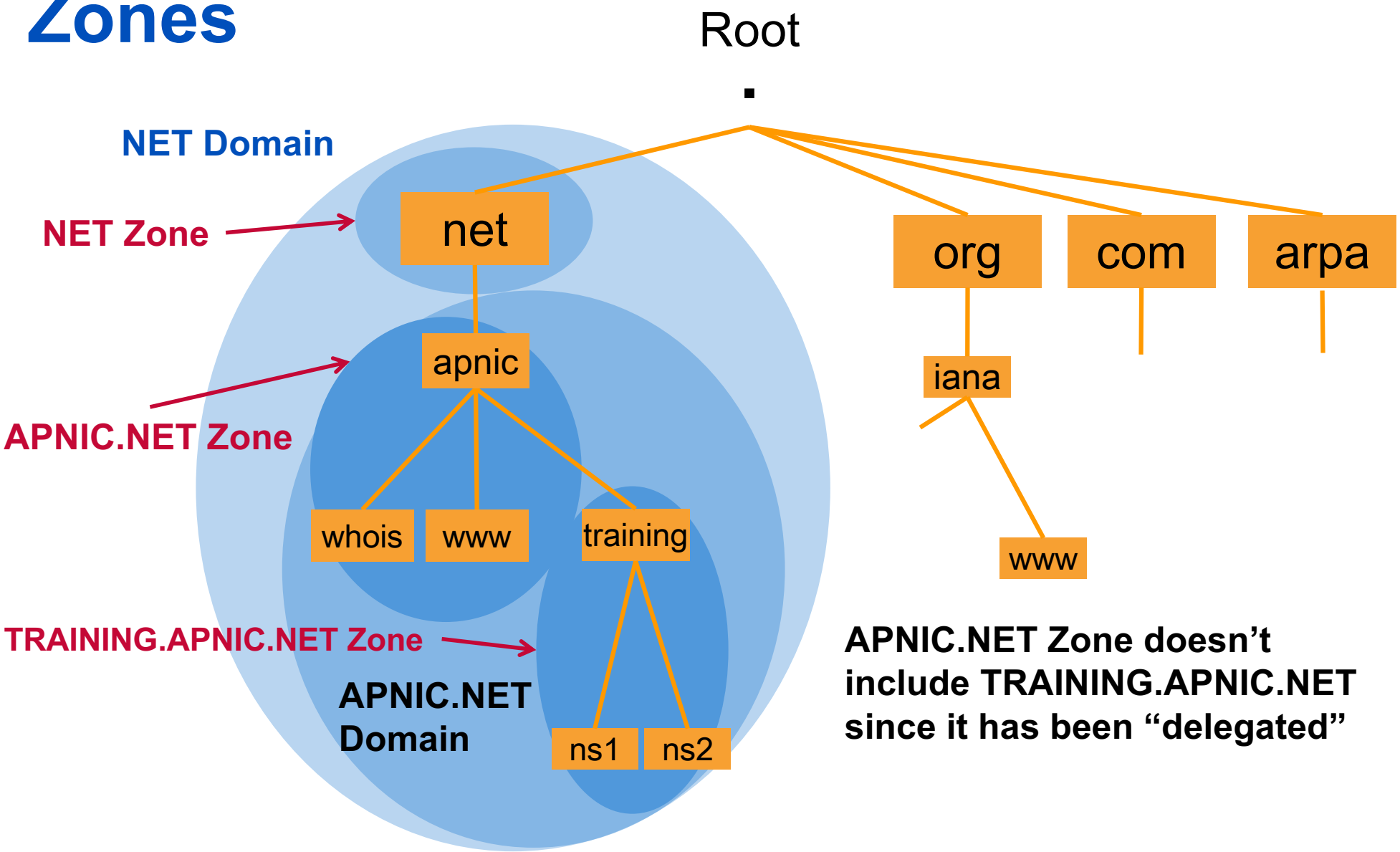
Delegation

- Administrators can create subdomains to group hosts
  - According to geography, organizational affiliation or any other criterion
- An administrator of a domain can delegate responsibility for managing a subdomain to someone else
  - But this isn't required
- The parent domain retains links to the delegated subdomain
  - The parent domain "remembers" to whom the subdomain is delegated

Zones and Delegations

- Zones are “administrative spaces”
- Zone administrators are responsible for a portion of a domain’s name space
- Authority is delegated from parent to child

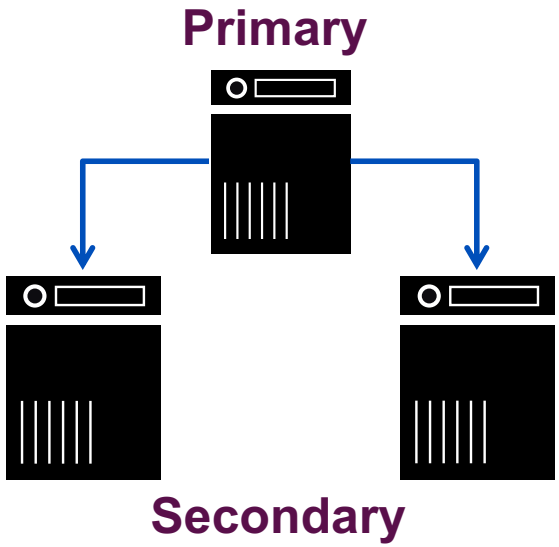
Zones



APNIC.NET Zone doesn't include TRAINING.APNIC.NET since it has been "delegated"

Name Servers

- Name servers answer ‘DNS’ questions
- Several types of name servers
  - Authoritative servers
    - master (primary)
    - slave (secondary)
  - Caching or recursive servers
    - also caching forwarders
- Mixture of functions



Root Servers

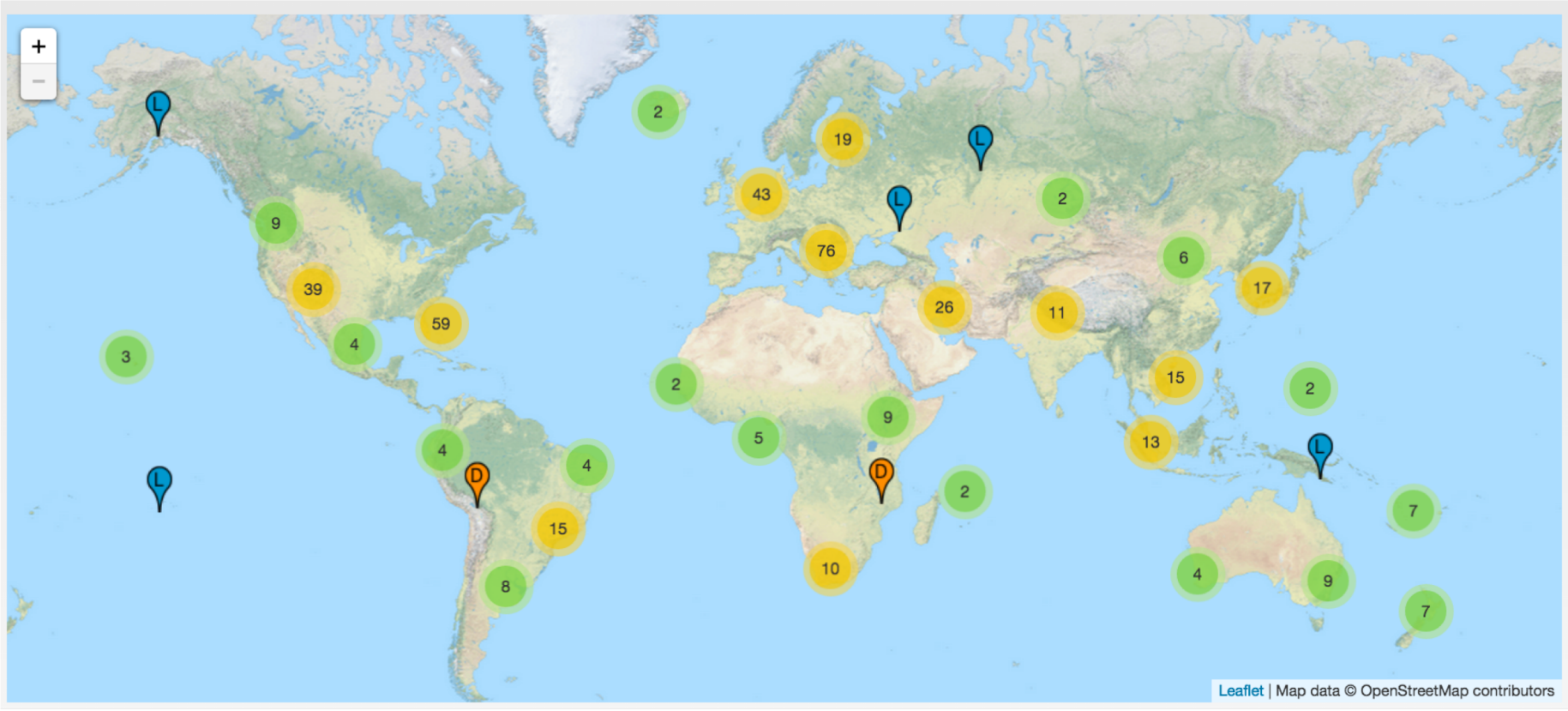
- The top of the DNS hierarchy

- There are 13 root name servers operated around the world

  `[a-m].root-servers.net`

- There are more than 13 physical root name servers
  - Each rootserver has an instance deployed via anycast

Root Servers



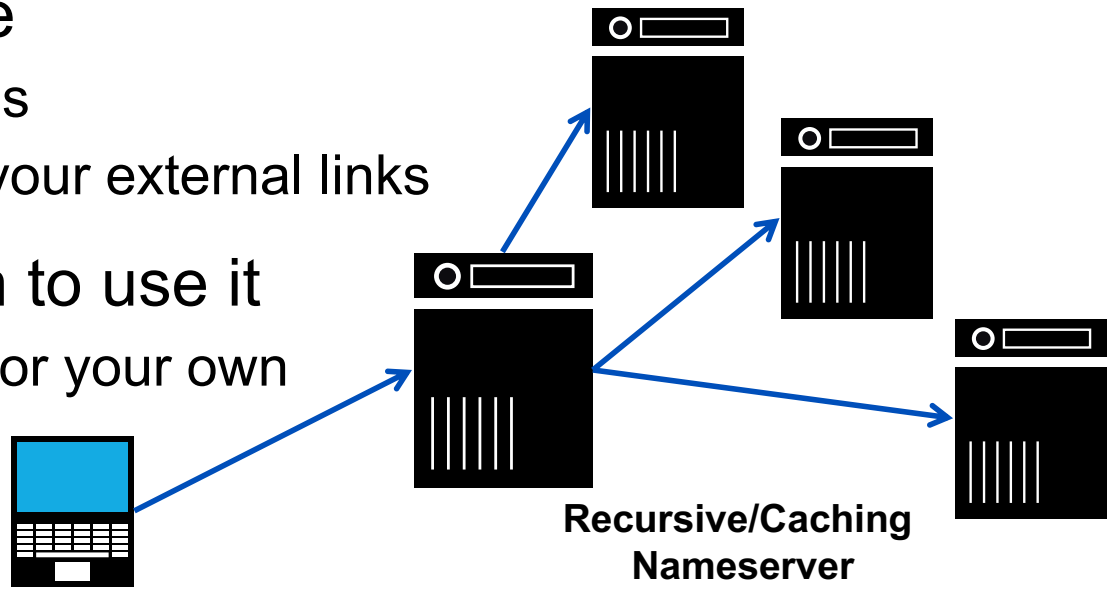
http://root-servers.org/

Root Server Deployment at APNIC

- Started in 2002, APNIC is committed to establish new root server sites in the AP region
- APNIC assists in the deployment providing technical support.
- Deployments of F, K and I-root servers in
  - Singapore, Hong Kong, China, Korea, Thailand, Malaysia, Indonesia, Philippines, Fiji, Pakistan, Bangladesh, Taiwan, Cambodia, Bhutan, and Mongolia

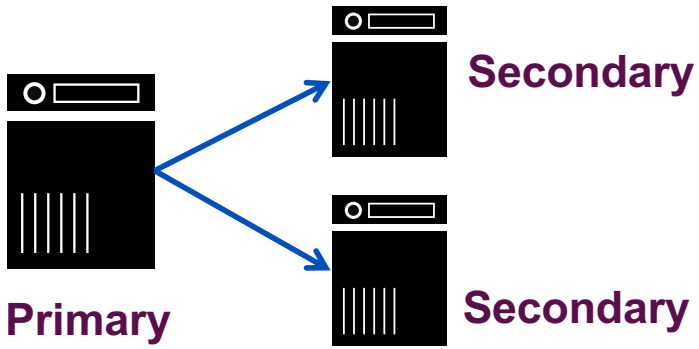
Recursive Nameserver

- The job of the recursive nameserver is to locate the authoritative nameserver and get back the answer
- This process is iterative – starts at the root
- Recursive servers are also usually caching servers
- Prefer a nearby cache
  - Minimizes latency issues
  - Also reduces traffic on your external links
- Must have permission to use it
  - Your ISP's nameserver or your own



Authoritative Nameserver

- A nameserver that is authorised to provide an answer for a particular domain
  - Can be more than one auth nameserver
- Two types based on management method:
  - Primary (Master) and Secondary (Slave)
- Only one primary nameserver
  - All changes to the zone are done in the primary
- Secondary nameserver/s will retrieve a copy of the zonefile from the primary server
  - Slaves poll the master periodically
- Primary server can “notify” the slaves



Resource Records

- Entries in the DNS zone file
- Components:

Resource Record	Function
Label	Name substitution for FQDN
TTL	Timing parameter, an expiration limit
Class	IN for Internet, CH for Chaos
Type	RR Type (A, AAAA, MX, PTR) for different purposes
RDATA	Anything after the Type identifier; Additional data

Common Resource Record Types

RR Type	Name	Functions
A	Address record	Maps domain name to IP address `www.example.com. IN A 192.168.1.1`
AAAA	IPv6 address record	Maps domain name to an IPv6 address `www.example.com. IN AAAA 2001:db8::1`
NS	Name server record	Used for delegating zone to a nameserver `example.com. IN NS ns1.example.com.`
PTR	Pointer record	Maps an IP address to a domain name `1.1.168.192.in-addr.arpa. IN PTR www.example.com.`
CNAME	Canonical name	Maps an alias to a hostname `web IN CNAME www.example.com.`

MX	Mail Exchanger	Defines where to deliver mail for user @ domain
		`example.com. IN MX 10 mail01.example.com.`
		`IN MX 20 mail02.example.com.`

Example: RRs in a zone file

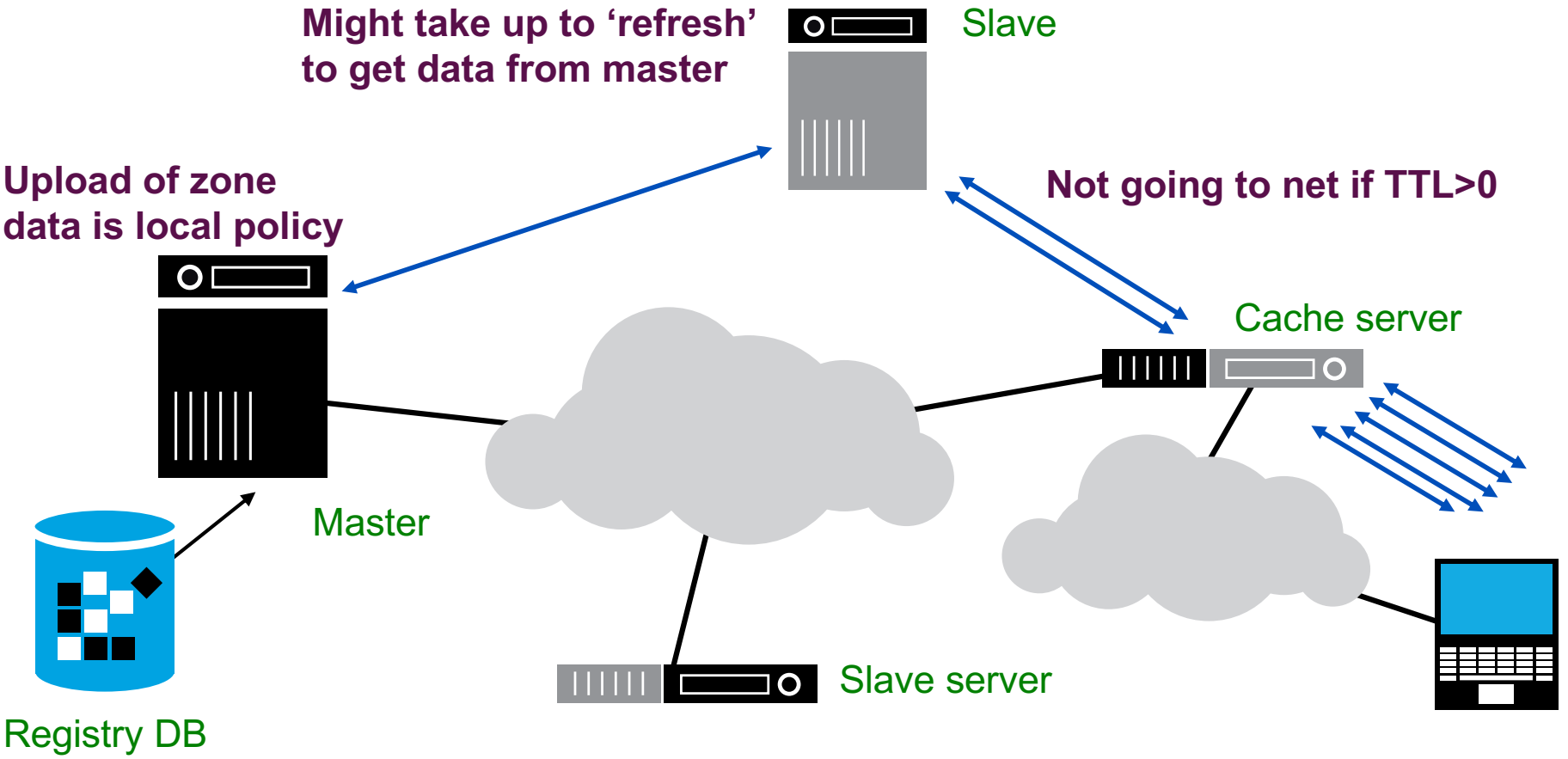
```
apnic.net.      7200 IN     SOA     ns.apnic.net. admin.apnic.net. (
                2018080301          ; Serial
                12h                 ; Refresh 12 hours
                4h                  ; Retry 4 hours
                4d                  ; Expire 4 days
                2h                  ; Negative cache 2 hours )

apnic.net.                  7200    IN      NS      ns.apnic.net.
apnic.net.                  7200    IN      NS      ns.ripe.net.
www.apnic.net.              3600    IN      A       192.168.0.2
www.apnic.net               3600    IN      AAAA    2001:DB8::2
```

Label TTL Class Type Rdata

Places where DNS data lives

Changes do not propagate instantly



Delegating a Zone

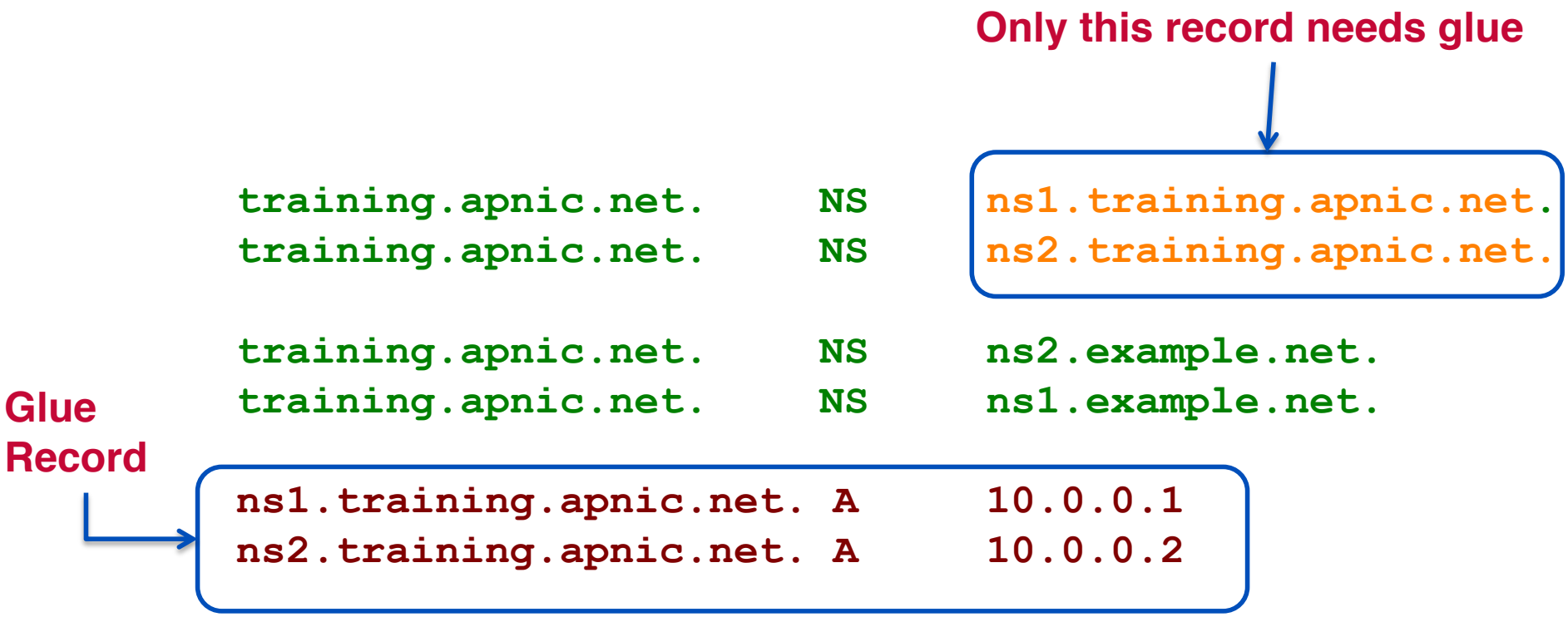
- Delegation is passing of authority for a subdomain to another party
- Delegation is done by adding NS records
  - Ex: if APNIC.NET wants to delegate TRAINING.APNIC.NET

```
training.apnic.net.        NS ns1.training.apnic.net.
training.apnic.net.        NS ns2.training.apnic.net.
```

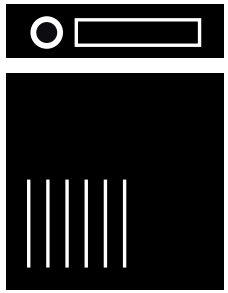
- Now how can we go to ns1 and ns2?
  - We must add a **Glue Record**

Glue Record

- Glue is a ‘non-authoritative’ data
- Don’t include glue for servers that are not in the sub zones

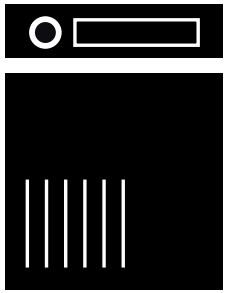


Delegating training.apnic.net. from apnic.net.



ns.apnic.net

1. Add NS records and glue
2. Make sure there is no other data from the training.apnic.net. zone in the zone file



ns.training.apnic.net

1. Setup minimum two servers
2. Create zone file with NS records
3. Add all training.apnic.net data

Resolver

- A piece of software (usually part of the operating system) which formats the DNS request into UDP packets
- A stub resolver is a minimal resolver that forwards all requests to a local recursive nameserver
  - The IP address of the local DNS server is configured in the resolver.
- Every host needs a resolver
  - In Linux, it uses /etc/resolv.conf
- It is always a good idea to configure more than one nameserver

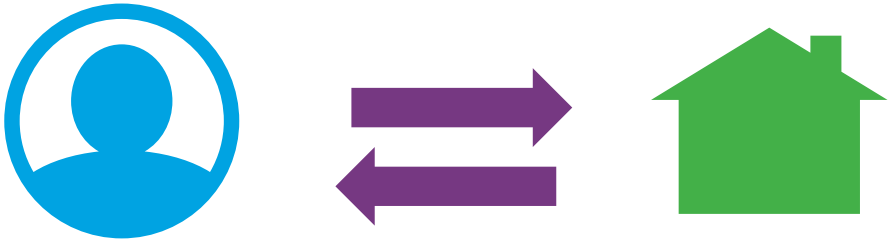
Remember ...

- Deploy multiple authoritative servers to distribute load and risk
  - Put your name servers apart from each other
- Use cache to reduce load to authoritative servers and response times
- SOA timers and TTL need to be tuned to the needs of the zone
  - For stable data, use higher numbers

What is ‘Reverse DNS’?

- ‘Forward DNS’ maps names to numbers
  - svc00.apnic.net ➔192.168.1.100
  - svc00.apnic.net ➔2001:DB8::1

- ‘Reverse DNS’ maps numbers to names
  - 192.168.1.100 ➔ svc00.apnic.net
  - 2001:DB8::1 ➔ svc00.apnic.net

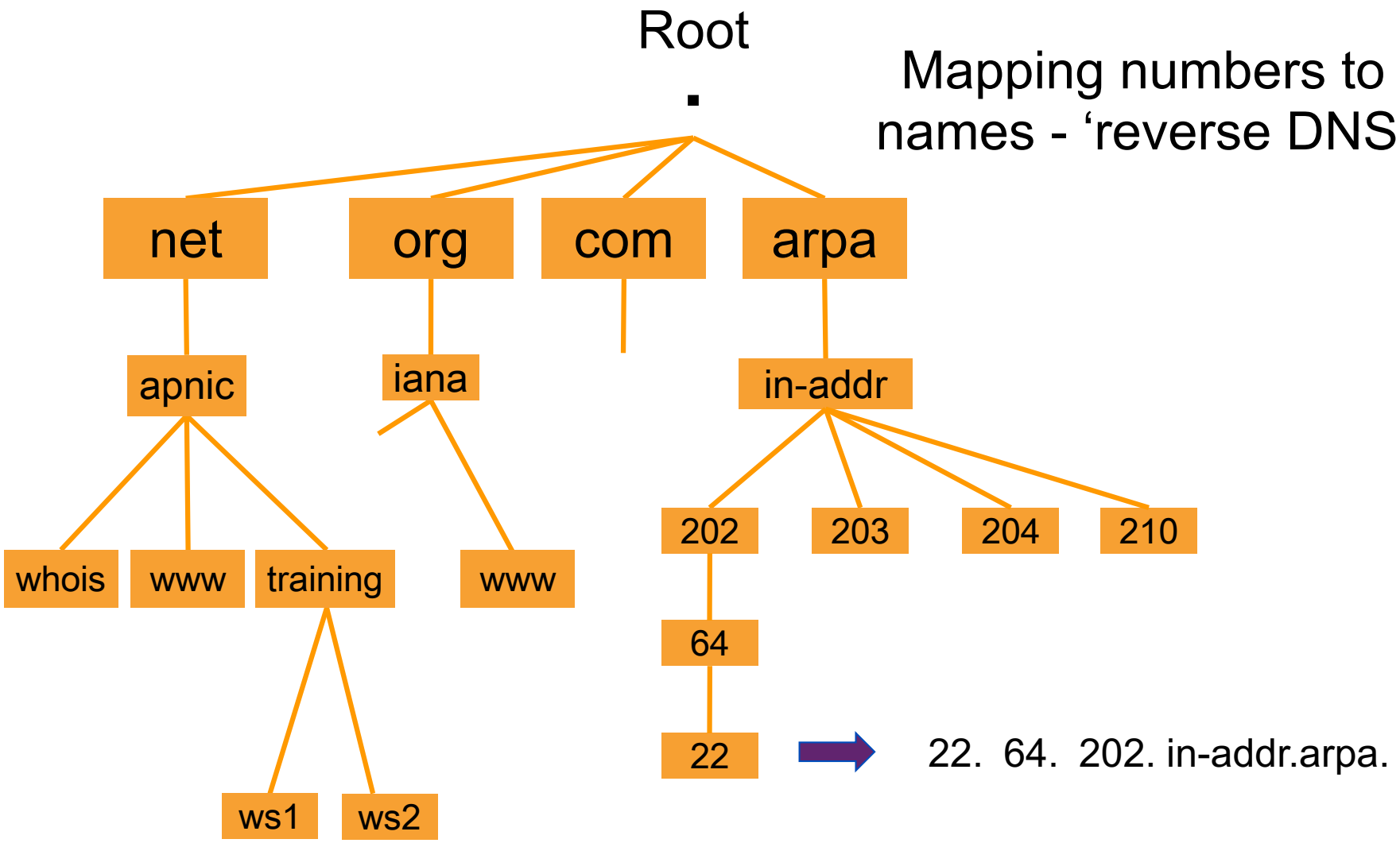


Person (Host) Address (IPv4/IPv6)

Reverse DNS - why bother?

- Service denial
  - only allow access when fully reverse delegated
  - Example: anonymous ftp
- Diagnostics
  - Used in tools such as traceroute
- Spam identifications
  - Failed reverse lookup results in a spam penalty score
- Registration responsibilities
  - APNIC members must make sure that all their address space are properly reverse delegated

Principles – DNS Tree



Mapping numbers to names - ‘reverse DNS’

22. 64. 202. in-addr.arpa.

Creating Reverse Zones

- Same as creating a forward zone file
  - SOA and initial NS records are the same as forward zone
- Create additional PTR records
- In addition to the forward zone files, you need the reverse zone files
  - Ex: for a reverse zone on a 203.176.189.0/24 block, create a zone file and name it as “db.203.176.189” (make it descriptive)

Pointer (PTR) Records

- Create pointer (PTR) records for each IP address

```
131.28.12.202.in-addr.arpa. IN PTR svc00.apnic.net.
```

or

```
131             IN      PTR             svc00.apnic.net.
```

Reverse Zone Example

```
$ORIGIN 1.168.192.in-addr.arpa.
@       3600  IN SOA test.company.org. (
                          sys\.admin.company.org.
                          2018080301          ; serial
                          1h                  ; refresh
                          30M                 ; retry
                          1W                  ; expiry
                          3600 )              ; neg. answ. ttl

        NS        ns.company.org.
        NS        ns2.company.org.

1       PTR       gw.company.org.
                  router.company.org.
2       PTR       ns.company.org.
```

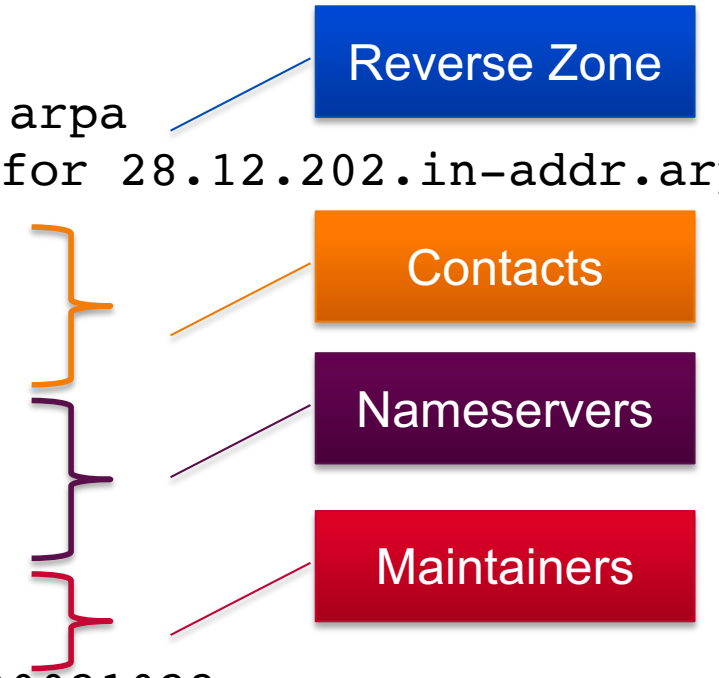
Reverse Delegation

- /24 Delegations
  - Address blocks should be assigned or allocated
  - At least two name servers
- /16 Delegations
  - Same as /24 delegations
  - APNIC delegates entire zone to member
- < /24 Delegations
  - Read “Classless IN-ADDR.ARPA delegation” (RFC 2317)

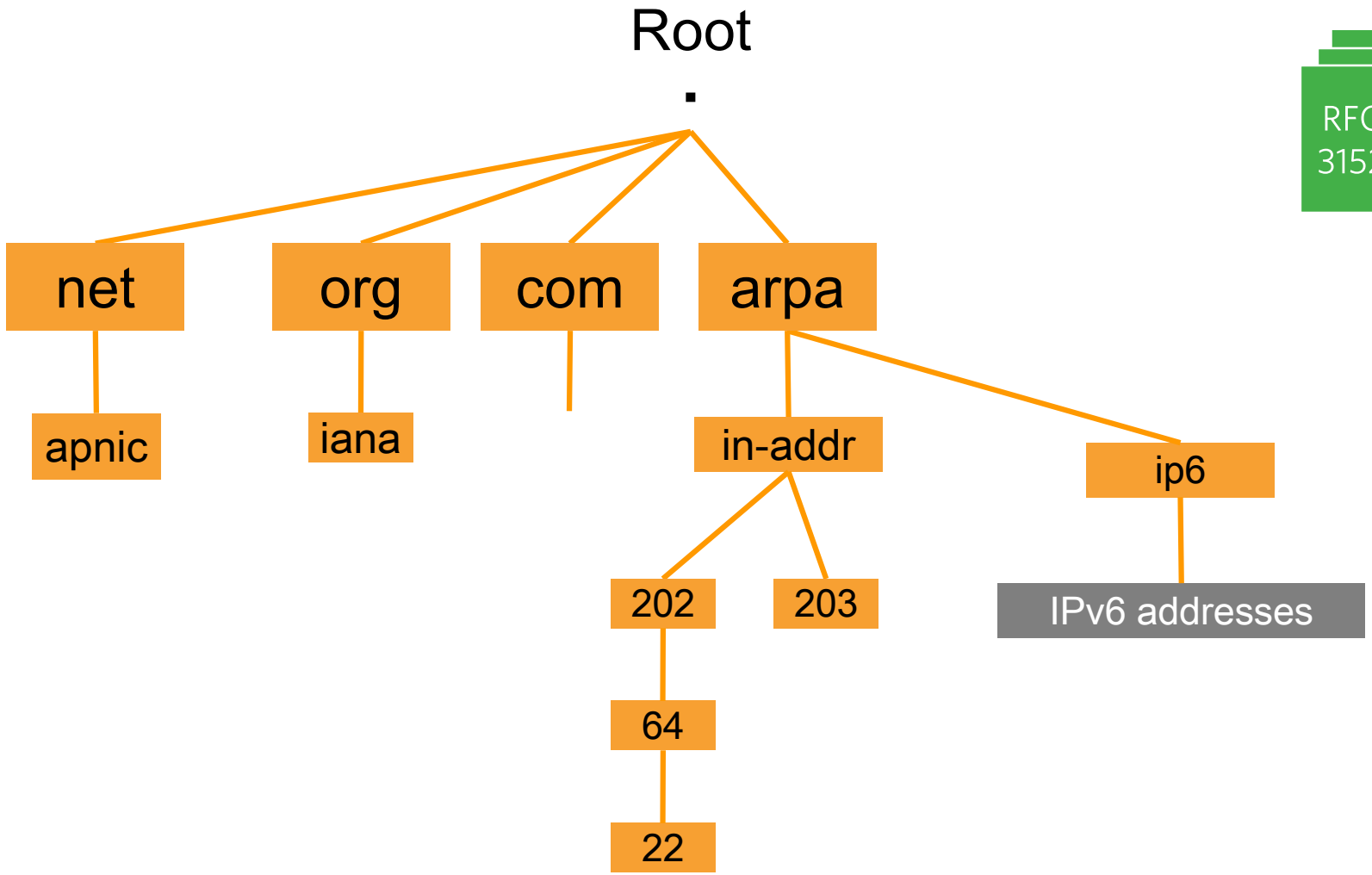


Whois domain object

```
domain:     28.12.202.in-addr.arpa
Descr:      in-addr.arpa zone for 28.12.202.in-addr.arpa
admin-c:    NO4-AP
tech-c:     AIC1-AP
zone-c:     NO4-AP
nserver:    cumin.apnic.net
nserver:    tinnie.apnic.net
nserver:    tinnie.arin.net
mnt-by:     MAINT-APNIC-AP
mnt-lower:  MAINT-AP-DNS
changed:    inaddr@apnic.net 20021023
changed:    inaddr@apnic.net 20040109
changed:    hm-changed@apnic.net 20091007
changed:    hm-changed@apnic.net 20111208
source:     APNIC
```



Reverse DNS Tree – with IPv6



IPv6 Representation in the DNS

- Forward lookup support: Multiple RR records for name to number
  - AAAA (Similar to A RR for IPv4)

- Reverse lookup support:
  - Reverse nibble format for zone ip6.arpa

IPv6 Reverse Lookups – PTR records

- Similar to the IPv4 reverse record

```
b.a.9.8.7.6.5.0.4.0.0.0.3.0.0.0.2.0.0.0.1.0.0.0.0.0.0.0.1.2.3.4.ip6.arpa.
        IN    PTR test.ip6.example.com.
```

- Example: The reverse name lookup for a host with address

```
3ffe:8050:201:1860:42::1

$ORIGIN 0.6.8.1.1.0.2.0.0.5.0.8.e.f.f.3.ip6.arpa.
1.0.0.0.0.0.0.0.0.0.0.0.0.2.4.0.0  14400  IN PTR
host.example.com.
```

IPv6 Forward Lookups

- Multiple addresses possible for any given name
  - Ex: in a multi-homed situation
- Can assign A records and AAAA records to a given name/domain
- Can also assign separate domains for IPv6 and IPv4

Example: Forward Zone

```
;; domain.edu

$TTL          86400

@     IN      SOA     ns1.domain.edu. root.domain.edu. (

                            2015050501                ; serial - YYYYMMDDXX

                            21600                     ; refresh - 6 hours

                            1200                      ; retry - 20 minutes

                            3600000                   ; expire - long time

                            86400)                    ; minimum TTL - 24 hours

;; Nameservers

                            IN          NS            ns1.domain.edu.

                            IN          NS            ns2.domain.edu.

;; Hosts with just A records

host1                       IN          A             1.0.0.1

;; Hosts with both A and AAAA records

host2                       IN          A             1.0.0.2

                            IN          AAAA          2001:468:100::2
```

Example: Reverse Zone

```
;; 0.0.0.0.0.0.1.0.8.6.4.0.1.0.0.2.rev
;; These are reverses for 2001:468:100::/64)

;; File can be used for both ip6.arpa and ip6.int.

$TTL            86400

@     IN       SOA      ns1.domain.edu. root.domain.edu. (
                                2015050501          ; serial - YYYYMMDDXX
                                21600               ; refresh - 6 hours
                                1200                ; retry - 20 minutes
                                3600000             ; expire - long time
                                86400)              ; minimum TTL - 24 hours

;; Nameservers
                      IN        NS        ns1.domain.edu.
                      IN        NS        ns2.domain.edu.
1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0      IN        PTR       host1.ip6.domain.edu
2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0      IN        PTR       host2.domain.edu
```


Questions

DNS Security Concepts

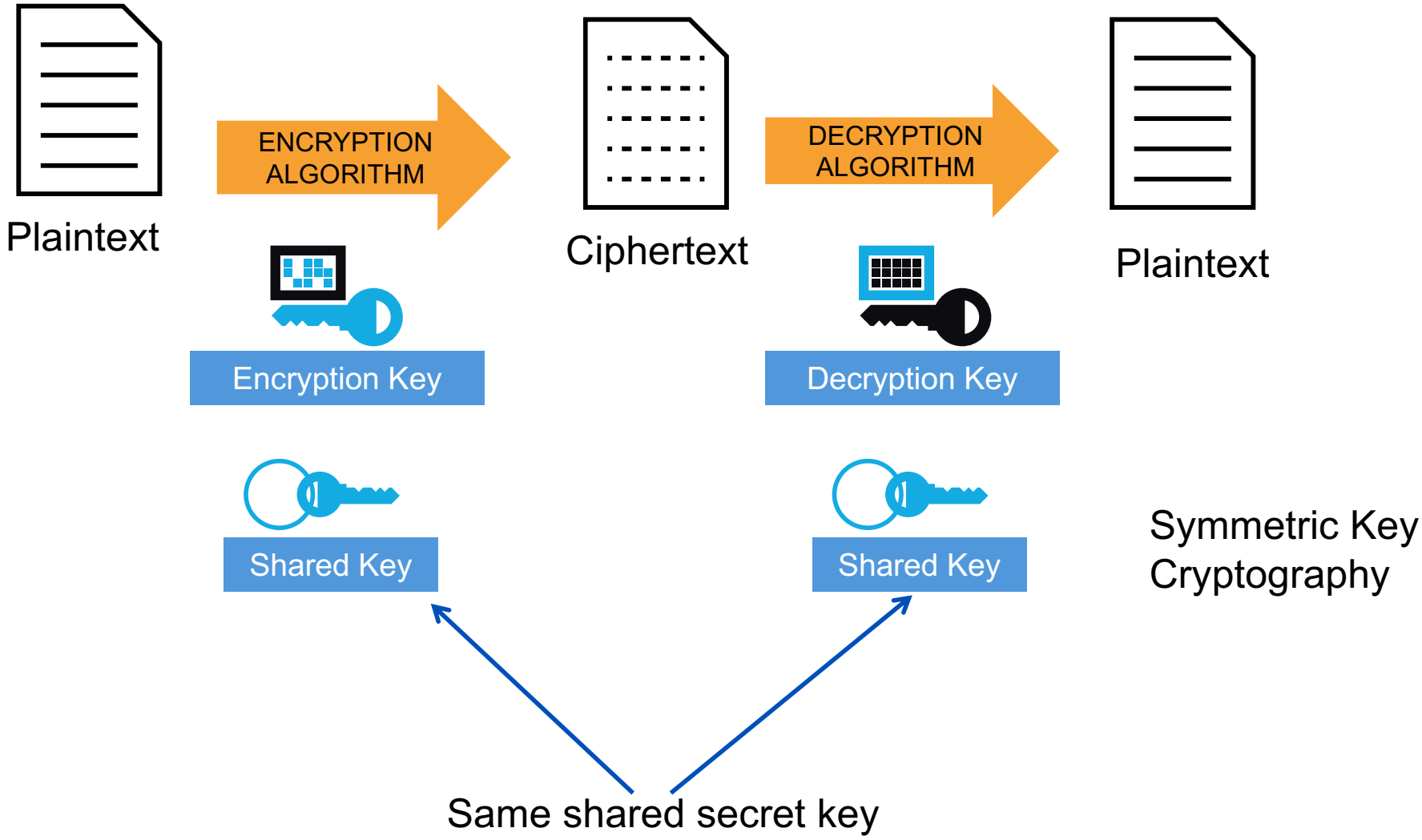
Crypto Review

- Most security applications use crypto algorithms
  - Symmetric key
  - Public key crypto
  - One-way hash functions

Symmetric Key Crypto

- Uses a single key to encrypt and decrypt data
- Also known as a secret-key or private key algorithm
- The key must be kept a “secret” to maintain security
- key lengths ranging from 40 to 256 bits
- Examples of symmetric key algorithms:
  - DES, 3DES, AES, IDEA, RC5, RC6, Blowfish

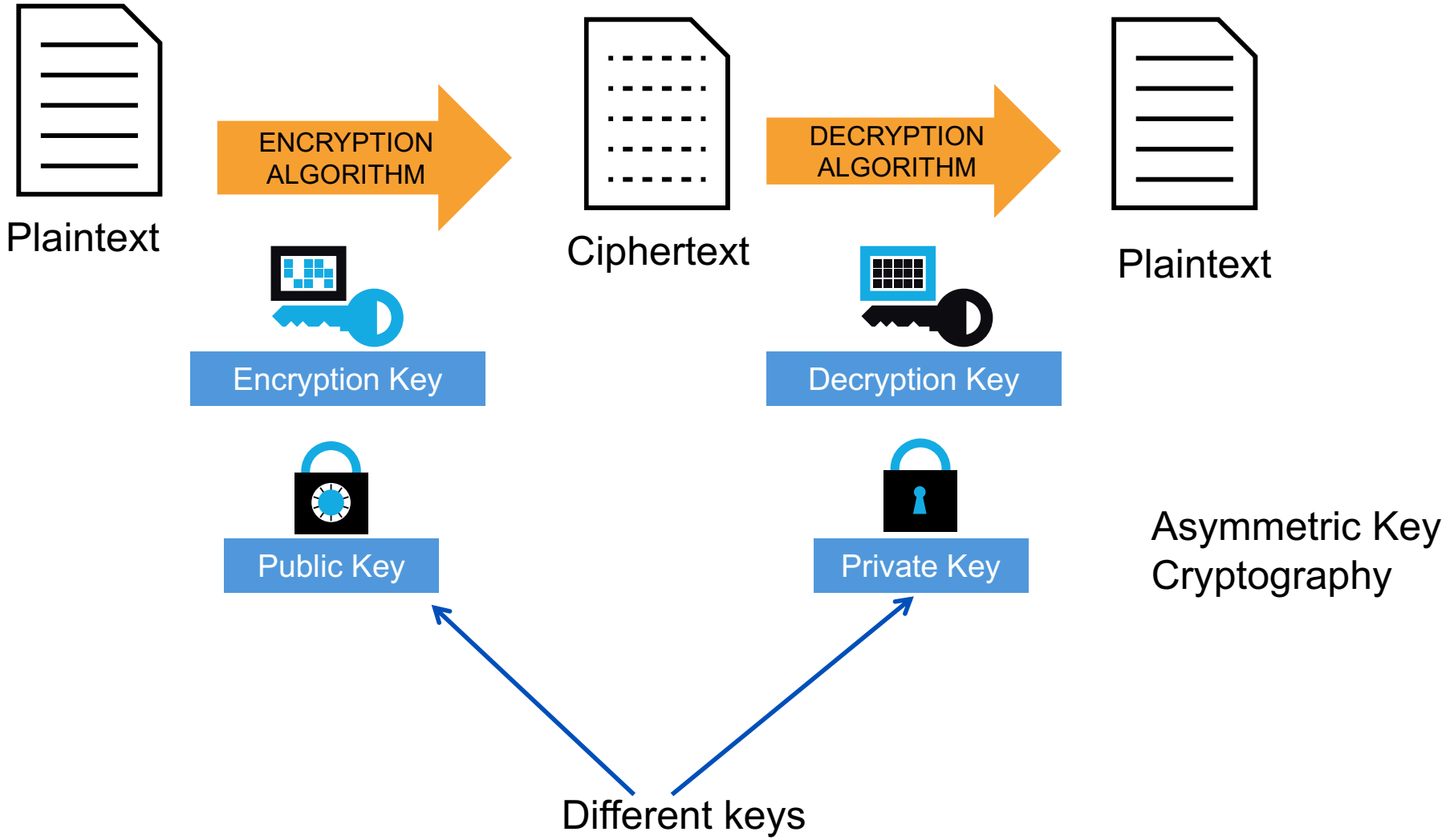
Symmetric Encryption



Asymmetric Key Crypto

- Uses a public-private keypair
- Also called public key crypto
- Use one key to sign data, then the other key to verify
- Examples:
  - RSA, DSA, El Gamal, Diffie-Hellman, PKCS

Asymmetric Encryption



Hash Functions

- produces a condensed representation of a message
- takes an input message of arbitrary length and outputs fixed-length code
  - The fixed-length output is called the **hash** or **message digest**
- A form of signature that uniquely represents the data
- Uses:
  - Verifying file integrity - if the hash changes, it means the data is either compromised or altered in transit.
  - Digitally signing documents
  - Hashing passwords

Hash Functions

- Message Digest (MD) Algorithm
  - Outputs a 128-bit fingerprint of an arbitrary-length input
  - MD4 is obsolete, MD5 is widely-used

- Secure Hash Algorithm (SHA)
  - SHA-1 produces a 160-bit message digest similar to MD5
  - Widely-used on security applications (TLS, SSL, PGP, SSH, S/MIME, IPsec)
  - SHA-256, SHA-384, SHA-512 can produce hash values that are 256, 384, and 512-bits respectively

Digital Signature

- a message appended to a packet
- used to prove the identity of the sender and the integrity of the packet
- how it works:
  - sender signs the message with own private key
  - receiver uses the sender's public key to verify the signature

Message Authentication Code

- Provides integrity and authenticity
- How it works:
  - In the sender side, the message is passed through a MAC algorithm to get a MAC (or Tag)
  - In the receiver side, the message is passed through the same algorithm
  - The output is compared with the received tag and should match
- Uses the same secret key
- Can also use hash function to generate the MAC, called Hash-based Message Authentication Code (HMAC)

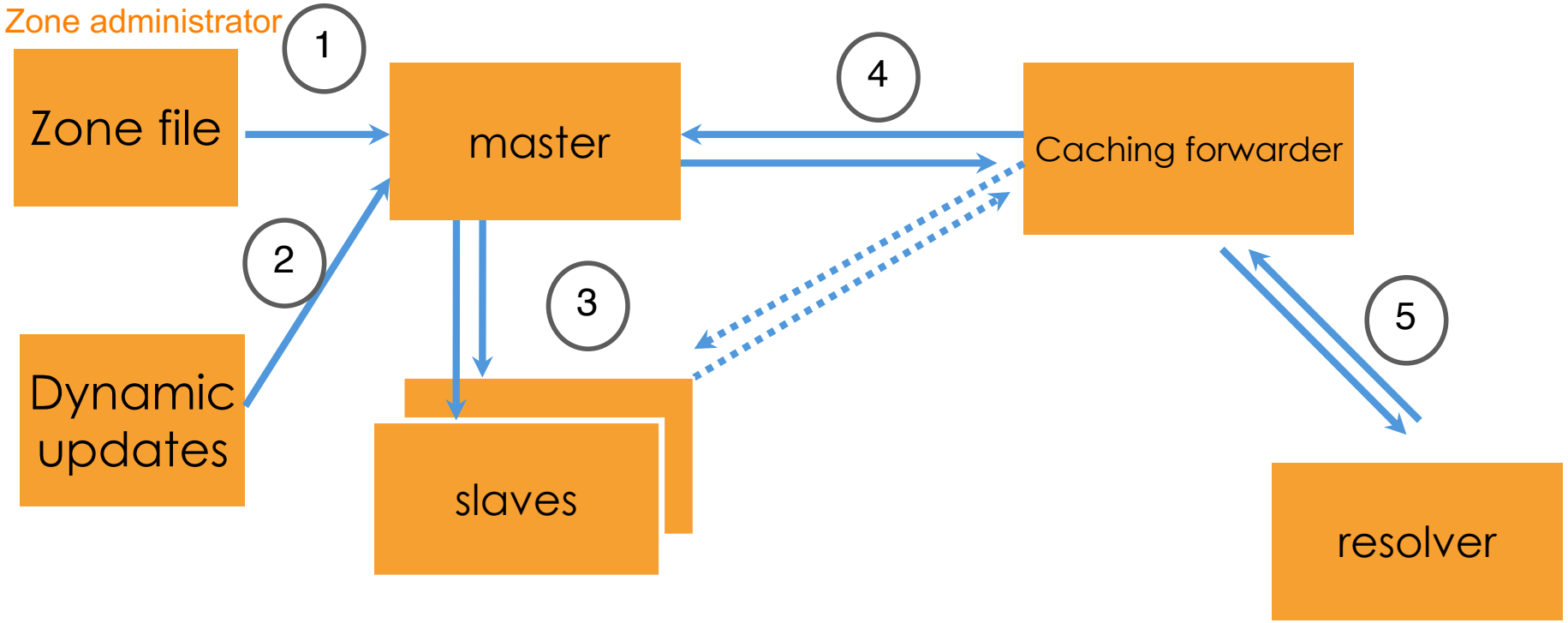
DNS Security - Background

- The original DNS protocol wasn't designed with security in mind
- As the Internet grows, it has become less trustworthy
- Some security problems:
  - Using reverse DNS to impersonate hosts
  - Software bugs (buffer overflows, bad pointer handling)
  - Cache poisoning (putting inappropriate data into the cache)

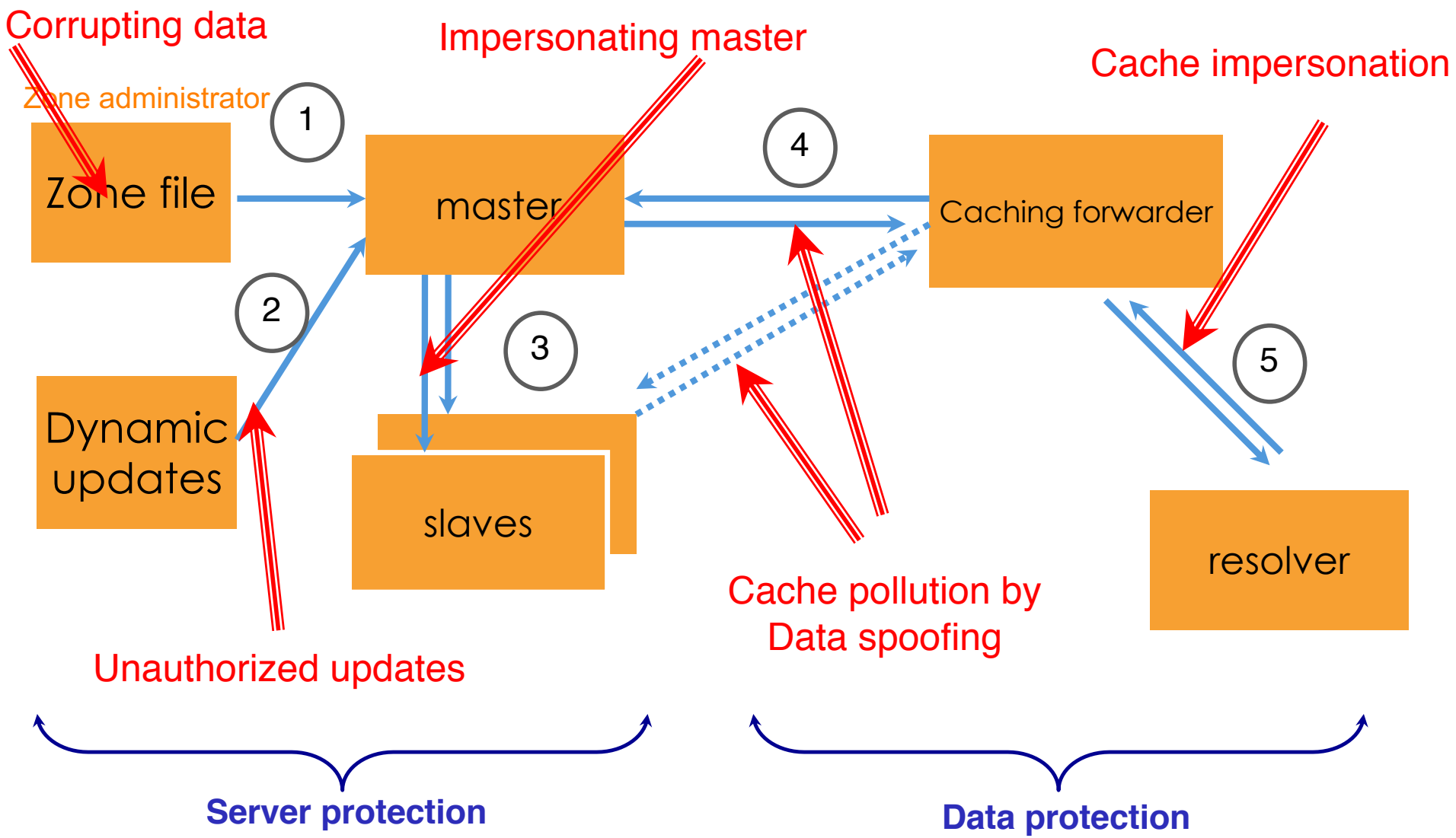
DNS Protocol Vulnerability

- DNS data can be corrupted as it transfers between primary server, resolver or forwarder
- There is no way to check the validity of DNS data
  - Resolver implementation can be exploited (predictable transaction ID, buffer overflow, pointer handling)
  - Caching forwarders can be polluted
  - Corrupted DNS data might end up in caches and stay there for a long time
- DNS transactions can be compromised
  - Primary server sending data to wrong secondary server

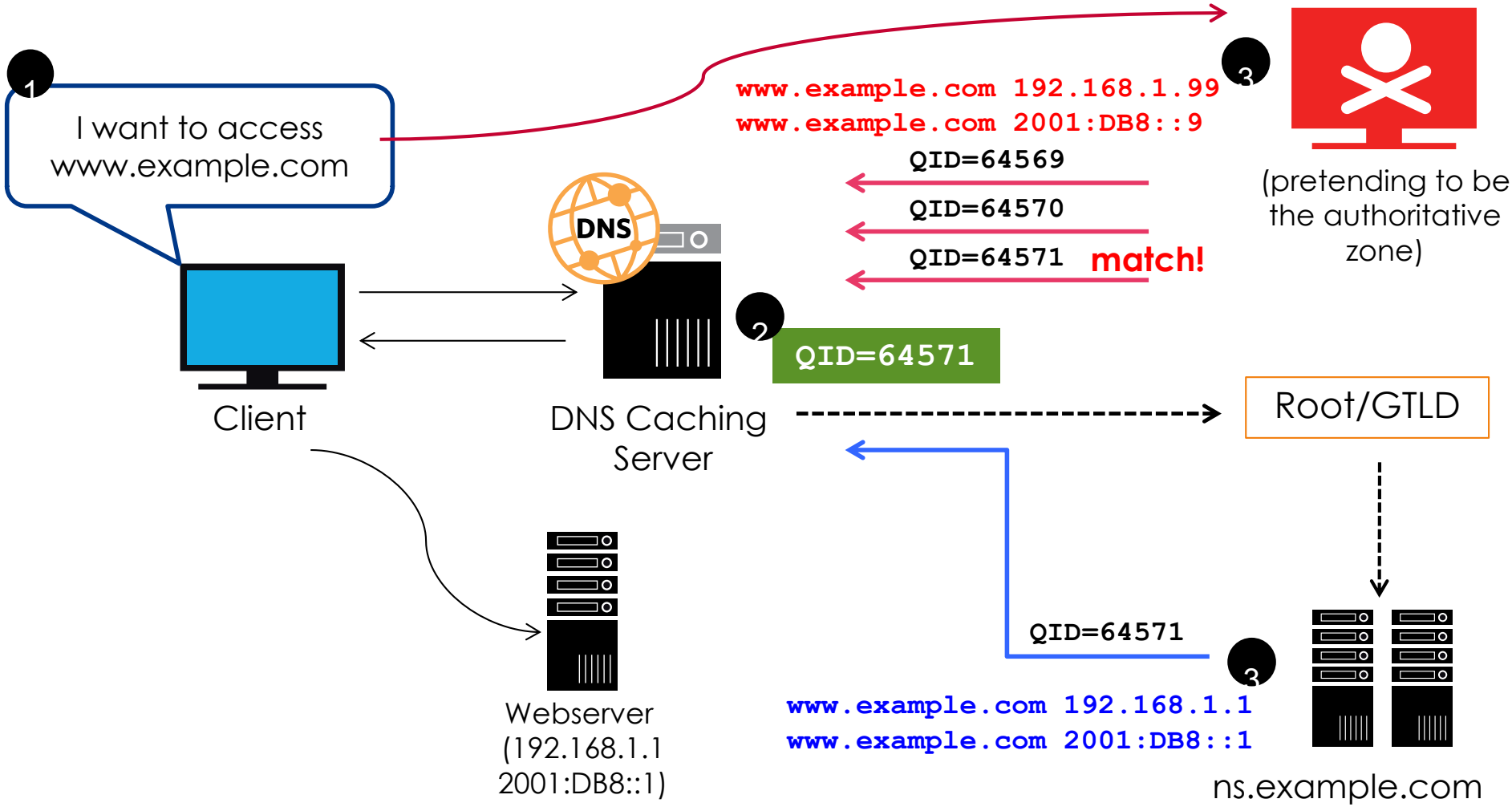
DNS: Data Flow



DNS Vulnerabilities



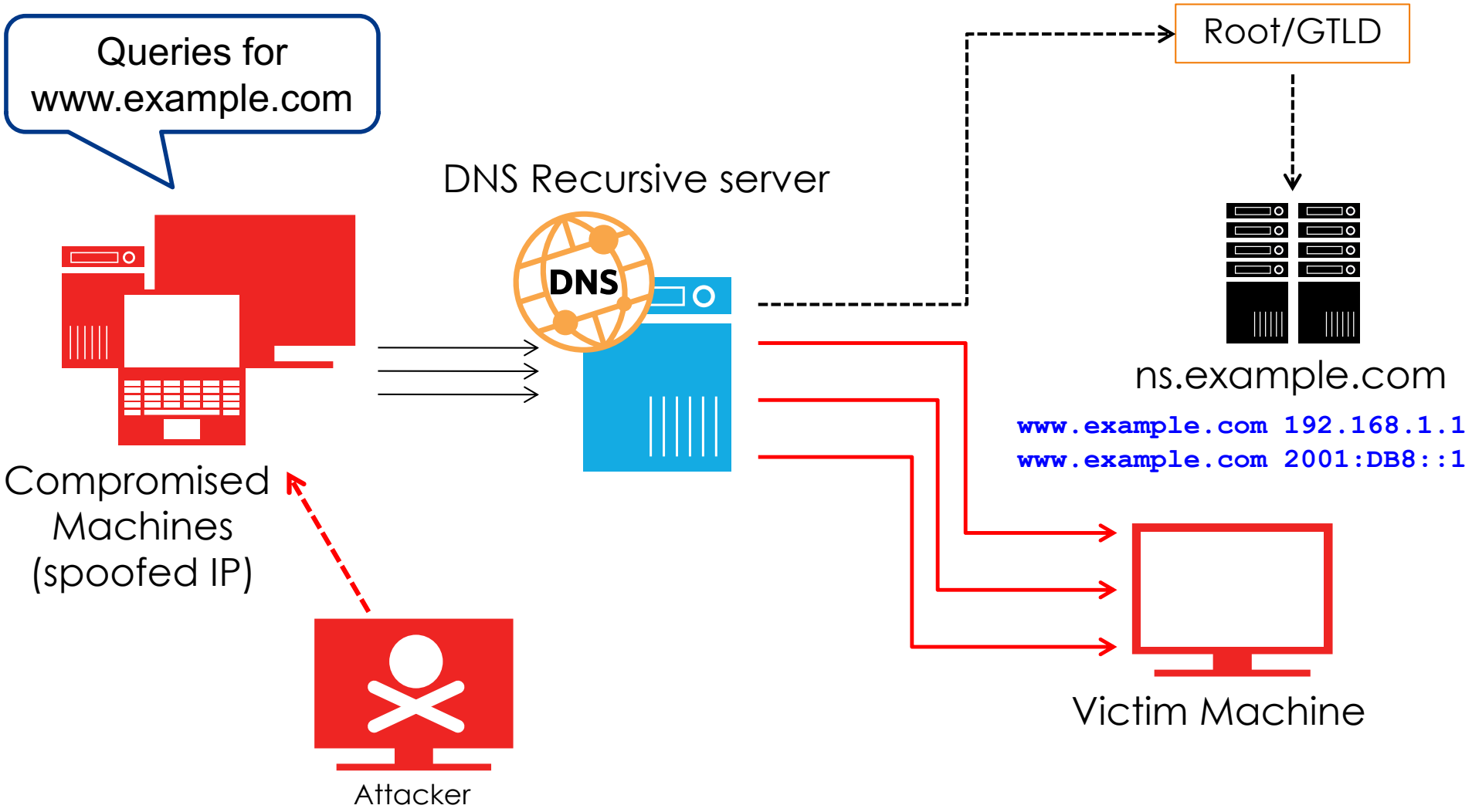
DNS Cache Poisoning



DNS Amplification

- A type of reflection attack combined with amplification
  - Source of attack is reflected off another machine
  - Traffic received is bigger (amplified) than the traffic sent by the attacker
- UDP packet's source address is spoofed

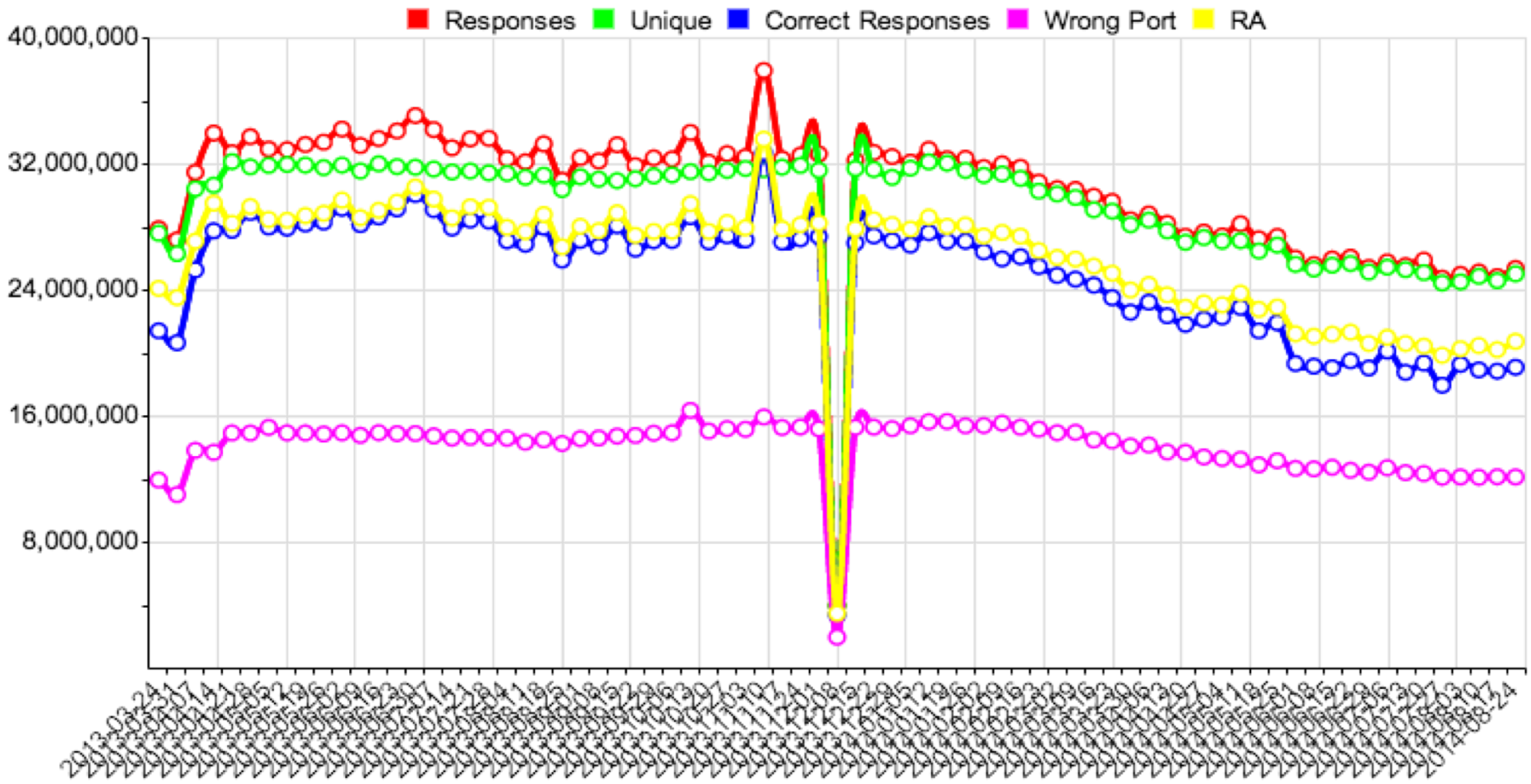
DNS Amplification



Open Resolvers

- DNS servers that answer recursive queries from any host on the Internet
  - pose some “significant threat” to the global network infrastructure
- Often used in DNS-based DDoS attacks
- There’s a project that maps out open resolvers on the Internet
  - Open Resolver Project - http://openresolverproject.org/
- Some utility available to check if running an open resolver

Open Resolvers



Reference: Open Resolver Project

Open Resolvers Statistics

This table shows the number of known open resolvers for each autonomous system as of Wed Aug 27 06:00:00 UTC 2014.

count	asn	name
1154	4766	KIXS-AS-KR Korea Telecom,KR
1080	4713	OCN NTT Communications Corporation,JP
1021	3462	HINET Data Communication Business Group,TW
728	7418	TELEFÃ“NICA CHILE S.A.,CL
621	1659	ERX-TANET-ASN1 Tiawan Academic Network (TANet) Information C
562	10834	Telefonica de Argentina,AR
521	2516	KDDI KDDI CORPORATION,JP
512	3786	LGDACOM LG DACOM Corporation,KR
423	4538	ERX-CERNET-BKB China Education and Research Network Center,C
391	9318	HANARO-AS Hanaro Telecom Inc.,KR
378	701	UUNET - MCI Communications Services, Inc. d/b/a Verizon Busi
371	2514	INFOSPHERE NTT PC Communications, Inc.,JP
347	36692	OPENDNS - OpenDNS, LLC,US
337	4134	CHINANET-BACKBONE No.31,Jin-rong Street,CN
332	15418	FASTHOSTS-INTERNET Fasthosts Internet Ltd. Gloucester, UK.,G
328	14992	CRYSTALTECH - CrystalTech Web Hosting Inc.,US
324	3320	DTAG Deutsche Telekom AG,DE
310	8167	Brasil Telecom S/A - Filial Distrito Federal,BR
297	21844	THEPLANET-AS - ThePlanet.com Internet Services, Inc.,US
292	7018	ATT-INTERNET4 - AT&T Services, Inc.,US
268	4323	TWTC - tw telecom holdings, inc.,US
265	4837	CHINA169-BACKBONE CNCGROUP China169 Backbone,CN
255	17813	MTNL-AP Mahanagar Telephone Nigam Ltd.,IN
243	209	ASN-QWEST - Qwest Communications Company, LLC,US
237	6400	CompaÃ±Ãa Dominicana de TelÃ©fonos, C. por A. - CODETEL,DO
223	8151	Uninet S.A. de C.V.,MX
215	20746	ASN-IDC Telecomitalia s.p.a.,IT
202	19429	ETB - Colombia,CO
199	13768	PEER1 - Peer 1 Network (USA) Inc.,US
192	22773	ASN-CXA-ALL-CCI-22773-RDC - Cox Communications Inc.,US

Source: DNS Measurement Factory

DNS Changer

- “Criminals have learned that if they can control a user’s DNS servers, they can control what sites the user connects to the Internet.”
- How: infect computers with a malicious software (malware)
- A malware changes the user’s DNS settings with that of the attacker’s DNS servers
- Points the DNS configuration to DNS resolvers in specific address blocks and use it for their criminal enterprise

Source: DCWG

DNS Hijacking

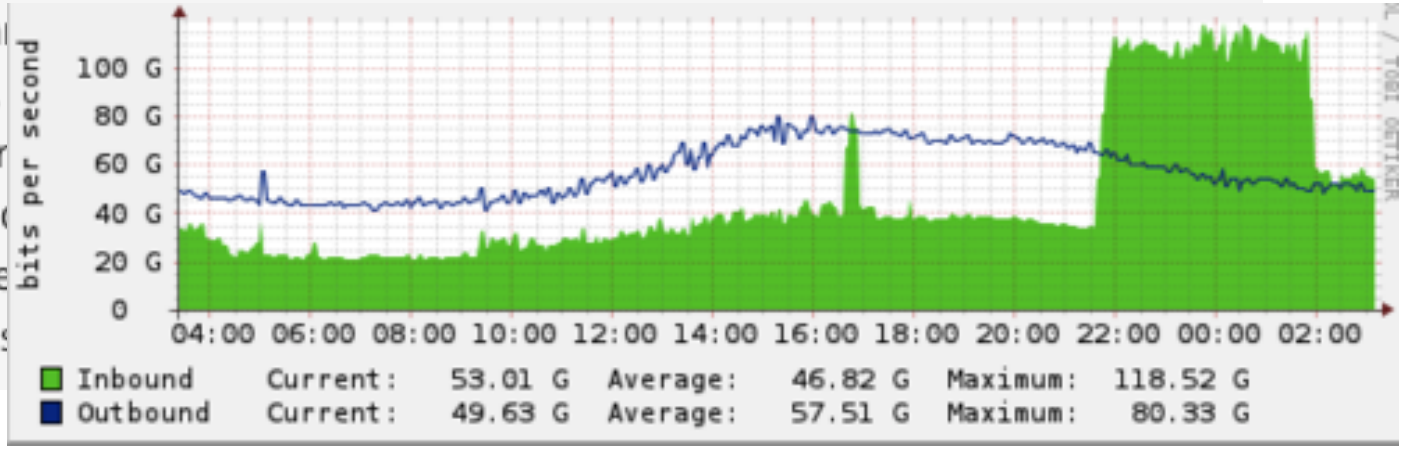
- Also called DNS redirection
- Can be achieved when
  - User's DNS settings has been modified through malware
  - DNS server has been compromised to provide incorrect responses

DNS-Based DDoS attacks are common and remarkably simple

Case: Attack at Spamhaus

In the Spamhaus case, the attacker was sending requests for the DNS zone file for ripe.net to open DNS resolvers. The attacker spoofed the CloudFlare IPs we'd issued for Spamhaus as the source in their DNS requests. The open resolvers responded with DNS zone file, generating collectively approximately 75Gbps of attack traffic. The requests were likely approximately 36 bytes long (e.g. dig ANY ripe.net @X.X.X.X +edns=0 +bufsize=4096, where X.X.X.X is replaced with the IP address of an open DNS resolver) and the response was approximately 3,000 bytes, translating to a 100x amplification factor.

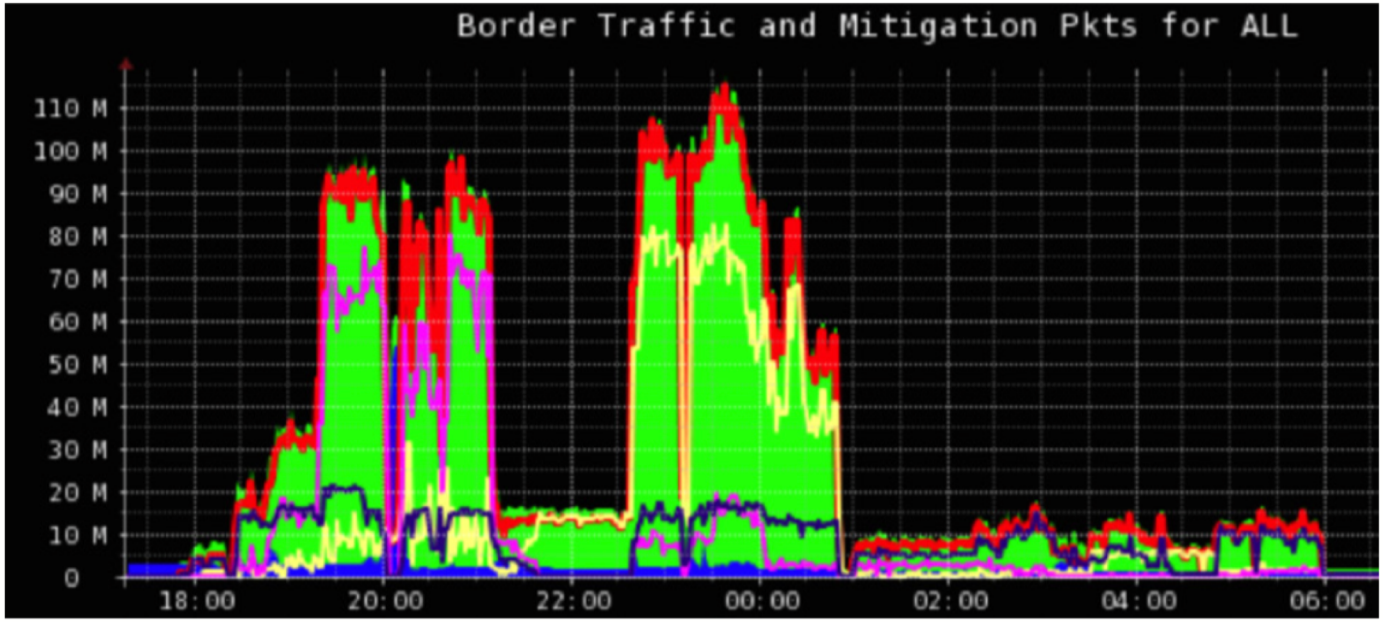
We recorded over 30,000 unique DNS resolvers involved in the attack. This translates to each open DNS resolver sending a
radar of most DNS resolvers.
needed to control a botnet or
a small sized botnet or a hand
are the scourge of the Interne
service providers take seriou



http://blog.cloudflare.com/deep-inside-a-dns-amplification-ddos-attack
http://blog.cloudflare.com/the-ddos-that-knocked-spamhaus-offline-and-ho

Case: DNS Query Floods

- (May 2014) Targeted a chat service provider under Akamai
- Bandwidth used maxed at 119 Gbps
- Resulted to 110 Mpps – one of the highest packet-per-second (pps) rate for Akamai in 2014



Source: Prolexic Q2 2014 Global Attack Report

Case: DDoS attack on DNS Provider NS1

But the new attacks have been entirely different. The sources of the attacks shifted over the week, cycling between bots (likely running on compromised systems) in eastern Europe, Russia, China, and the United States. And the volume of the attacks increased to the 30Gbps to 50Gbps range. While the attacks rank in the "medium" range in total volume, and are not nearly as large as previous huge amplification attacks, they were tailored specifically to degrading the response of NS1's DNS structure.

Rather than dumping raw data on NS1's servers with amplification attacks—where an attacker sends spoofed DNS requests to open DNS servers that will result in large blocks of data being sent in the direction of the target—the attackers sent programmatically generated DNS lookup requests to NS1's name servers, sometimes at rates of 50 million to 60 million packets per second. The packets looked superficially like genuine requests, but they were for resolution of host names that don't actually exist on NS1's customers' networks.

http://arstechnica.com/information-technology/2016/05/major-dns-provider-hit-by-mysterious-focused-ddos-attack/

Why is DNS prone to DDoS attacks?

DNS uses UDP

- UDP = best effort, connectionless transmission
- Easy to spoof the source address
- Similar case with NTP, SNMP, SSDP, Chargen protocols

Each query returns large responses

- EDNS0 allows DNS messages to carry bigger data
- DNSSEC returns large replies

It's usually open to all

- Open resolvers

Basic DNS Security Practices

- Run the most recent version of the DNS software or apply the latest patch
- Restrict queries
- Prevent unauthorized zone transfers
- Run BIND with the least privilege (use `chroot`)
- Randomize source ports
- Secure the box
- Implement TSIG and DNSSEC

DNS DDoS Mitigation

- Set up monitoring to know when you are being attacked
  - Use previous statistics to know your baseline load
- Avoid single point of failure
  - DNS server, router, firewall, uplinks, etc
  - Authoritative nameservers must be geographically distributed
- Provision for your DNS infrastructure
  - Find your DNS capacity (using tools like dnsperf)
  - Be ready to deploy more as needed
- Deploy anycast
  - Attack is isolated in one group at a time
  - Alternatively use cloud-based DNS providers
- Don’t run an open resolver!

Response Rate Limiting (RRL)

- Protects against DNS amplification attack
- Implemented in CZ-NIC Knot (v1.2-RC3), NLNetLabs NSD (v3.2.15), and ISC BIND 9 (v9.9.4) release

```
rate-limit {
            responses-per-second 5;
            log-only yes;
  };
```

- If using older versions, a patch is available from
  - http://ss.vix.su/~vjs/rrlrpz.html
  - `patch -p0 -l`

Sender Policy Framework (SPF)

- Using DNS for email validation
- Checks the sender IP address
- Defined in RFC 4408 with updates in RFC 6652

```
apnic.net.              3600    IN      TXT     "v=spf1 mx
a:clove.apnic.net a:asmtp.apnic.net ip4:203.119.93.0/24
ip4:203.119.101.0/24 ip4:203.89.255.141/32
ip4:203.190.232.30/32 ip4:122.248.232.184/32
include:_spf.google.com -all"
```

DANE

- DNS-Based Authentication of Named Entities
- RFC 6698 (proposed standard)
- “secure method to associate the certificate that is obtained from the TLS server with a domain name using DNS”
- Adds a TLSA resource record

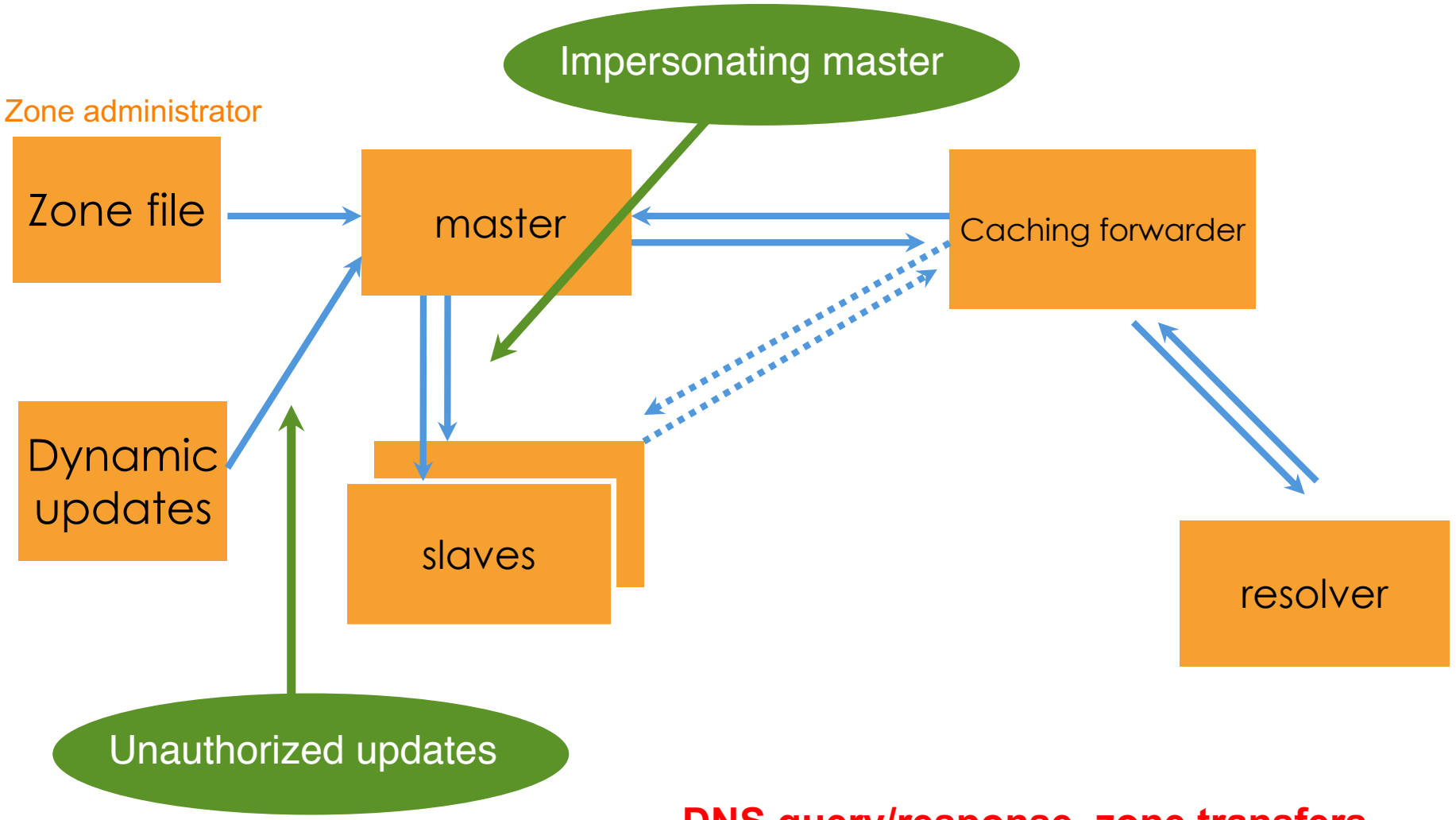
DNS RPZ

- Resource Policy Zone
- Developed for ISC Bind. Built in from version 9.8
- Turns a recursive DNS server into a “DNS firewall”
- “reputation-based” zones
- Like creating a reputation server for recursive DNS servers
  - Function is similar to DNSBL for email SMTP servers
- Blocks DNS resolution to malicious hosts


Questions

DNS Transactions

Transactions - Protected Vulnerabilities



DNS query/response, zone transfers, Dynamic updates

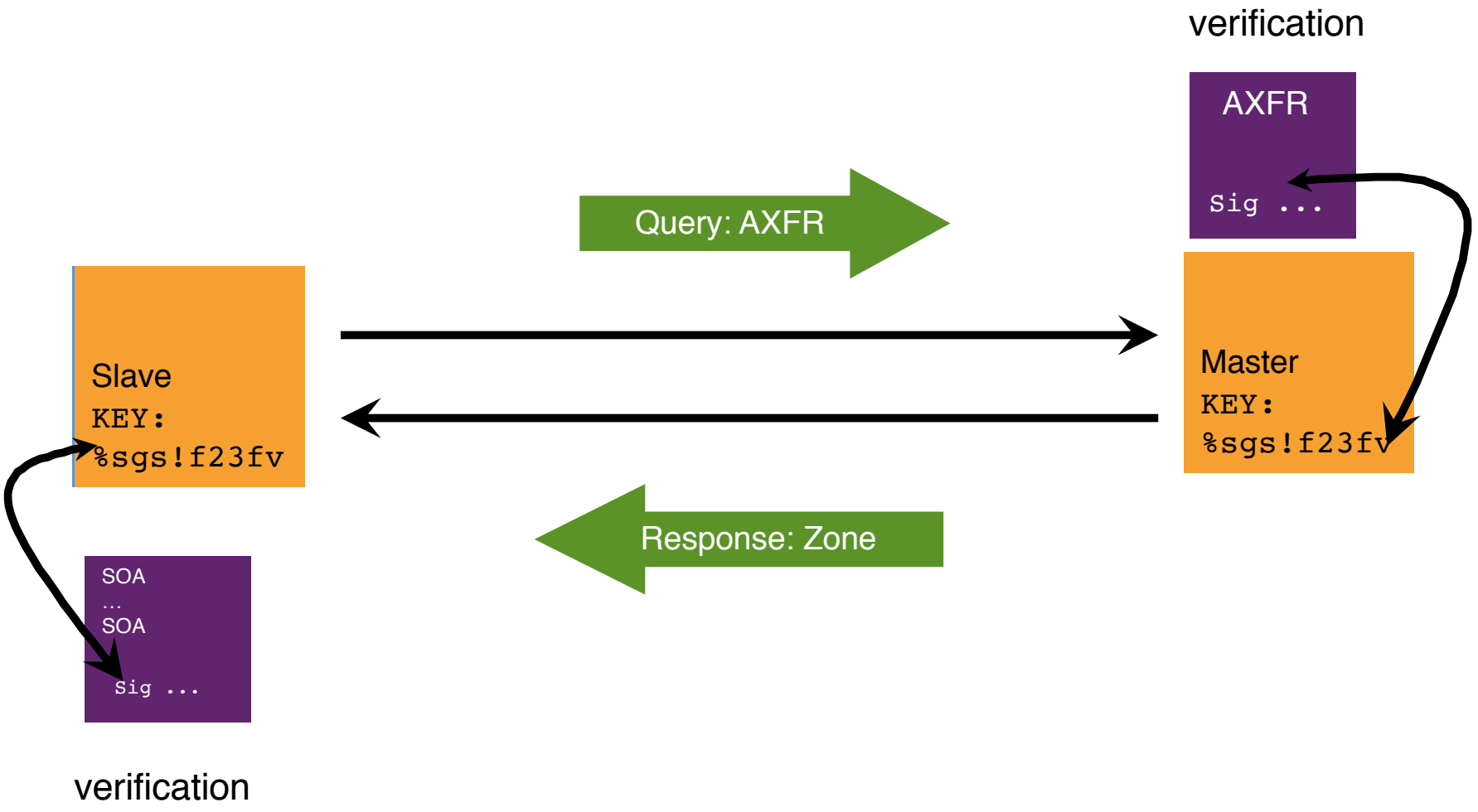
DNS Transactions

- Remote Name Daemon Controller (RNDC)
  - Protects the remote CLI administration using shared key
  - Prevents unauthorized access to named
- Transaction Signature (TSIG)
  - Protects transactions using shared keys between both parties
- SIG(0)
  - Protects transactions using asymmetric key (public and private keypair)

What is Transaction Signature?

- A mechanism for protecting a message from primary to secondary (and vice versa)
- Provides secure communication of queries and responses
  - Also protects zone transfers and dynamic updates
- How?
  - A keyed-hash is applied so recipient can verify the message source
- Based on a shared secret - both sender and receiver are configured with it

TSIG example



TSIG steps

1. Generate secret
2. Communicate secret
3. Configure servers
4. Test

TSIG - Names and Secrets

- TSIG name
  - A name is given to the key, the name is what is transmitted in the message (so receiver knows what key the sender used)

- TSIG secret value
  - A value determined during key generation
  - Usually seen in Base64 encoding

TSIG – Generating a Secret

- dnssec-keygen
  - A simple tool to generate keys
  - Used here to generate TSIG keys

```
dnssec-keygen -a <algorithm> -b <bits> -n host <name
  of the key>
```

TSIG – Generating a Secret

- Example

```
> dnssec-keygen –a HMAC-SHA256 –b 256 –n HOST ns1-
  ns2.pcx.net

This will generate the key
  Kns1-ns2.pcx.net.+157+15921

>ls
Kns1-ns2.pcx.net.+157+15921.key
Kns1-ns2.pcx.net.+157+15921.private
```

TSIG – Generating a Secret

- TSIG is used in server configuration, not in zone file
- Could be confusing because it looks like RR

```
ns1-ns2.pcx.net. IN KEY 128 3 157 nEfRX9…bbPn7lyQtE=
```

TSIG – Configuring Servers

- Configuring the key

```
key { algorithm ...; secret ...;}
```

- Making use of the key

```
server x { key ...; }
```

where x is the IP address of the other server

Configuration Example – named.conf

Primary server 192.168.1.100

```
key ns1-ns2.pcx. net {
        algorithm hmac-md5;
        secret "APlaceToBe";
};
server 192.168.1.200 {
        keys {ns1-ns2.pcx.net;};
};
zone "my.zone.test." {
        type master;
        file “db.myzone”;
        allow-transfer {
        key ns1-ns2.pcx.net ;};
};
```

Secondary server 192.168.1.200

```
key ns1-ns2.pcx.net {
        algorithm hmac-md5;
        secret "APlaceToBe";
};
server 192.168.1.100 {
  keys {ns1-ns2.pcx.net;};
};
zone "my.zone.test." {
        type slave;
        file “myzone.backup”;
        masters {192.168.1.100;};
};
```

You can save this in a file and refer to it in the named.conf using ‘include’ statement:

```
include “/var/named/master/tsig-key-ns1-ns2”;
```

TSIG Testing - dig

- You can use dig to check TSIG configuration
  - `dig @<server> <zone> AXFR -k <TSIG keyfile>`

```
dig @localhost example.net AXFR \
  -k Kns1-ns2.pcx.net.+157+15921.key
```

- A wrong key will give “Transfer failed” and will be logged on the server’s using the security-category

TSIG Testing - Time

- TSIG is time sensitive
- Message protection expires in 5 minutes
  - Make sure time is synchronized
  - For testing, set the time
  - In operations, (secure) NTP is needed

TSIG steps

1. Generate secret

```
dnssec-keygen -a <algorithm> -b <bits> -n host
<name of the key>
```

2. Communicate secret

```
scp <keyfile> <user>@<remote-server>:<path>
```

3. Configure servers

```
key { algorithm ...; secret ...;}
server x { key ...; }
```

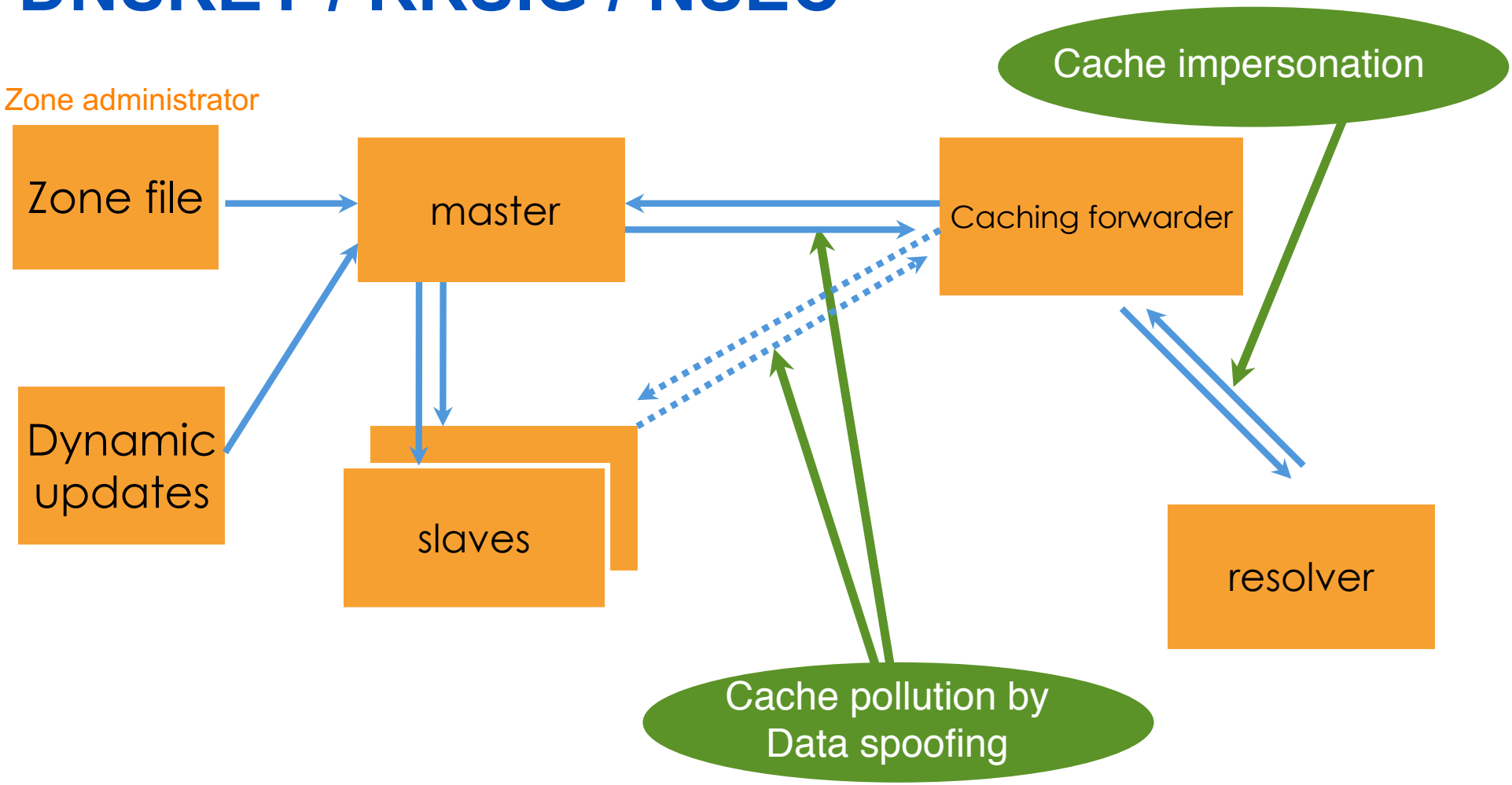
4. Test

```
dig @<server> <zone> AXFR -k <keyfile>
```


Questions

DNSSEC Primer

Vulnerabilities protected by DNSKEY / RRSIG / NSEC



What is DNSSEC?

- **DNS Sec**urity Extensions
- Protects the integrity of data in DNS by establishing a chain of trust
- A form of digitally signing the data to attest its validity
- Uses public key cryptography – each link in the chain has a public/private key pair
- Provides a mechanism to:
  - establish authenticity and integrity of data
  - delegate trust to third parties or parent zones



DNSSEC History

- **1990**: Steven Bellovin discovers a major flaw in the DNS
- **1995**: Bellovin publishes his research; DNSSEC becomes a topic within IETF
- **1998**: Dan Kaminsky discovers some security flaw
- **1999**: RFC 2535, the DNSSEC protocol, is published
- **2005**: Three new RFCs published to update RFC2535
  - RFC 4033 (DNS Security Introduction and Requirements)
  - RFC 4034 (Resource Records for DNS Security Extensions)
  - RFC 4035 (Protocol Modifications)

https://wiki.tools.isoc.org/DNSSEC_History_Project

DNSSEC History

- **2005**: In October, Sweden (.SE) becomes the first ccTLD to deploy DNSSEC
- **2008**: new DNSSEC record created to address privacy concerns (RFC 5155)
- **2010**
  - In July 15, the root zone was signed
  - In July 29, .edu was signed
  - In December 9, .net was signed
- **2011**: In March 31, `.com` was signed

https://wiki.tools.isoc.org/DNSSEC_History_Project

How DNSSEC Works

- Records are signed with private key to prove its authenticity and integrity
- The signatures are published in DNS
- Public key is also published so record signatures can be verified
- Child zones also sign their records with their private key
- Parent signs the hash of child zone's public key to prove authenticity

How DNSSEC Works

Authoritative servers

Sign their zones

Answer queries with the record requested

Also send the digital signature corresponding to the record

Validating Resolvers

Authenticates the responses from the server

Data that is not validated results to a “`SERVFAIL`” error

New Concepts in DNSSEC

- New resource records
- Chain of trust
- Key generation and signing
- Validation

New Resource Records



Resource Record		Function
RRSIG	Resource Record Signature	Signature over RRset made using private key
DNSKEY	DNS Key	Public key needed for verifying a RRSIG
DS	Delegation Signer	Pointer for building chains of authentication
NSEC / NSEC3	Next Secure	indicates which name is the next one in the zone and which type codes are available for the current name

New Resource Records

- **RRsets** are signed with private key to prove its authenticity and integrity
- The signatures are published in DNS as **RRSIG**
- Public **DNSKEY** is also published so RRSIG can be verified
- Child zones also sign their records with their private key
- Parent signs the child zone's **DS record** to prove authenticity

RRs and RRsets

- Resource Record – each entry in the zonefile

```
www.example.net.  7200  IN  A 192.168.1.1
```

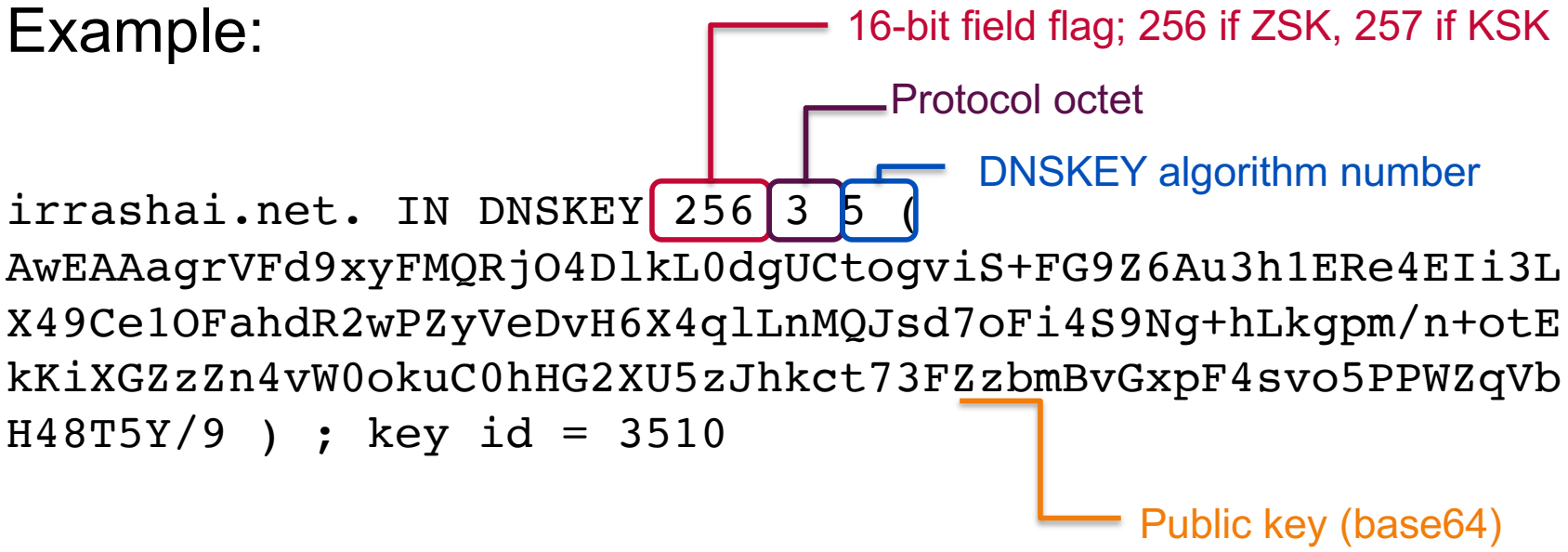
- RRset - RRs with same name, class and type

```
www.example.net. 7200 IN    A     192.168.1.1
web1.example.net. 7200 IN   A     10.0.0.1
web2.example.net. 7200 IN   A     172.16.0.20
```

In DNSSEC, RRsets are signed and not the individual RRs

DNSKEY

- Contains the zone's public key
- Uses public key cryptography to sign and authenticate DNS resource record sets (RRsets).
- Example:



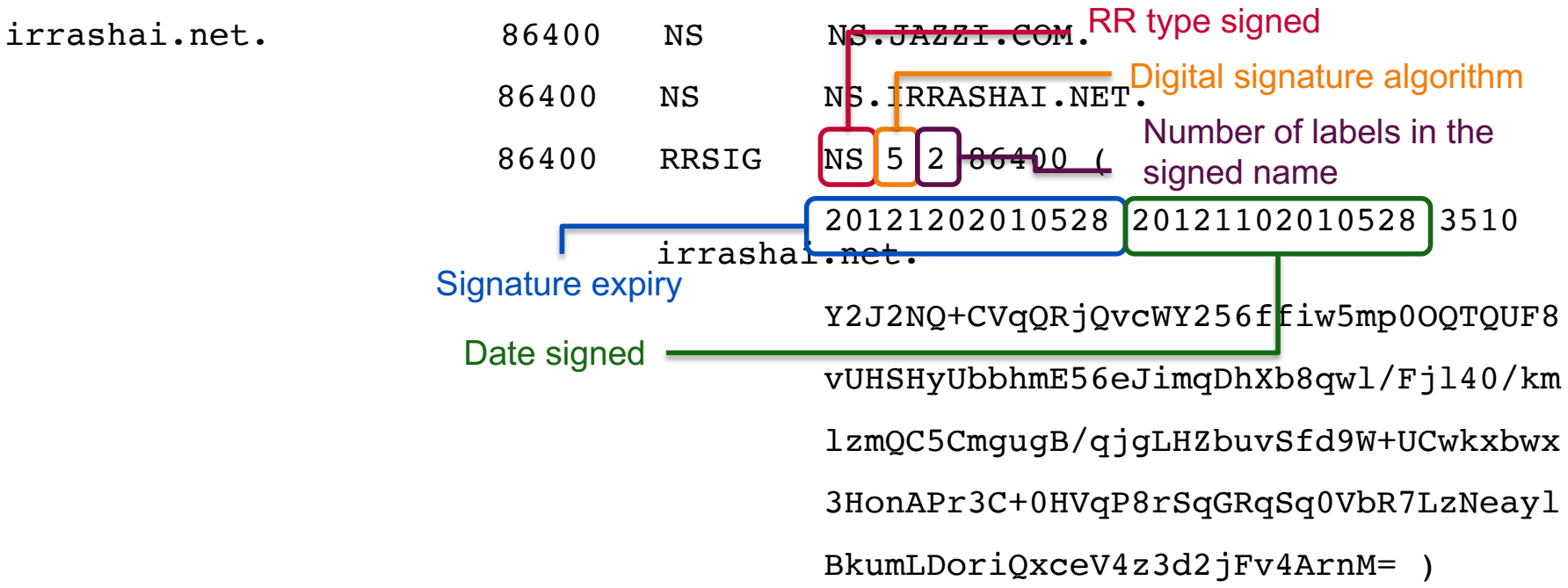
DNSKEY

- Also contains some timing metadata – as a comment in the key file

```
; This is a key-signing key, keyid 19996, for myzone.net.
; Created: 20121102020008 (Fri Nov  2 12:00:08 2012)
; Publish: 20121102020008 (Fri Nov  2 12:00:08 2012)
; Activate: 20121102020008 (Fri Nov  2 12:00:08 2012)
```

RRSIG

- The private part of the key-pair is used to sign the resource record set (Rrset)
- The digital signature per RRset is saved in an RRSIG record



NSEC Record

- **N**ext **Sec**ure
- Forms a chain of authoritative owner names in the zone
- Lists two separate things:
  - Next owner name (canonical ordering)
  - Set of RR types present at the NSEC RR's owner name
- Also proves the non-existence of a domain
- Each NSEC record also has a corresponding RRSIG

```
myzone.net.  NSEC blog.myzone.net. A NS SOA MX RRSIG NSEC DNSKEY
```

NSEC RDATA

- Points to the next domain name in the zone
  - also lists what are all the existing RRs for “name”
  - 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

NSEC Record – Example

```
$ORIGIN example.net.
@ SOA        …
       NS     NS.example.net.
       DNSKEY …
       NSEC    mailbox.example.net. SOA NS NSEC DNSKEY        RRSIG

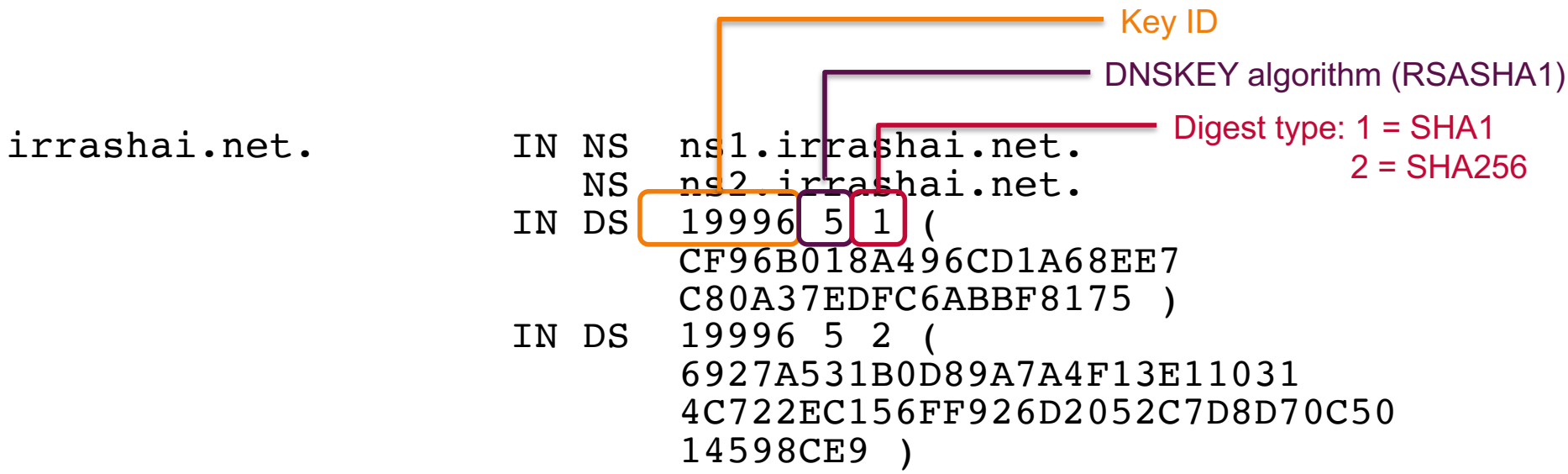
mailbox       A      192.168.10.2
                     NSEC  www.example.net.  A NSEC RRSIG
 WWW          A      192.168.10.3
                     TXT    Public webserver
                     NSEC  example.net. A NSEC RRSIG TXT
```

NSEC3

- NSEC allows an attacker to walk through the linked list to find all the records in the zone file. This is called zone walking.
- NSEC3 uses a hashing algorithm to list the next available domain in “hashed” format
- It is still possible for an attacker to do zone walking, although at a higher computation cost.

DS Record

- **D**elegation **S**igner
- Establishes authentication chains between DNS zones
- Must be added in the parent's zonefile
- In this example, irrashai.net has been delegated from .net. This record is added in the.net zone file



DS Record

- indicates that delegated zone is digitally signed
- Verifies that indicated key is used for the delegated zone
- Parent is authoritative for the DS of the child zone
  - Not for the NS record delegating the child zone
  - DS **should not** be added in the child zone

Chain of Trust

- Establishes a chain of trust from parent to child zone
- How?
  - Parent does not sign child zone
  - Parent only signs a pointer to the child zone (key) – DS RECORD
- The root is on top of the chain

Creation of keys

- In practice, we use two keypairs
  - one to sign the zones, another to sign the other key
- Using a single key or both keys is an operational choice (RFC allows both methods)
- If using a single key-pair:
  - Zones are digitally signed using the private key
  - Public key is published using DNSKEY RR
  - When key is updated, DS record must again be sent to parent zone
- To address this administrative load, two keypairs will be used

Types of Keys

Zone Signing Key (ZSK)

- Sign the RRsets within the zone
- Signed by the KSK
- Uses flag 256

Key Signing Key (KSK)

- Signs the ZSK
- Pointed to by the parent zone
- Acts as the security entry point

Signature Expiration

- Keys do not expire
  - Still a good practice to generate new ones regularly for added security
- Signatures have validity period
  - By default set to 30 days
  - This info is added in the key metadata
- Expired signatures will not validate
  - Must re-sign the zones

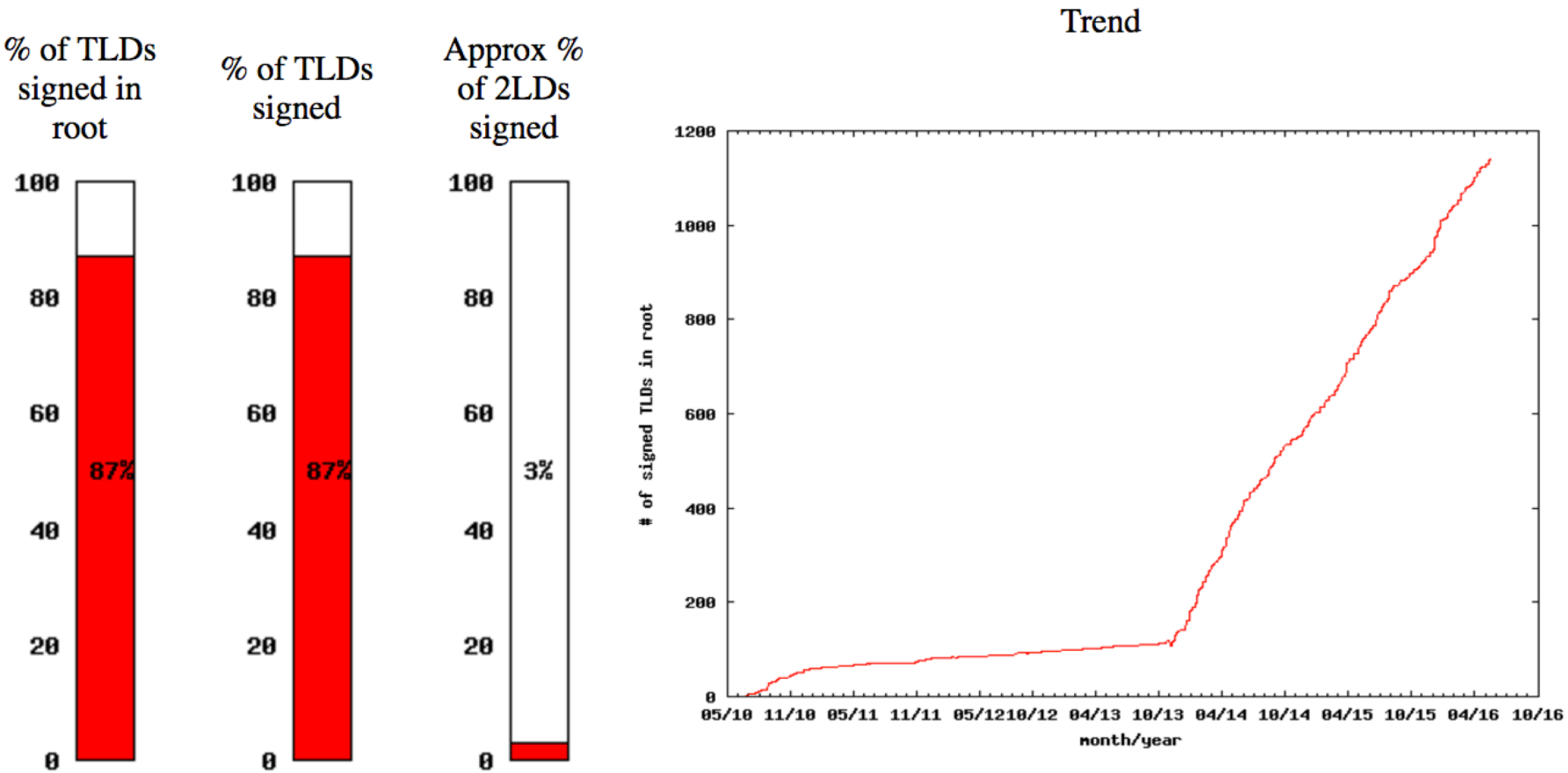
DNSSEC Algorithms

Number	Description	Mnemonic	Zone Signing	Trans. Sec.	Reference
0	Reserved				[RFC4034][RFC4398]
1	RSA/MD5 (deprecated, see 5)	RSAMD5	N	Y	[RFC3110][RFC4034]
2	Diffie-Hellman	DH	N	Y	[RFC2539][proposed standard]
3	DSA/SHA1	DSA	Y	Y	[RFC3755][proposed standard][RFC2536][proposed standard][Federal Information Processing Standards Publication (FIPS PUB) 186, Digital Signature Standard, 18 May 1994.][Federal Information Processing Standards Publication (FIPS PUB) 180-1, Secure Hash Standard, 17 April 1995. (Supersedes FIPS PUB 180 dated 11 May 1993.)]
4	Reserved				[RFC6725]
5	RSA/SHA-1	RSASHA1	Y	Y	[RFC3110][RFC4034]
6	DSA-NSEC3-SHA1	DSA-NSEC3-SHA1	Y	Y	[RFC5155][proposed standard]
7	RSASHA1-NSEC3-SHA1	RSASHA1-NSEC3-SHA1	Y	Y	[RFC5155][proposed standard]
8	RSA/SHA-256	RSASHA256	Y	*	[RFC5702][proposed standard]
9	Reserved				[RFC6725]
10	RSA/SHA-512	RSASHA512	Y	*	[RFC5702][proposed standard]
11	Reserved				[RFC6725]
12	GOST R 34.10-2001	ECC-GOST	Y	*	[RFC5933][standards track]
13	ECDSA Curve P-256 with SHA-256	ECDSAP256SHA256	Y	*	[RFC6605][standards track]
14	ECDSA Curve P-384 with SHA-384	ECDSAP384SHA384	Y	*	[RFC6605][standards track]
15-122	Unassigned				
123-251	Reserved				[RFC4034][RFC6014]
252	Reserved for Indirect Keys	INDIRECT	N	N	[RFC4034][proposed standard]
253	private algorithm	PRIVATEDNS	Y	Y	[RFC4034]
254	private algorithm OID	PRIVATEOID	Y	Y	[RFC4034]
255	Reserved				[RFC4034][proposed standard]

New algorithms such as ECDSA and GOST are faster and can generate smaller keys and signatures

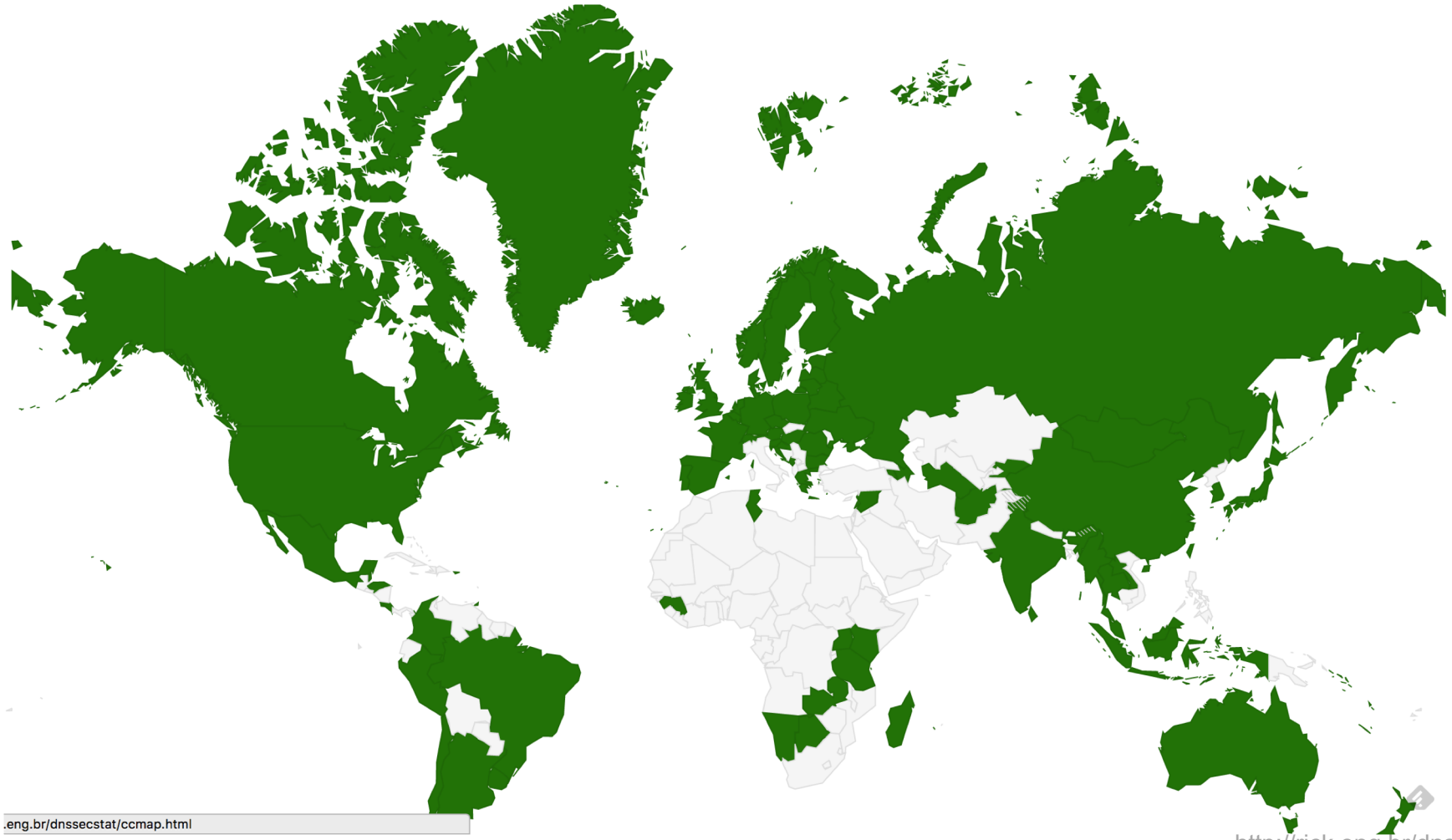
http://www.iana.org/assignments/dns-sec-alg-numbers/dns-sec-alg-numbers.xhtml

DNSSEC Deployment in ccTLDs



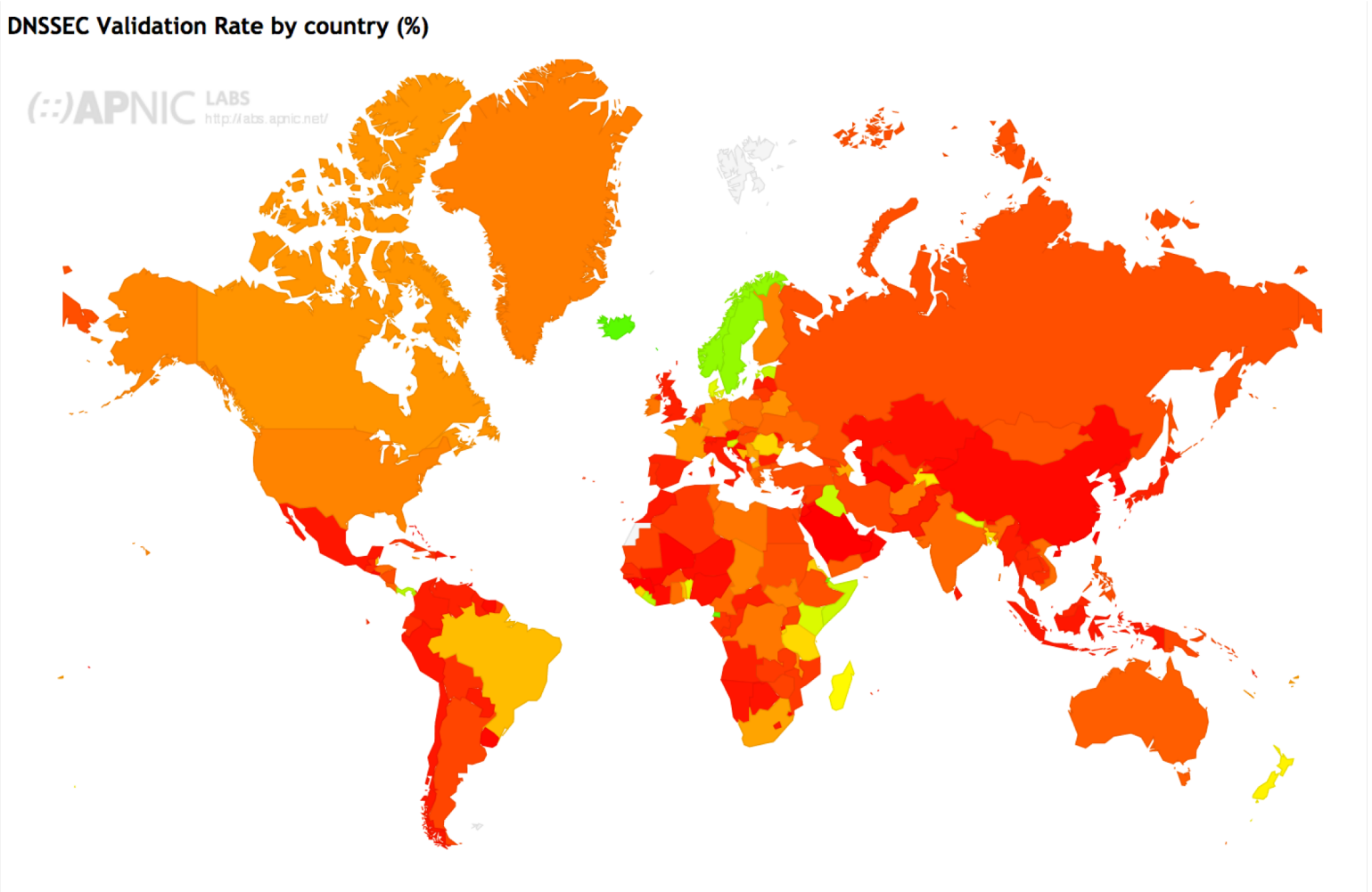
http://rick.eng.br/dnssecstat/

DNSSEC Deployment in ccTLDs



http://rick.eng.br/dnssecstat/

DNSSEC Validation Rate



http://stats.labs.apnic.net/dnssec

DNSSEC for Registries and Hosting Providers

- Sign your zones
- Before fully implementing:
  - Plan about key rollover
  - Think about securing your keys
  - What happens if your key gets compromised
- Support more and newer algorithms (such as ECDSA)

DNSSEC for Network Service Providers

- Enable DNSSEC on your recursive servers and validate responses
  - Deploy DNSSEC-validating resolvers
- Before you fully implement:
  - Domains that can't be validated will be inaccessible
  - Be prepared to answer helpdesk queries related to this


Questions

Implementing DNSSEC

DNSSEC in the Resolver

- Recursive servers that are dnssec-enabled can validate signed zones
- Enable DNSSEC validation

```
dnssec-validation yes;
```

- The AD bit in the message flag shows if validated

DNSSEC Validation

- Other options if you don't have a validating resolver
  - validator add-on for your web browser
    - ex: https://www.dnssec-validator.cz/
  - Online web tools
    - http://dnsviz.net/
    - http://dnssec-debugger.verisignlabs.com/
- Use an open DNSSEC-validating resolver
  - DNS-OARC's ODVR (link)
    - 149.20.64.20 (BIND9), 149.20.64.21 (Unbound)
  - Google Public DNS
    - 8.8.8.8 or 8.8.4.4

DNSSEC - Setting up a Secure Zone

- Enable DNSSEC in the configuration file (named.conf)
  - `dnssec-enable yes; dnssec-validation yes;`
- Create key pairs (KSK and ZSK)
  - `dnssec-keygen -a rsasha1 -b 1024 -n zone myzone.net`
- Publish your public key
- Signing the zone
- Update the config file
  - Modify the zone statement, replace with the signed zone file
- Test with dig

Updating the DNS Configuration

- Enable DNSSEC in the configuration file (named.conf)

```
options {
        directory "…."
        dnssec-enable yes;
        dnssec-validation yes;
};
```

- Other options that can be added later

```
auto-dnssec { off | allow | maintain} ;
```

  - These options are used to automate the signing and key rollover

Generating Key Pairs

- Generate ZSK and KSK

```
dnssec-keygen -a rsasha1 -b 1024 -n zone <myzone>
```

Default values are RSASHA1 for algorithm, 1024 bits for ZSK and 2048 bits for KSK

The command above can be simplified as:

```
dnssec-keygen –f KSK <myzone>
```

This generates four files.

Note: There has to be at least one public/private key pair for each DNSSEC zone

Generating Key Pairs

- To create ZSK

```
dnssec-keygen -a rsasha1 -b 1024 -n zone
myzone.net
```

- To create KSK

```
dnssec-keygen -a rsasha1 -b 2048 -f KSK -n
zone myzone.net
```

Generating Key Pairs - Reverse

- To create ZSK

```
dnssec-keygen -a rsasha1 -b 1024 -n zone
100.168.192.in-addr.arpa
```

- To create KSK

```
dnssec-keygen -a rsasha1 -b 2048 -f KSK -n
zone 100.168.192.in-addr.arpa
```

Publishing the Public Key

- Using `$INCLUDE` you can call the public key (DNSKEY RR) inside the zone file

```
$INCLUDE /path/Kmyzone.net.+005+33633.key ; ZSK
$INCLUDE /path/Kmyzone.net.+005+00478.key ; KSK
```

- You can also manually enter the DNSKEY RR in the zone file

Signing the Zone

- Sign the zone using the secret keys:

```
dnssec-signzone –o <zonename> -N INCREMENT -f
<output-file> -k <KSKfile> <zonefile> <ZSKfile>
```

```
dnssec-signzone –o myzone.net db.myzone.net
Kmyzone.net.+005+33633
```

- Once you sign the zone a file with a .signed extension will be created
  - `db.myzone.net.signed`

Signing the Zone

- Note that only authoritative records are signed
  - NS records for the zone itself are signed
  - NS records used for delegations are not signed
  - DS records are signed
  - Glue records are not signed
- Notice the difference in file size
  - db.myzone.net vs. db.myzone.net.signed

Smart Signing

- Searches the key repository for any keys that will match the zone being signed

```
options {
     keys-directory { “path/to/keys”;
};
```

- Then the command for smart signing is

```
dnssec-signzone –S db.myzone.net
```

Publishing the Zone

- Reconfigure to load the signed zone. Edit named.conf and point to the signed zone.

```
zone “<myzone>” {
     type master;
     # file “db.myzone.net”;
     file “db.myzone.net.signed”;
};
```

Publishing the Zone – Reverse

- Reconfigure to load the signed zone. Edit named.conf and point to the signed zone.

```
zone “<myzone>” {
     type master;
     # file “db.192.168.100”;
     file “db.192.168.100.signed”;
};
```

Testing the Server

- Ask a dnssec-enabled server and see whether the answer is signed

```
dig @localhost www.apnic.net +dnssec
+multiline
```

Testing with Dig

```
dig @localhost www.irrashai.net +dnssec (+multiline)
```

```
; <<>> DiG 9.9.5-P1 <<>> @localhost www.irrashai.net +dnssec
; (2 servers found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 10871
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 2, ADDITIONAL: 3

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;www.irrashai.net.              IN      A

;; ANSWER SECTION:
www.irrashai.net.       864000  IN      A       192.168.100.100
www.irrashai.net.       864000  IN      RRSIG   A 5 3 864000 20150604031347 20150505031347 44727 irrashai.net. HBFfuooWXCIOyOuySO11/rSru5smi/E2mXAHR2tEP093IT8gMIPSQIL4 78XN3ecg3xQ1o
oeYTFjX6dgknE6Y4o179Ufba+zreHRP6sbBf852Btf4 wSExAZdOS9BmTEtDlhKXRDMnc0/9enqcfnku7IQqDYxudGBGfNmF5mnr gGY=

;; AUTHORITY SECTION:
irrashai.net.           864000  IN      NS      NS.IRRASHAI.NET.
irrashai.net.           864000  IN      RRSIG   NS 5 2 864000 20150604031347 20150505031347 44727 irrashai.net. 0BdYHJMLtvhhfbdwtcA4Z0Ja83L6iB51msJpurYzzffmiB5amq1V30YR vaFHqYM64Lmi
iXAePvq/mpdvutx6FiggNTyVbOHQ7+1ecHdNXO+AkGuF 2h4Go/rpjBpBN9a4Fexvuw7laO8CSykpfTNZ4hNaFagO/WmzbE9Pzm1K Vmg=

;; ADDITIONAL SECTION:
ns.irrashai.net.        864000  IN      A       192.168.100.8
ns.irrashai.net.        864000  IN      RRSIG   A 5 3 864000 20150604031347 20150505031347 44727 irrashai.net. MQQsnqWjMDJXI1VHNzXWywbRqDhYrEqxd3tMtx2Ua8ep+HYMfsJ/8/Im F9IfdPKm3TN+6
okecCionMixtzuvNLAs9FXY5q5V0lpSuC+oRe6Fulip i75uvARtYoLttB3zBHVzAIIlULzsDyrgagZZNrSS+EF12oeKNwOSYEir 64k=

;; Query time: 0 msec
;; SERVER: ::1#53(::1)
;; WHEN: Wed May 06 17:10:44 EST 2015
;; MSG SIZE  rcvd: 625
```

Testing with Dig – Reverse

```
dig @localhost -x 192.168.100.100 +dnssec
```

```
; <<>> DiG 9.9.5-P1 <<>> @localhost -x 192.168.100.100 +dnssec
; (2 servers found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 10393
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 2, ADDITIONAL: 3

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;100.100.168.192.in-addr.arpa.  IN      PTR

;; ANSWER SECTION:
100.100.168.192.in-addr.arpa. 864000 IN PTR     www.irrashai.net.
100.100.168.192.in-addr.arpa. 864000 IN RRSIG   PTR 5 6 864000 20150604031101 20150505031101 22107 100.168.192.in-addr.arpa. FyBAUv5Z8Z+8H8ZpbxZjAaFIpC9cJfzwY8Ojuol92wetwdzF0dyUV9v/
 XSwizzqGO9Pe3nchwRJNt7Of27xB52HgYOryyO/UudxF5TxN8Dp10rmj AbbR/9GrWIW9TOunBWfv17Pnxb1AMvTckncdogZeSghRV5QZ6rvmMtx2 yxk=

;; AUTHORITY SECTION:
100.168.192.in-addr.arpa. 864000 IN     NS      NS.IRRASHAI.NET.
100.168.192.in-addr.arpa. 864000 IN     RRSIG   NS 5 5 864000 20150604031101 20150505031101 22107 100.168.192.in-addr.arpa. mXv26lJVvtAZxM7Ni/DZwr7Vw/xZ5da8iflNRTm0zWe3huKiBkCoXnB0
TXmTNQKxfknfA1pLPrC4OZL4UyP00vAOwi5VYFZzwF/KA9xI9o8f59ng KbxWsbGtHLl3/e4Q8+lKSfVb4Al0cAF/m3yauQjYHGXzCHBO76w9nhk+ E7A=

;; ADDITIONAL SECTION:
ns.irrashai.net.        864000  IN      A       192.168.100.8
ns.irrashai.net.        864000  IN      RRSIG   A 5 3 864000 20150604031347 20150505031347 44727 irrashai.net. MQQsnqWjMDJXI1VHNzXWywbRqDhYrEqxd3tMtx2Ua8ep+HYMfsJ/8/Im F9IfdPKm3TN+6
okecCionMixtzuvNLAs9FXY5q5V0lpSuC+oRe6Fulip i75uvARtYoLttB3zBHVzAIIlULzsDyrgagZZNrSS+EF12oeKNwOSYEir 64k=

;; Query time: 0 msec
;; SERVER: ::1#53(::1)
;; WHEN: Wed May 06 17:08:58 EST 2015
;; MSG SIZE  rcvd: 675

[root@testserver master]# _
```

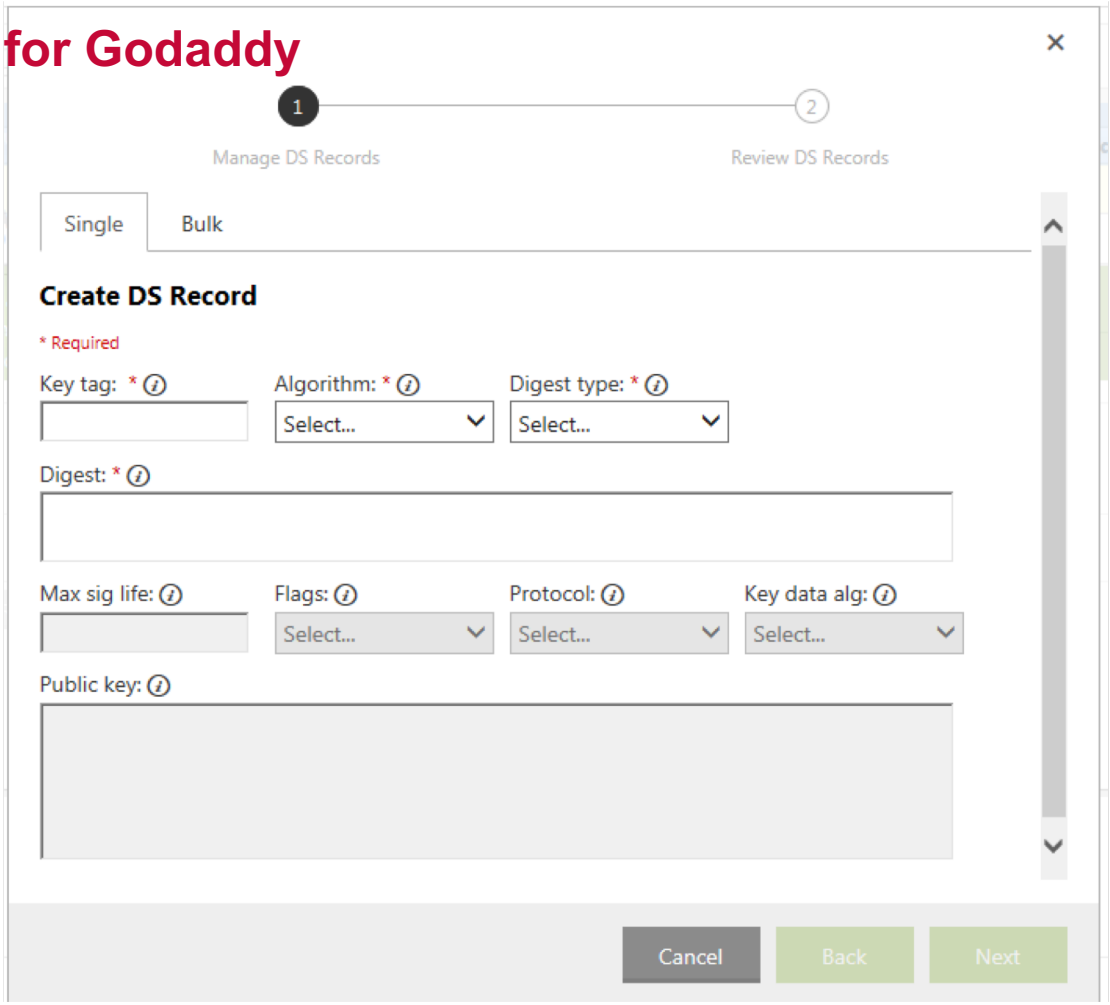
Pushing the DS record

- The DS record must be published by the parent zone.
- Contact the parent zone to communicate the KSK to them.

- There are proposals in the IETF DNSOP WG to address this:
  - Automating DNSSEC Delegation Trust Maintenance (link)
  - Child to Parent Synchronization in DNS (link)

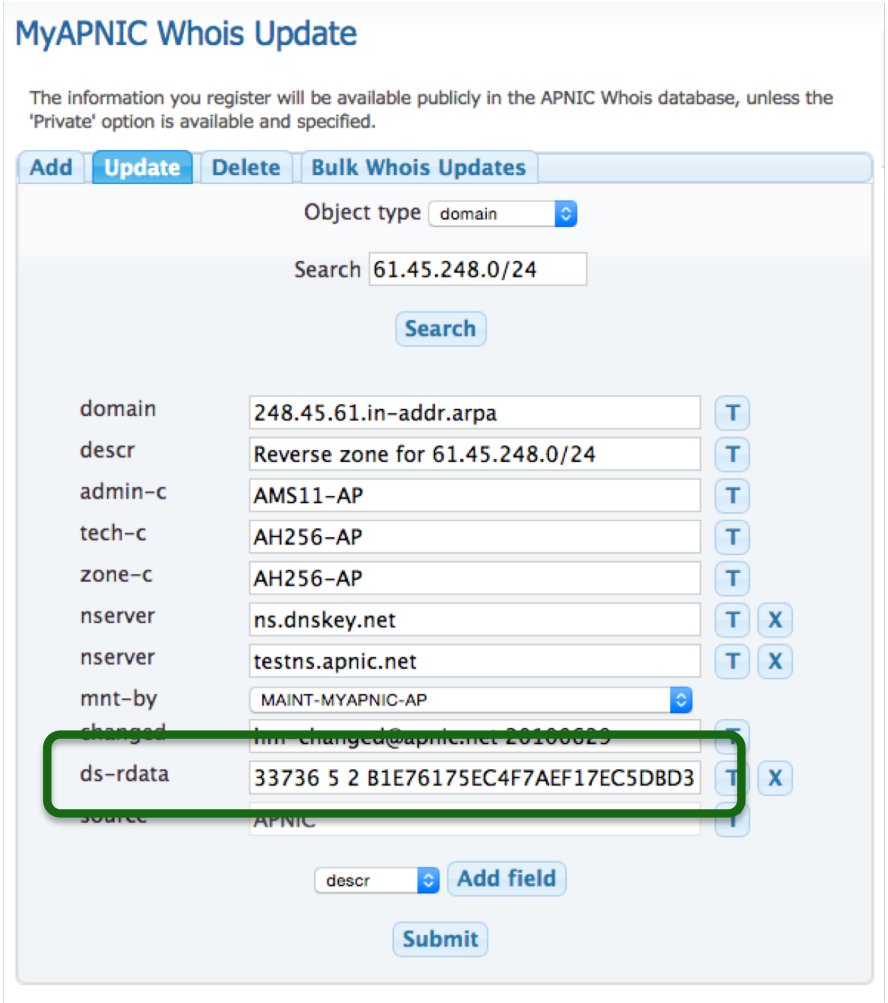
Pushing DS Records for Forward Zone

Example form for Godaddy



Pushing DS Record for Reverse Zone

Using MyAPNIC



DS record added in the domain object

Ways to Deploy DNSSEC

- As part of the DNS software used
  - Manual key management
  - Can be quite complex
  - For static environment
  - Some means of automation using
    - option commands and scripts

DNSSEC tools for BIND, NSD, PowerDNS, etc

- Use with a hardware security module (HSM)
  - Semi-automatic
  - Good for dynamic environment

HSM, OpenDNSSEC

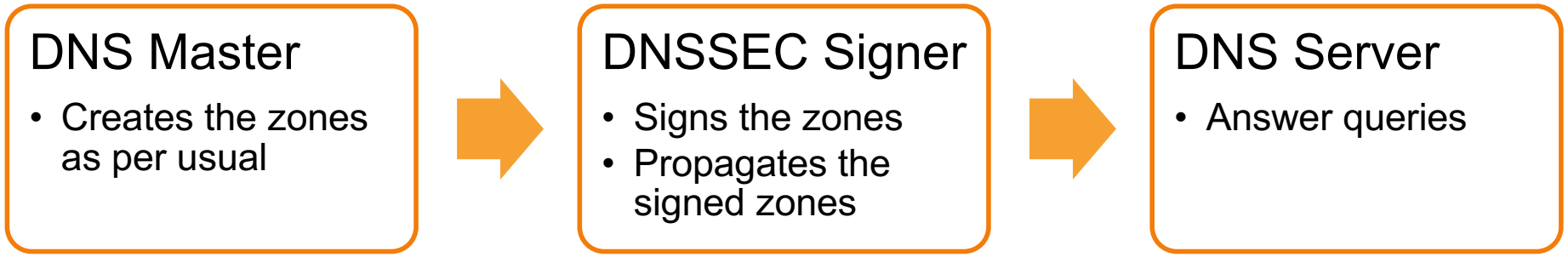
- Using an external appliance
  - 'dnssec-in-a-box'
  - Fully automates key generation, signing and rollover

DNS Appliance

Hardware Security Module

- Cryptographic devices used for storage of the encryption keys
  - Smart cards, PCI cards, USB tokens
- It also speeds up the cryptographic key generation
- Implements PKCS#11 (Cryptographic Token Key Interface)
  - A standard interface or API to cryptographic tokens

DNSSEC Signer Appliance



- Can be a pure signer or packaged with an IPAM or a DNS server
- In pure signer, the hardware appliance interfaces between the master/slave servers
- Examples: Secure64, Xelerance, SolidDNS, etc


Questions

DNSSEC Key Management

KSK Key Rollover

Using Double signing

- When you change the KSK keys, the DS record in the parent zone must also be updated

```
dnssec-signzone –o myzone.net –N increment –f <output- \
file> -k Kmyzone.net.+005+11111 db.myzone.net \
Kmyzone.net.+005+67890
```

- Send the new DS record to the parent, and wait for it to propagate
- Remove the old key and re-sign

KSK Key Rollover

Using Pre-publication

- In this method, the new key will be published but will not be used for signing yet.

```
dnssec-keygen –K keydir –f ksk –A none <myzone.net>
rndc loadkeys <myzone.net>
```

- Publish both keys, but use only the old one for signing
- Wait for the propagation time and TTL of the DNSKEY RR to expire.

KSK Key Rollover

- Then use dnssec-settime once you are ready to sign the zone. Use the new key for zone signing, leaving the old one published.

```
dnssec-settime –K keydir –A now Kmyzone.net.+005+12345
rndc loadkeys myzone.net
```

- Wait for the propagation and TTL in the old zone. Set the old key to no longer sign with the key, but leaves it in the zone.

```
dnssec-settime –K keydir -I now Kmyzone.net.+005+12345
rndc loadkeys myzone.net
```

- Now remove the old keys. This completely removes the keys.

```
dnssec-settime –K keydir -D now Kmyzone.net.+005+12345
rndc loadkeys myzone.net
```

Other Options – Automated Signing

- Using RNDC

- Add the option to named.conf

```
auto-dnssec allow;
```

- Then you can use the commands:

```
rndc loadkeys zone
rndc sign zone
```

DNSSEC Operational Practices RFC 6781

- Lists down choices and decisions available when deploying DNSSEC
- Keep the chain of trust
  - Broken chains result in data being marked as Bogus
  - Shared responsibility by admins
- Key generation and storage
  - The motivations to differentiate KSK and ZSK are purely operational
  - Timing parameters
  - Key compromise and risk of cryptanalysis
  - Keys should be large enough to avoid all known crypto attacks during the effectivity period of the key
  - zone private keys and the zone file master copy to be signed be kept and used in off-line

https://tools.ietf.org/html/rfc6781

DNSSEC Operational Practices RFC 6781

- Signature generation, key rollover and policies
  - Data published in previous versions still live in caches
  - ZSK can be rolled without taking into account the DS record from parent
  - KSK rollover requires interaction with the parent
  - Emergency key rollover
- Motivation to deploy NSEC3 over NSEC
  - Prevention of zone enumeration

https://tools.ietf.org/html/rfc6781

DNSSEC Practice Statement – RFC 6841

- a means for stakeholders to evaluate the strength and security of the DNSSEC chain of trust
- DNSSEC Policies (DPs) – security requirements and standards to be implemented for a DNSSEC-signed zone
- DNSSEC Practice Statement (DPS) – practice disclosure document; states how the management of a given zone implements procedures and controls at a high level

https://tools.ietf.org/html/rfc6841

DNSSEC Practice Statement

- The DPS for Root Zone Signing Key (ZSK) is published
  - https://www.iana.org/dnssec/icann-dps.txt
- Published DPS of TLD operators

.SE's DNSSEC Practice Statement

www.iis.se/docs/se-dnssec-dps-eng.pdf

.CL's DNSSEC Practice Statement

http://www.nic.cl/dnssec/en/dps.html

.NET DNSSEC Practice Statement

http://www.verisigninc.com/assets/20100925-NET+DPS-FINAL.pdf

DNSSEC Guides

- Good Practice Guide for Deploying DNSSEC
  - ENISA
  - Published 2010
- Secure Domain Name System Deployment Guide
  - NIST
  - Published 2013


Questions

Thank You!

END OF SESSION





www.facebook.com/APNIC



www.twitter.com/apnic



www.youtube.com/apnicmultimedia



www.flickr.com/apnic



www.weibo.com/APNICrir

APNIC