DNS Security

Sheryl Hermoso
sheryl@apnic.net

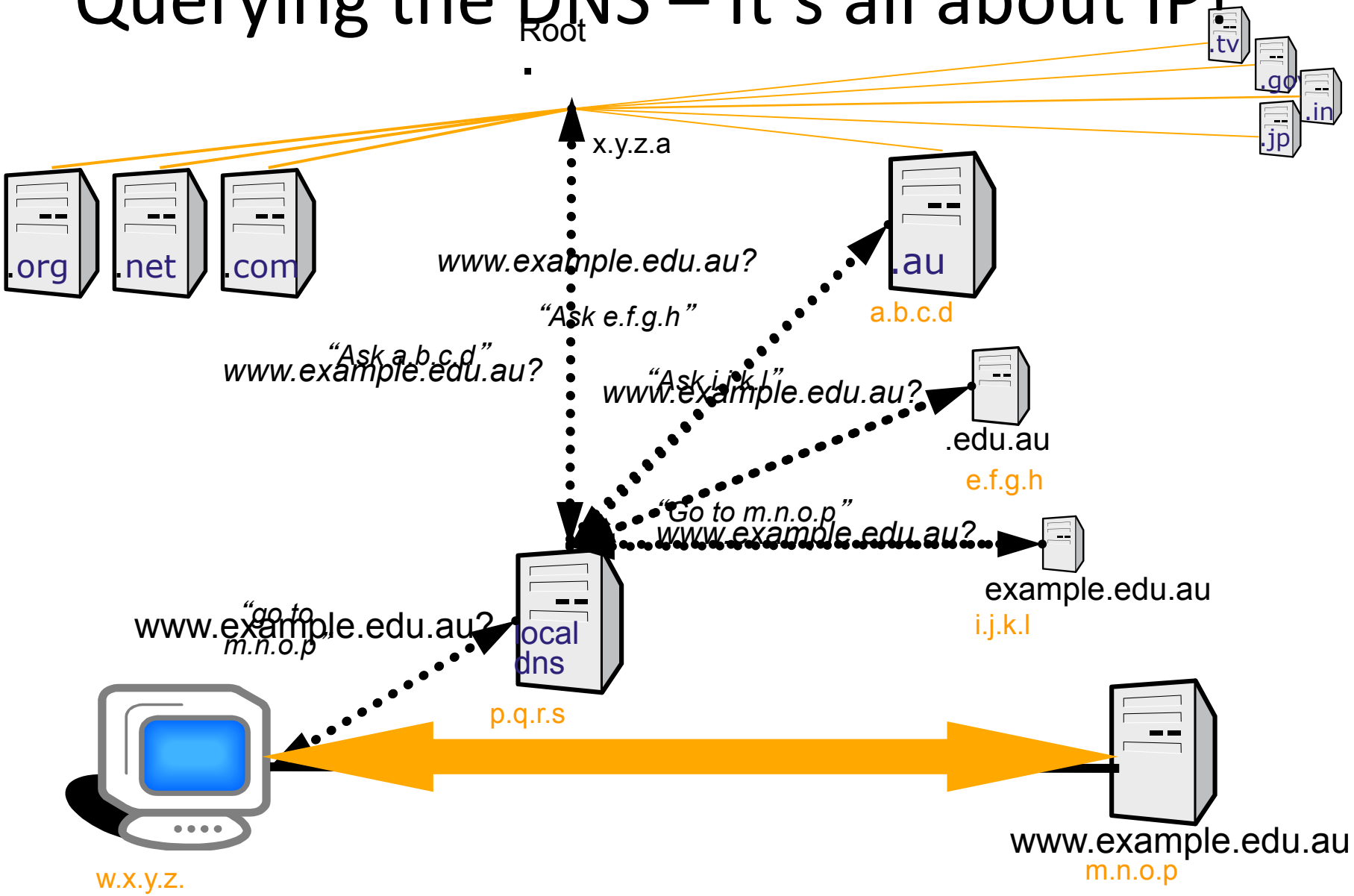
Background

- The original DNS protocol wasn't designed with security in mind
- It has very few built-in security mechanism
- As the Internet grew wilder & wollier, IETF realized this would be a problem
  - For example DNS spoofing was to easy
- DNSSEC and TSIG were develop to help address this problem

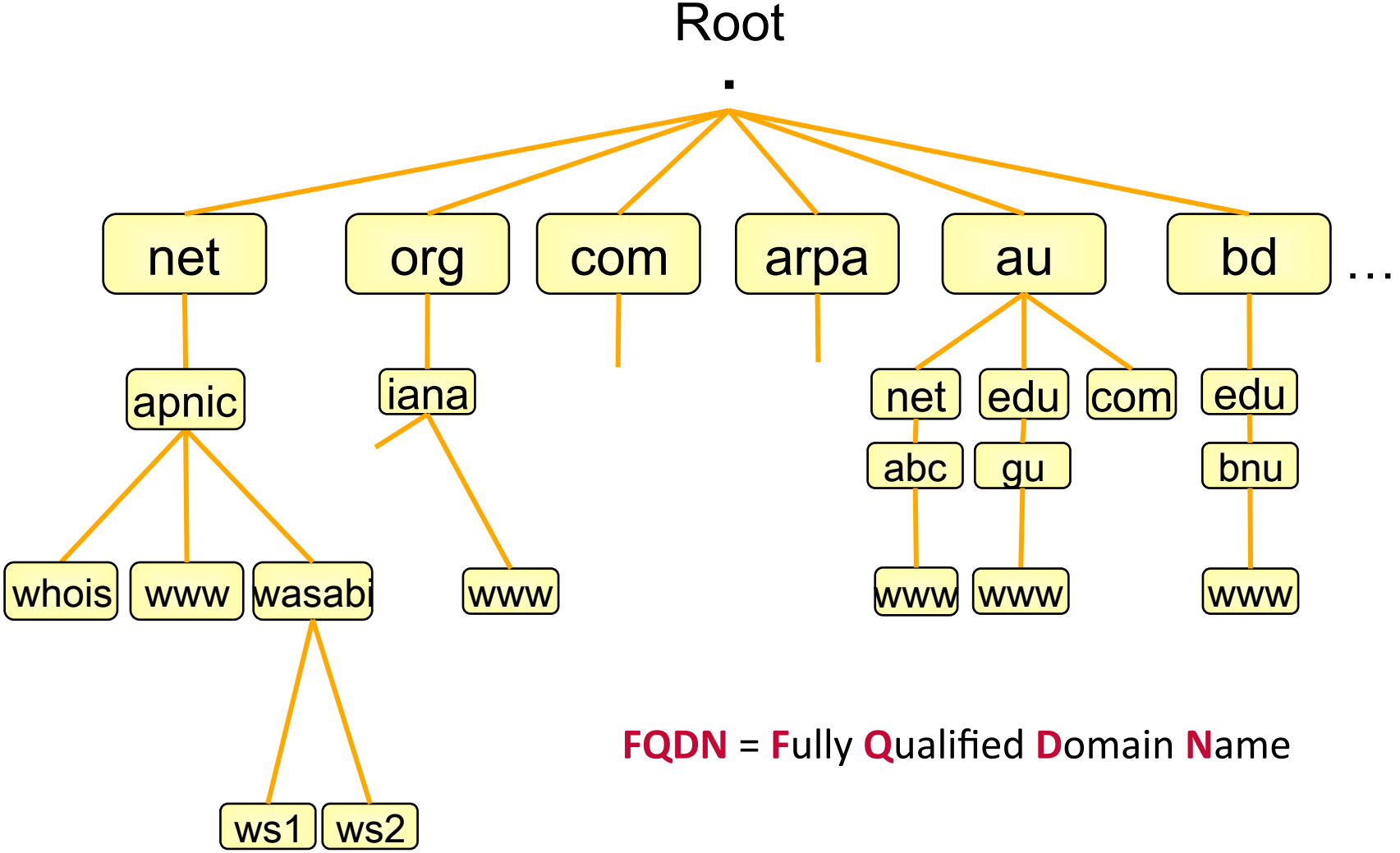
A little review …

We can skip if everyone’s familiar.

Querying the DNS – It's all about IP!



The DNS Tree Hierarchy



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

Delegating a Zone

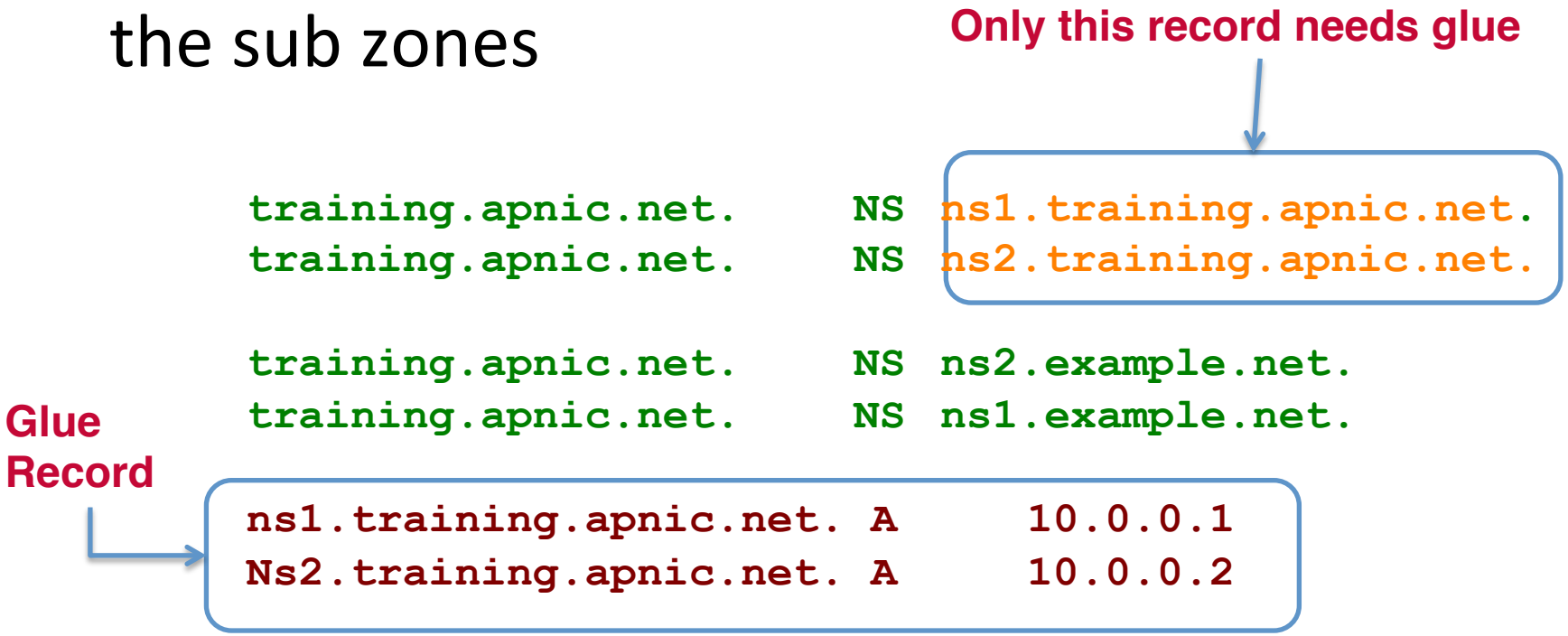
- Delegation is passing of authority for a subdomain to another party
- Delegation is done by adding NS records
  - Ex: if APNIC.NET wants to delegate TRAINING.APNIC.NET

```
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



Lame Delegations

- When a nameserver has been listed as authoritative for a domain, but does not seem to be performing authoritative service for that domain.
  - nameserver appears to be answering out of its cache instead of out of its data
  - even a server which is performing secondary service for a domain is still an authoritative server, and should be returning authoritative data
- Fixing lame delegations
  - Syntax errors in the nameserver boot file or zone file are the most common cause
  - http://www.cymru.com/DNS/lame.html

NXDOMAIN

- Companies are using 3rd Party providers to redirect NXDOMAIN responses to specified URLs
  - Large motivator for companies to do this is that it's a significant revenue generator
- How effective have companies been at stopping this practice?
- Are there issue with DNSSEC deployments?
  - Since validation is done at the resolver, DNSSEC has no impact on redirection done post-validation

DNS Security

DNS Security

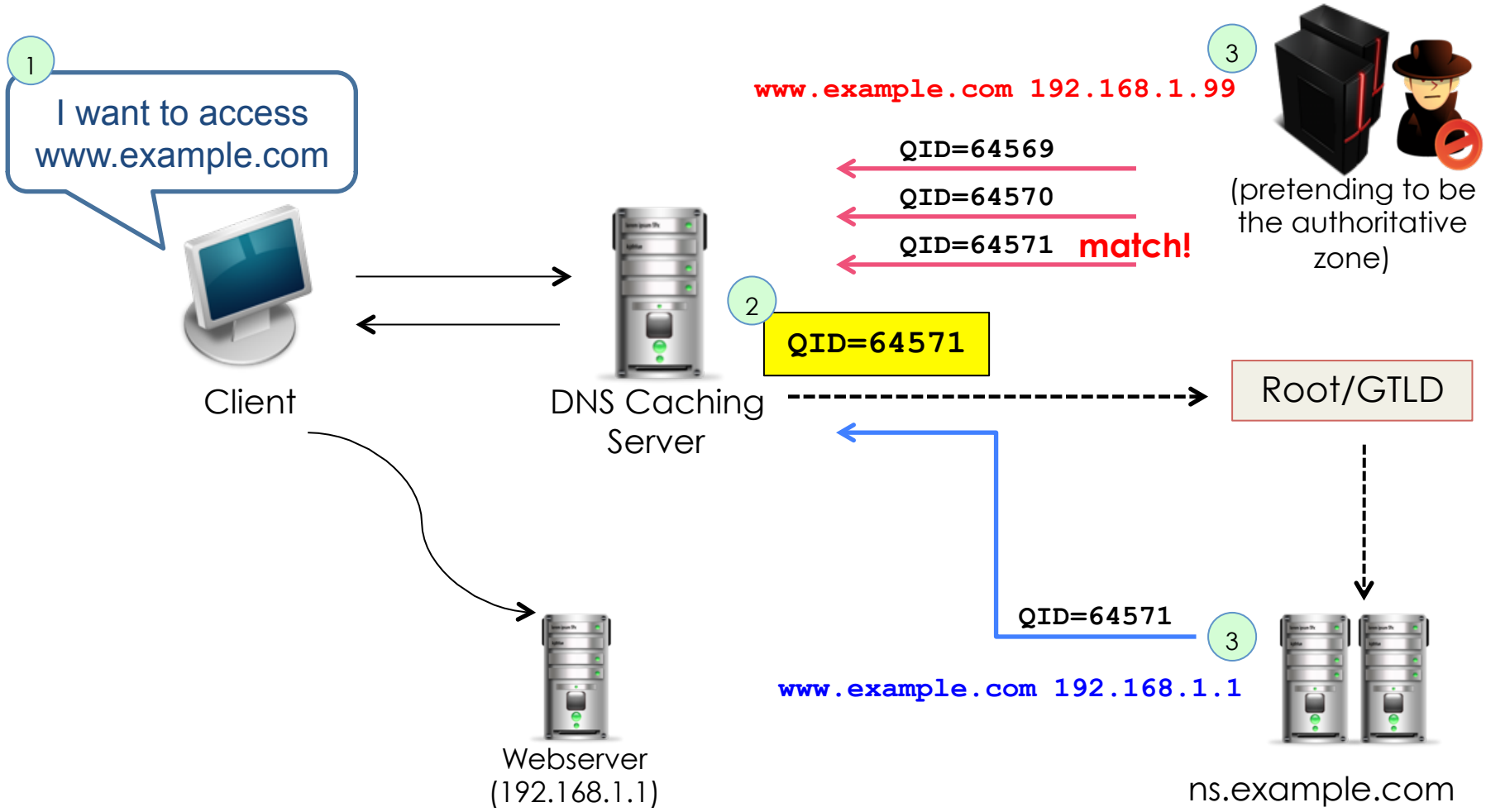
- DNS by itself has no built-in security measures
- Vulnerable to malicious attacks
- Some security problems:
  - Using reverse DNS to impersonate hosts
  - Software bugs (buffer overflows, bad pointer handling)
  - Bad crypto (predictable sequences, forgeable signatures)
  - Cache poisoning (putting inappropriate data into the cache)

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

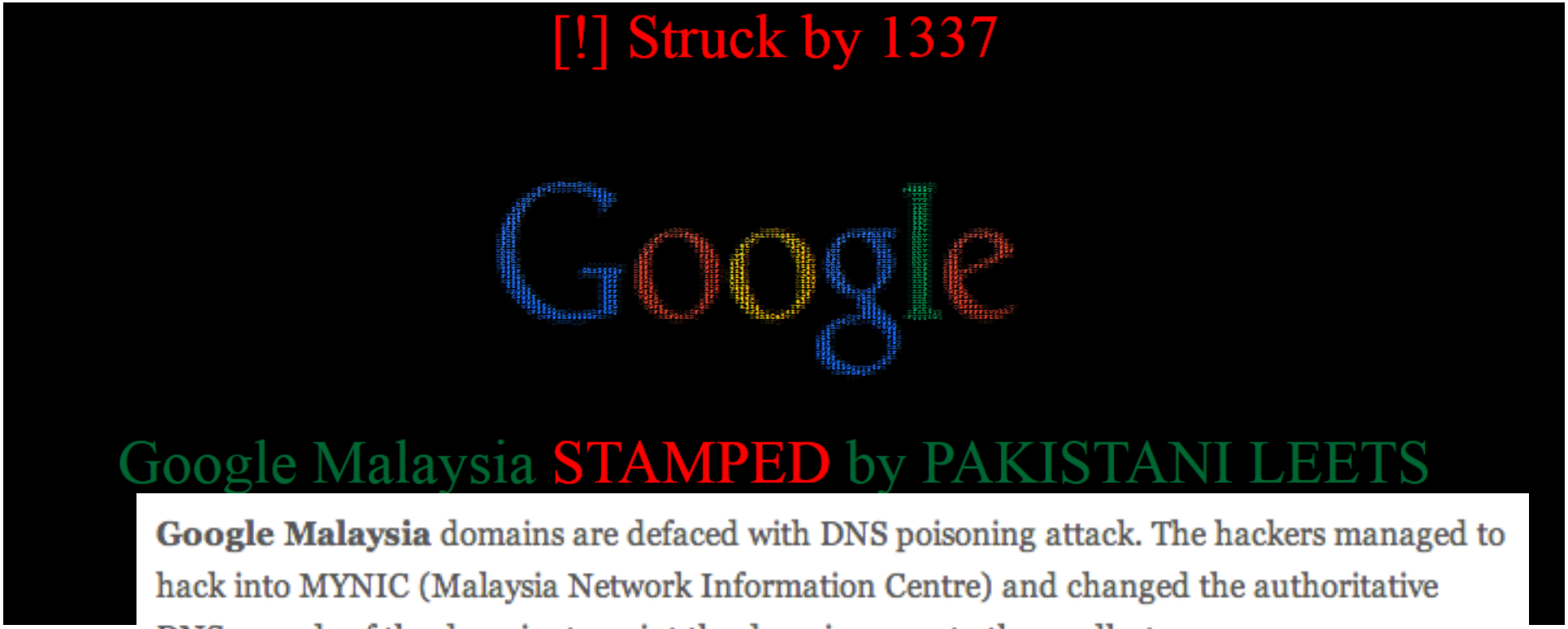
DNS Security

- You have control over your own DNS infrastructure
- What about the aspects you do not control?
  - How many domains are you responsible for?
    - Typical to have ownership at legal level (Hmmmmm….)
  - Do you know when someone else is using your domain?
  - Do you know when someone is redirecting DNS traffic from your site?
- Have you even thought about WHO you register your domain with and ensuring no one can easily change records on who owns the domain?

DNS Cache Poisoning



DNS Poisoning at google.my

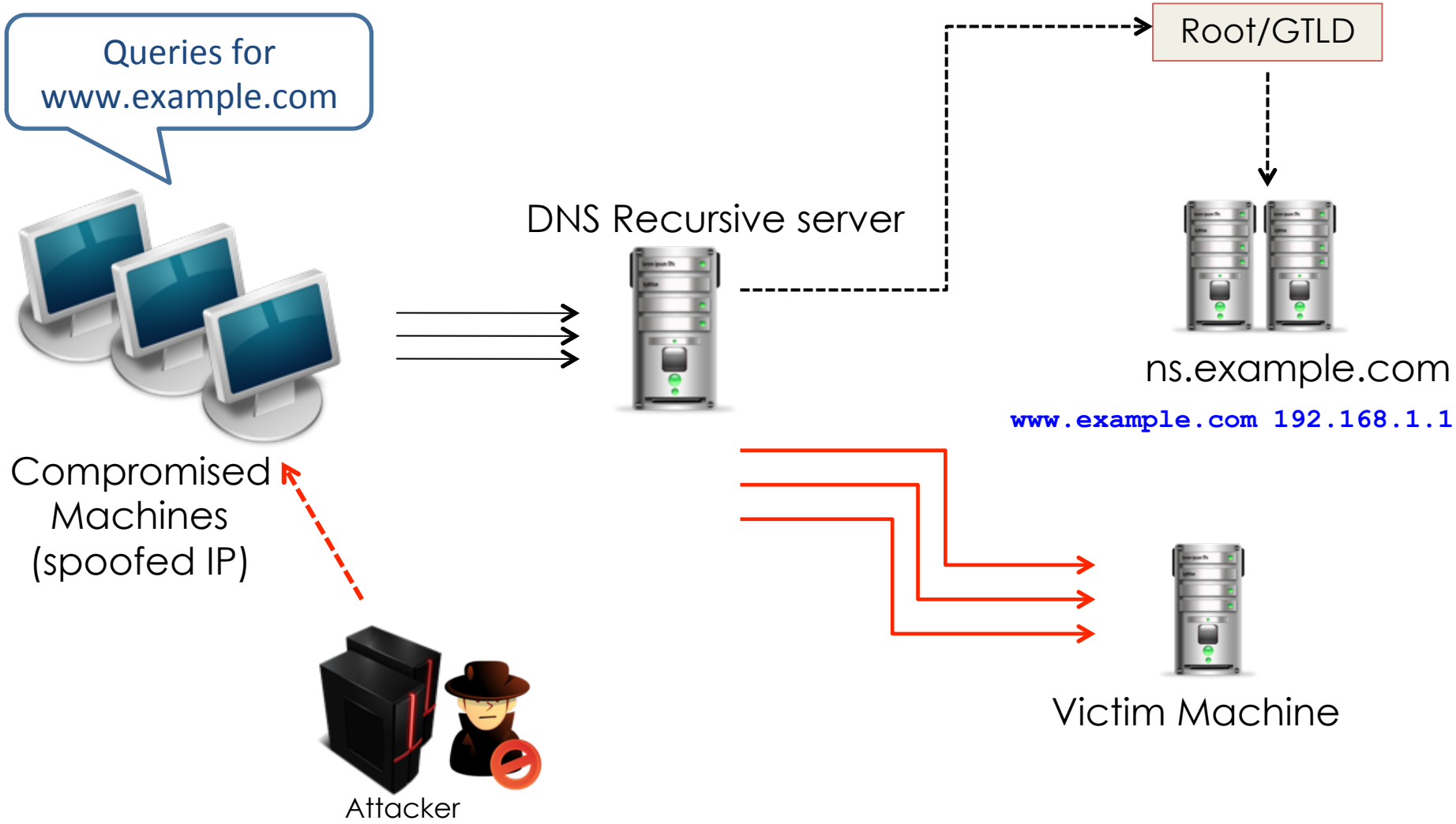


http://thehackerspost.com/2013/10/google-malaysia-gets-hacked-1337-hacker-madleets.html

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

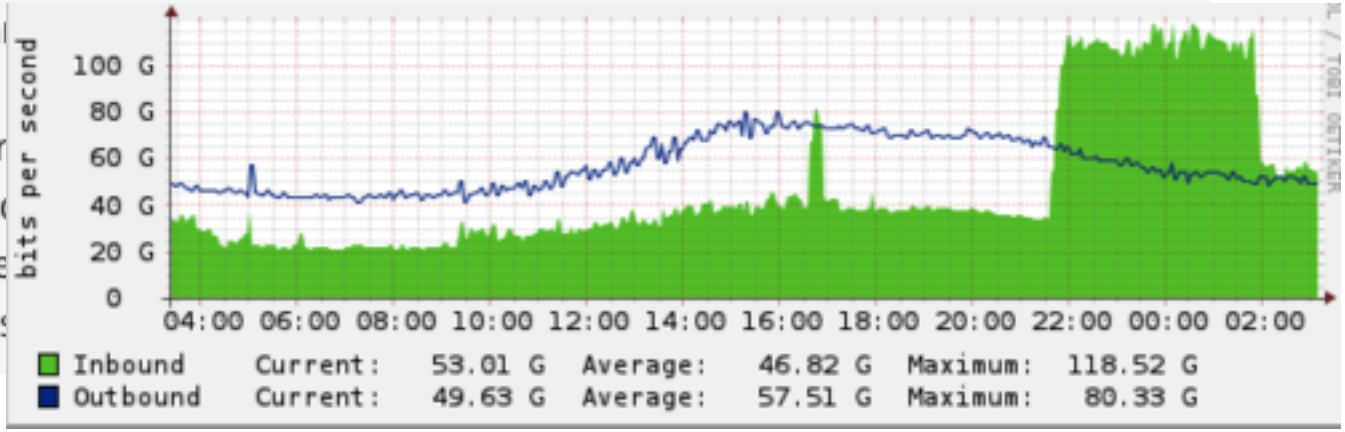
DNS Amplification



Amplification 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

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;
        };
```

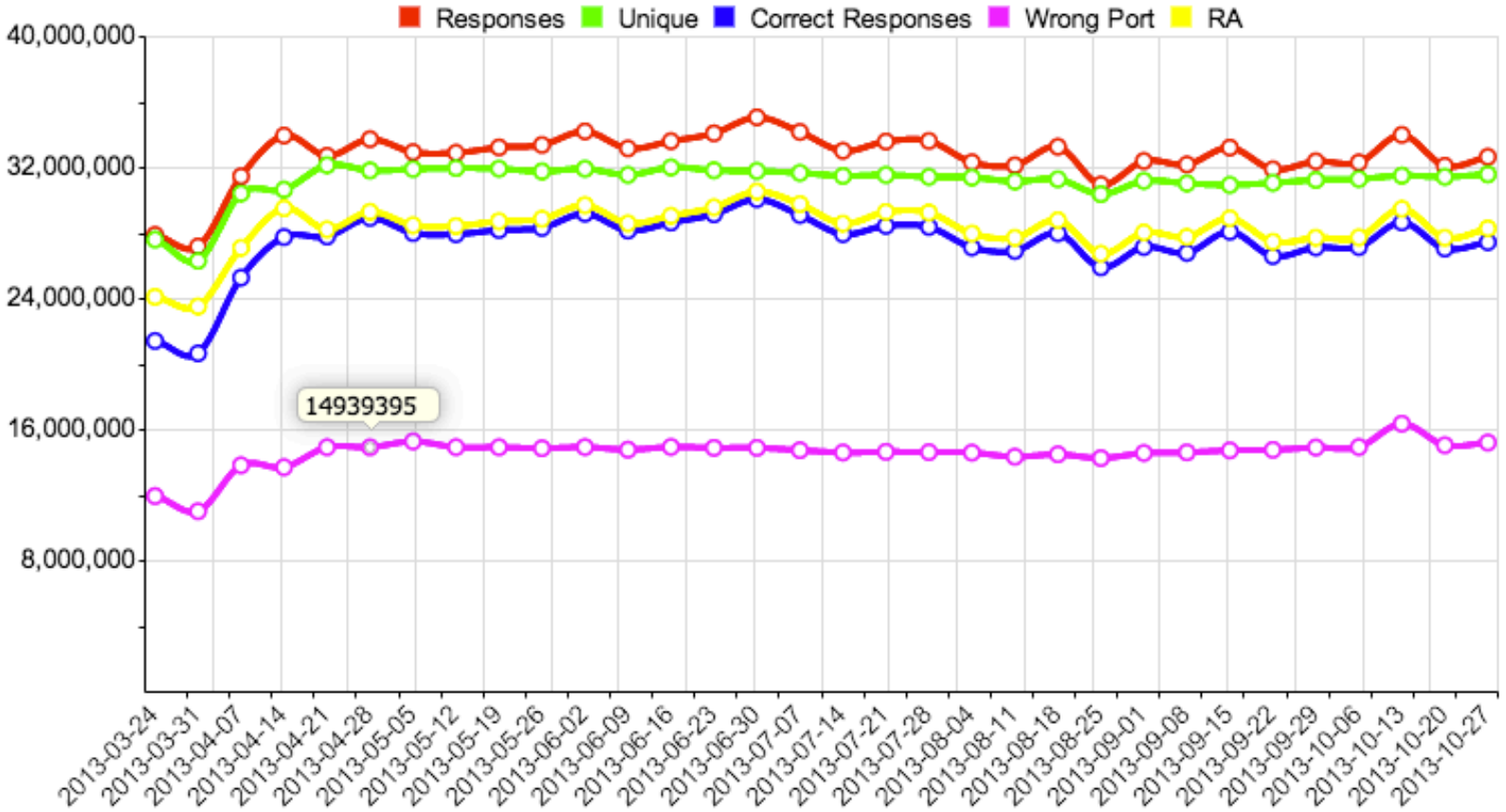
- If using older versions, a patch is available from
  - http://ss.vix.su/~vjs/rrlrpz.html
  - `patch -p0 -l`

Open Resolvers

- DNS servers that answer recursive queries from any host on the Internet
- http://openresolverproject.org/
- Check if you're running open resolvers
  - http://dns.measurement-factory.com/cgi-bin/openresolvercheck.pl
- More statistics at
  - http://dns.measurement-factory.com/surveys/openresolvers/ASN-reports/latest.html

Open Resolvers

As of 27 Oct 2013:
32,673,337 servers responded to udp/53 probe
28,681,520 returned OK



Reference: http://openresolverproject.org/

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)
- This 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
- The data collection ran until July 2012

Rogue DNS Servers

- 85.225.112.0 through 85.255.127.255
- 67.210.0.0 through 67.210.15.255
- 93.188.160.0 through 93.188.167.255
- 77.67.83.0 through 77.67.83.255
- 213.109.64.0 through 213.109.79.255
- 64.28.176.0 through 64.28.191.255

- If your computer is configured with one of these DNS servers, it is most likely infected with DNSChanger malware

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"
```

http://www.openspf.org/

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

Passive DNS

- Passive DNS replication is a technology invented in 2004 by Florian Weimer.
  - Many uses! Malware, e-crime, legitimate Internet services all use the DNS.
- Inter-server DNS messages are captured by sensors and forwarded to a collection point for analysis.
- After being processed, individual DNS records are stored in a database.

https://www.dnsdb.info/

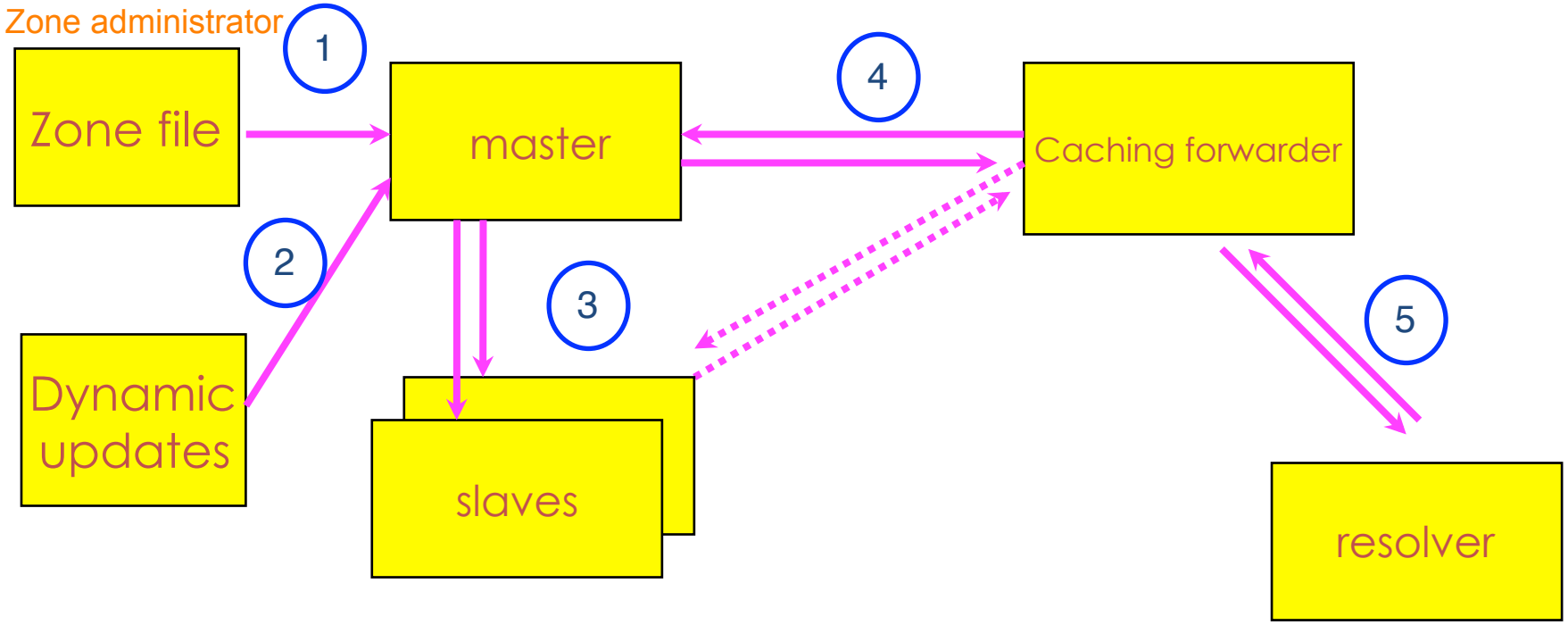
Good DNS Practice

- Use physically different machines for authoritative and recursive functions
- Use multiple authoritative servers to distribute load and risk:
  - Put your name servers geographically apart from each other
- Utilize caches to reduce load to authoritative servers and reduce response times
- Limiting views to control what data systems can be known
- Restrict resolution to specific address ranges if needed
- Be wary of incorrect use and monitor authoritative name servers to ensure correct behavior

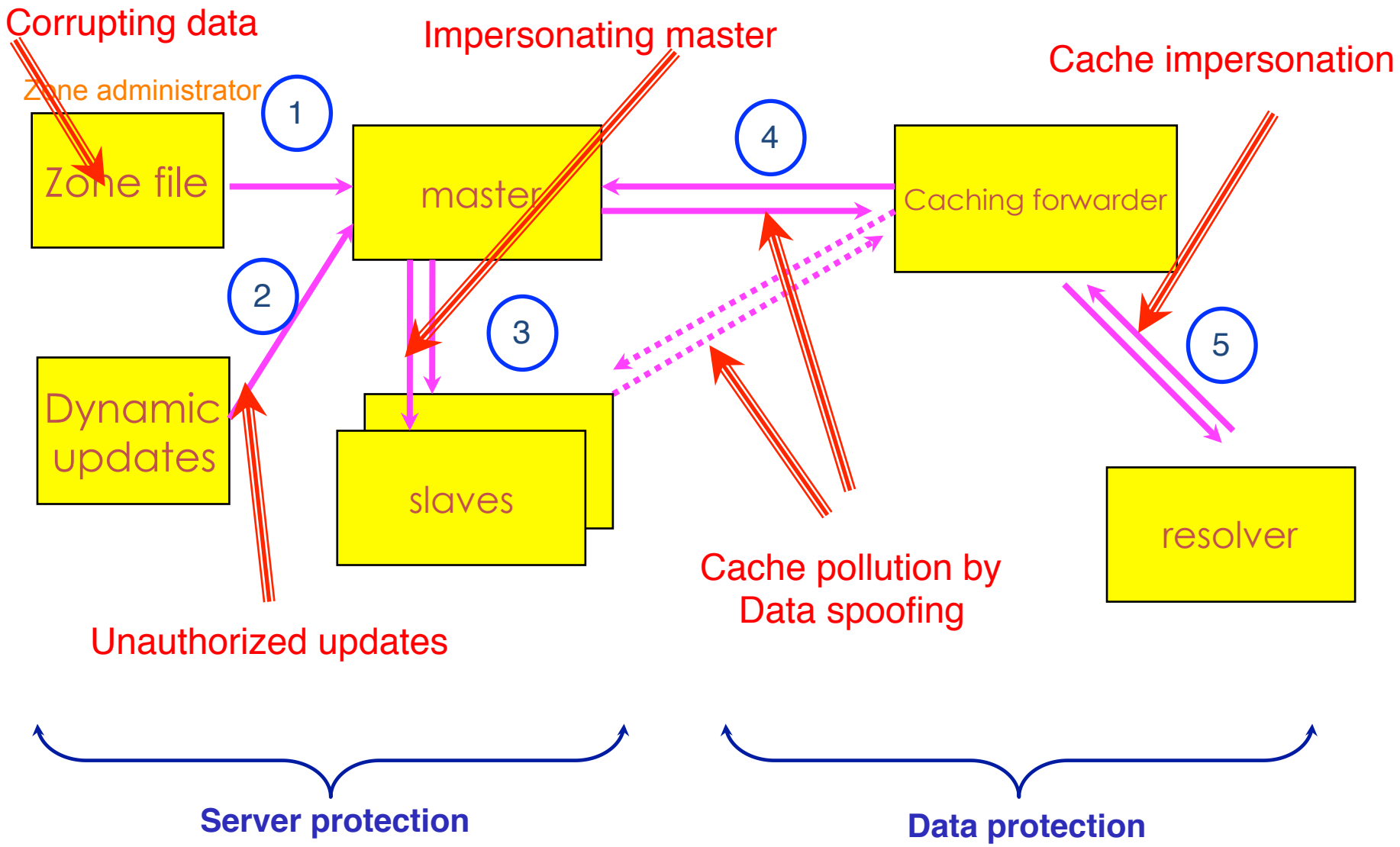
Securing the Nameserver

- Run the most recent version of the DNS software
  - Bind 9.9.1 or Unbound 1.4.16
  - Apply the latest patches
- Hide version
- Restrict queries
  - `Allow-query { acl_match_list; };`
- Prevent unauthorized zone transfers
  - `Allow-transfer { acl_match_list; };`
- Run BIND with the least privilege (use `chroot`)
- Randomize source ports
  - don't use `query-source` option
- Secure the box
- Use TSIG and DNSSEC

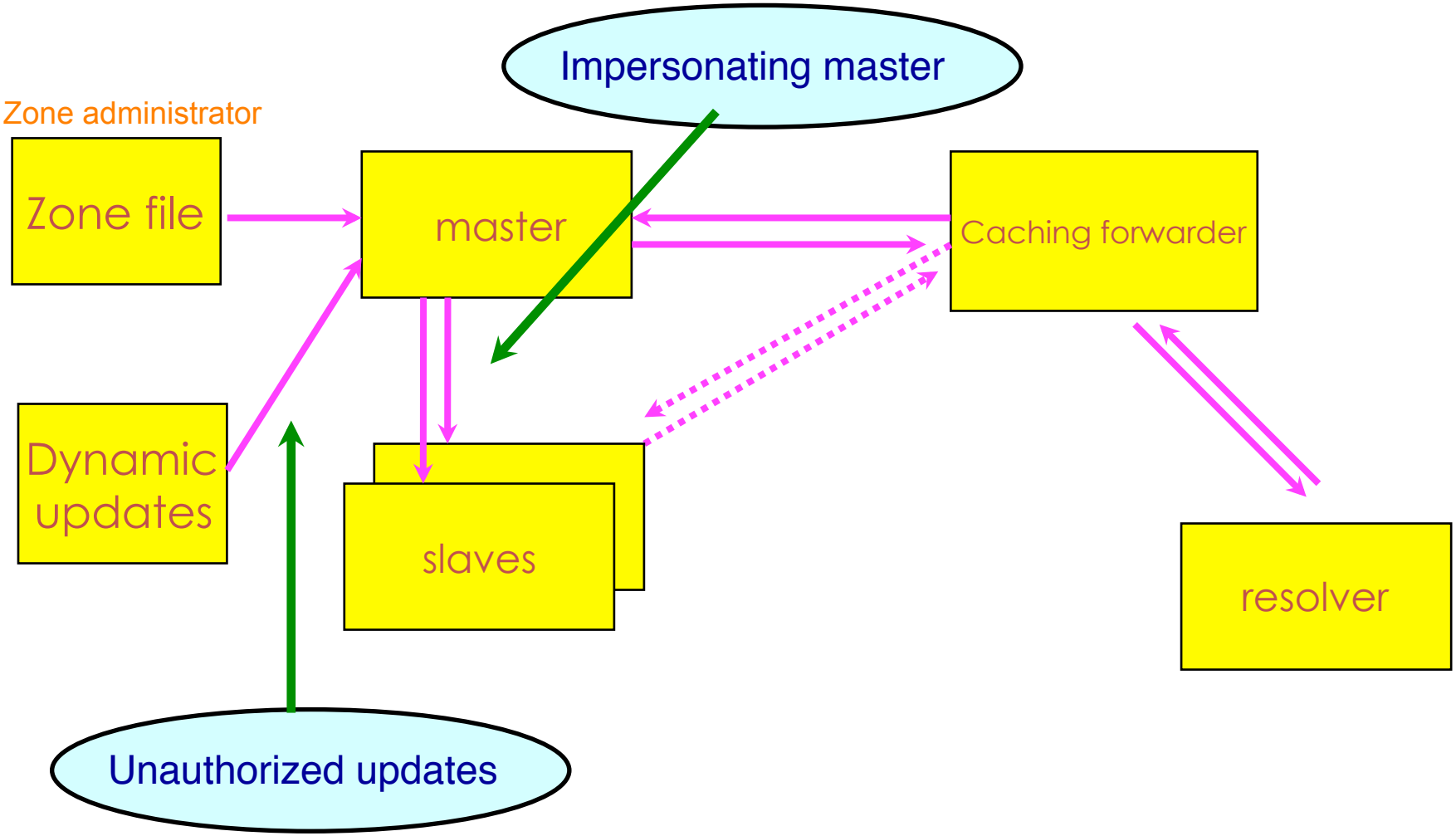
DNS: Data Flow



DNS Vulnerabilities



TSIG Protected Vulnerabilities



What is TSIG - Transaction Signature?

- A mechanism for protecting a message from a primary to secondary and vice versa
- A keyed-hash is applied (like a digital signature) so recipient can verify message
  - DNS question or answer
  - and the timestamp
- Based on a shared secret - both sender and receiver are configured with it

Transaction Signatures (TSIG)

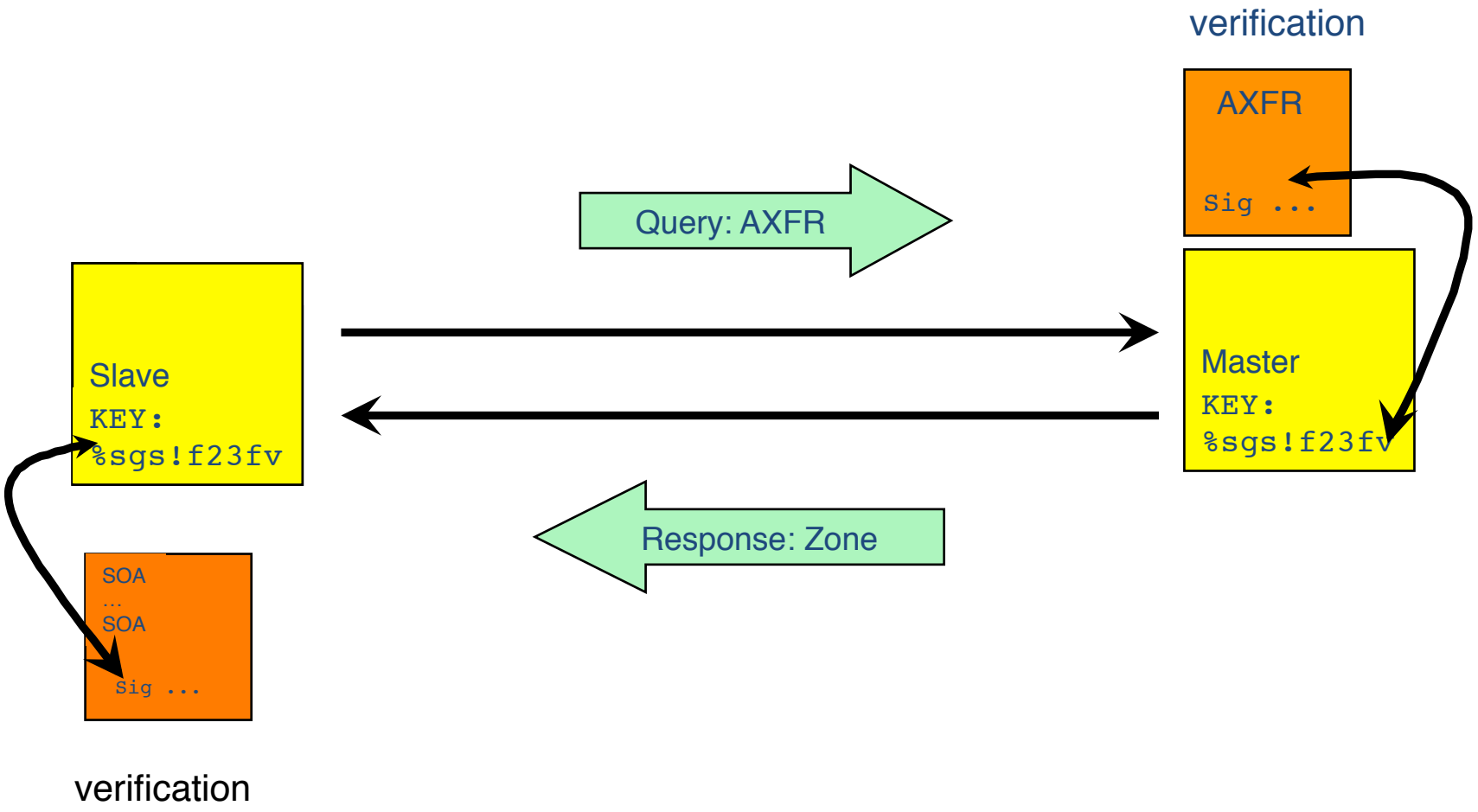
- TSIG is most-commonly used to authenticate slave servers to master servers during zone transfers
  - Protects against impersonating master and unauthorized updates
- Master and slave servers:
  - share a common secret key & agree on key name
  - Synchronized clocks (NTP)
- The shared information (key) is used to authenticate a client to a server
  - Remember to change the key periodically

What is TSIG - Transaction Signature?

- TSIG (RFC 2845)
  - authorizing dynamic updates & zone transfers
  - authentication of caching forwarders

- Used in server configuration, not in zone file

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
  - 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-MD5 -b 128 -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 should never be put in zone files!!!
  - might be confusing because it looks like RR:

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

TSIG – Configuring Servers

- Configuring the key
  - in named.conf file, same syntax as for rndc
  - `key { algorithm ...; secret ...;}`
- Making use of the key
  - in named.conf file
  - `server x { key ...; }`
  - where 'x' is an IP number of the other server

Configuration Example – named.conf

Primary server 10.33.40.46

```
key ns1-ns2.pcx. net {
    algorithm hmac-md5;
    secret "APlaceToBe";
};
server 10.33.50.35 {
    keys {ns1-ns2.pcx.net;};
};
zone "my.zone.test." {
    type master;
    file “db.myzone”;
    allow-transfer {
    key ns1-ns2..pcx.net ;};
};
```

Secondary server 10.33.50.35

```
key ns1-ns2.pcx.net {
    algorithm hmac-md5;
    secret "APlaceToBe";
};
server 10.33.40.46 {
  keys {ns1-ns2.pcx.net;};
};
zone "my.zone.test." {
    type slave;
    file “myzone.backup”;
    masters {10.33.40.46;};
};
```

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

- A wrong key will give “Transfer failed” and on the server the security-category will log this.

TSIG Testing - TIME!

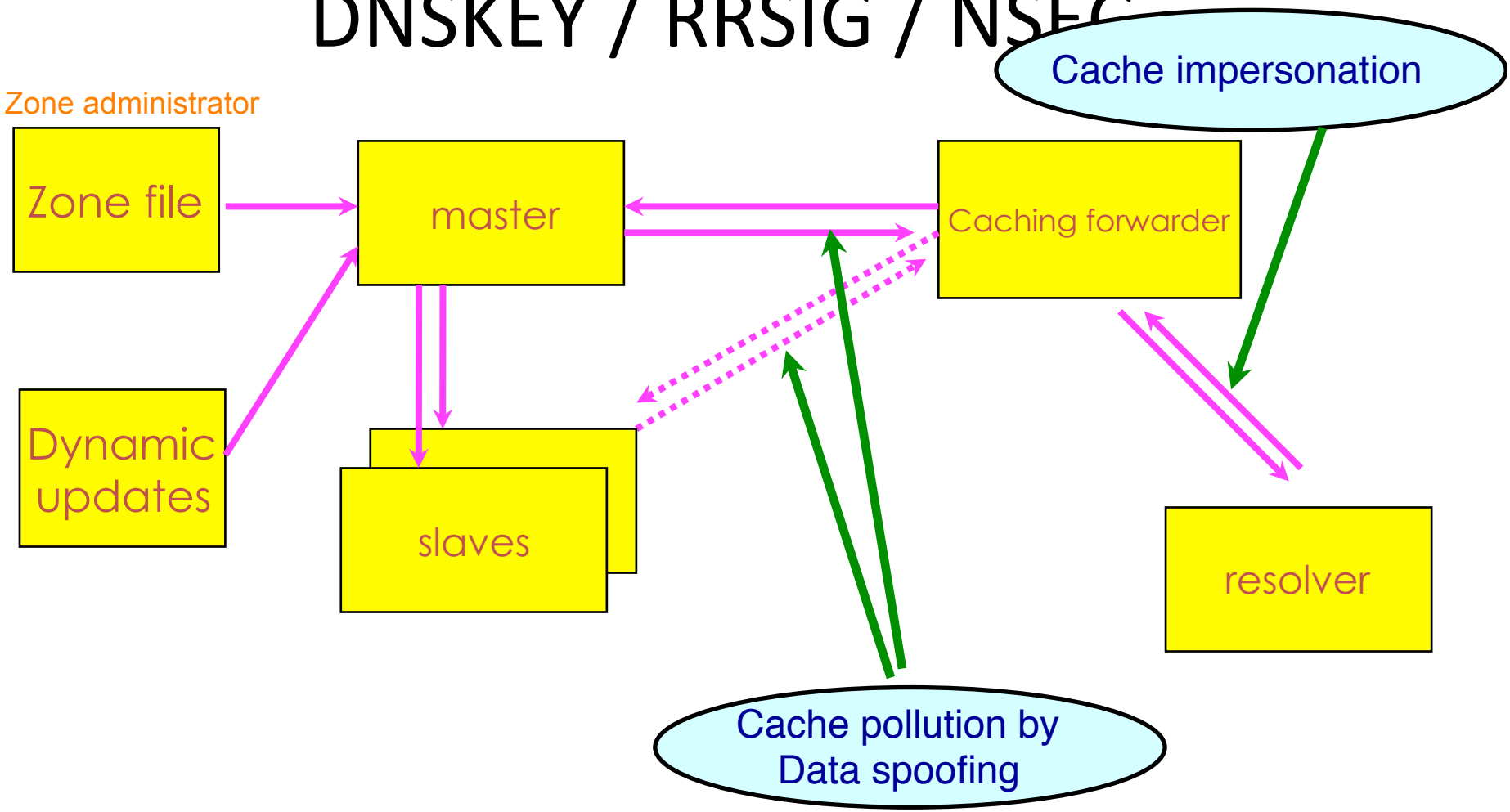
- TSIG is time sensitive - to stop replays
  - Message protection expires in 5 minutes
  - Make sure time is synchronized
  - For testing, set the time
  - In operations, (secure) NTP is needed

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 <TSIG keyfile>`

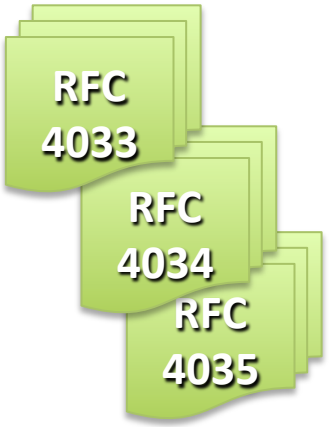
DNSSEC

Vulnerabilities protected by DNSKEY / RRSIG / NSEC



DNS Security Extensions (DNSSEC)

- Protects the integrity of data in the DNS by establishing a chain of trust
- Uses public key cryptography – each link in the chain has a public/private key pair
- A form of digitally signing the data to attest its validity
- Standard is defined in RFC4033, RFC4034, and RFC4035
- Guarantees
  - Authenticity
  - Integrity
  - Non-existence of a domain



DNSSEC History

- **1990**: Steven Bellovin discovers a major flaw in the DNS
- **1995**: Bellovin publishes his research; DNSSEC (as it became later known) becomes a topic within IETF
- **1997**: RFC 2065 (adding security extensions) was published
- **1998**: Dan Kaminsky discovers some security flaw
- **1999**: RFC 2535, the DNSSEC protocol, is published; BIND 9 developed to be DNSSEC-capable
- **2001**: key handling in RFC2535 is causing operational problems
- **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

Reflection Attacks

- DNS servers can act as very efficient packet amplifiers
  - Use of UDP, small queries, large responses
- DNSSEC makes DNS servers better packet amplifiers
  - Still lots of UDP, larger responses

Reliability

- In the grand scheme of things, DNSSEC does not help make your DNS more reliable
  - in fact it makes the DNS more brittle, and makes it harder to maintain reliable service

Confidentiality

- DNSSEC does not address confidentiality of queries or responses
  - anybody who can intercept a secure response can still see the details
  - there is no encryption here

Integrity, Authenticity

- DNSSEC provides a mechanism for data published in the DNS to carry cryptographic signatures
  - secure responses include signatures
  - clients receiving a secure response can tell whether it is authentic

Benefits to End-Users

- Users who validate will not see answers from the DNS that fail validation
  - might increase helpdesk load, but the alternative is infected computers, stolen bank details, etc
- Ongoing work to improve SSL security using DNSSEC-signed certificates
  - IETF “dane” working group

Benefits to Content Providers

- Reduce the risk that your content is being intercepted by unknown third parties
  - for end-users that validate, at least
- Demonstrate technical proficiency and security awareness

DNSSEC Resource Records

- 3 Public key crypto related RRs
  - **RRSIG** = Signature over RRset made using private key
  - **DNSKEY** = Public key, needed for verifying a RRSIG
  - **DS** = Delegation Signer; 'Pointer' for building chains of authentication
- One RR for internal consistency
  - **NSEC** = Next Secure; indicates which name is the next one in the zone and which typecodes are available for the current name
    - authenticated non-existence of data

DNSSEC Resource Records

- DNSKEY, RRSIG, and NSEC records provide mechanisms to establish authenticity and integrity of data
- DS record provides a mechanism to delegate trust to public keys of third parties

RR's and RRsets

- Resource Record:

Name	TTL	class	type	rdata
www.example.net.	7200	IN	A	192.168.1.1

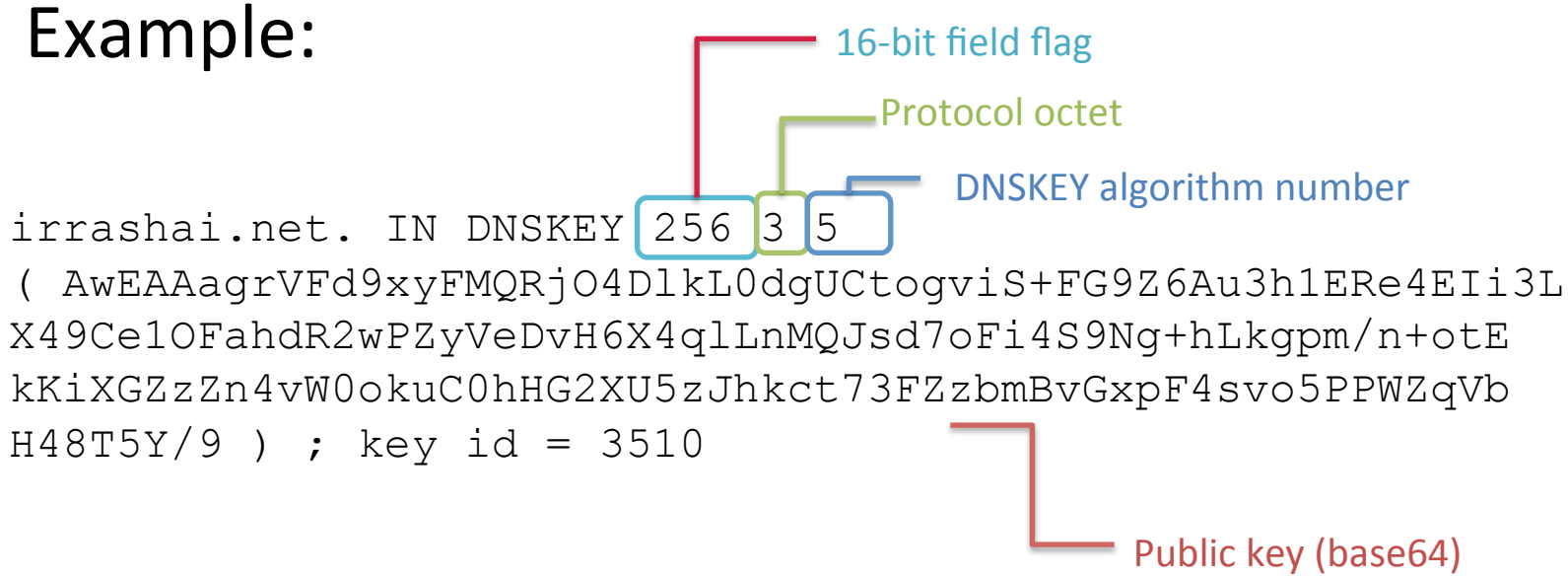
- RRset: RRs with same name, class and type:

www.example.net.	7200	IN	A	192.168.1.1
		IN	A	10.0.0.3

RRsets are signed, 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:



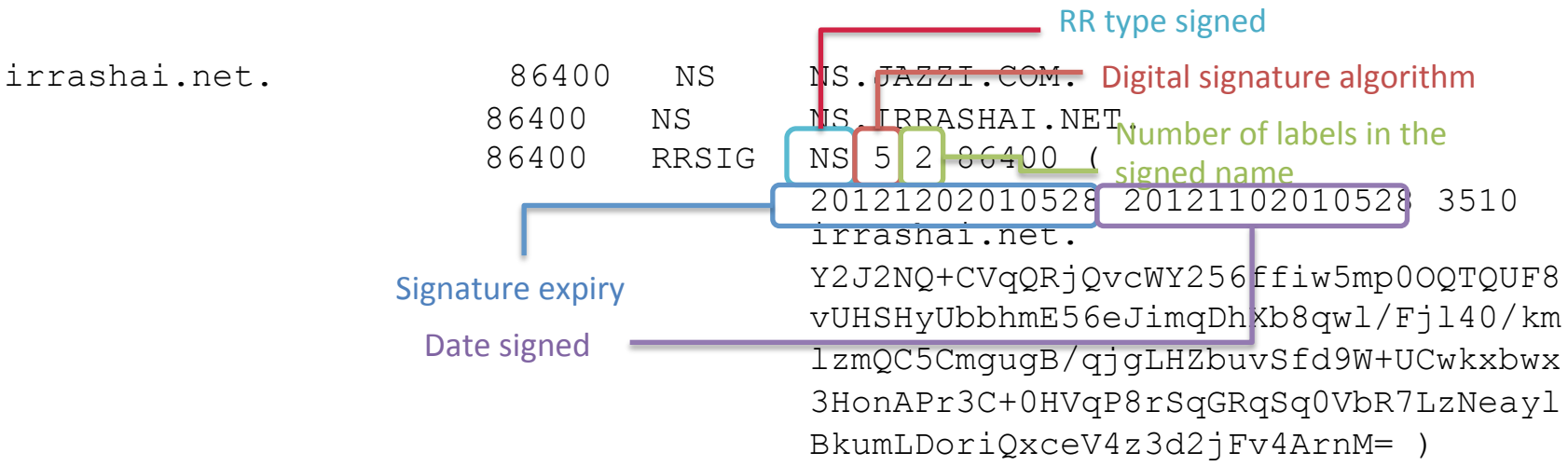
DNSKEY

- 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) per zone
- The digital signature per RRset is saved in an RRSIG record



NSEC / NSEC3

- Next Secure
- 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
- "The last NSEC wraps around from the last name in the ordered zone to the first"

NSEC and NSEC3

- NSEC3 is a more secure
  - Prevents zone walking
  - More computationally expensive
- Simple rule of thumb
  - if you are happy for anybody in the world to obtain a copy of your zone, and your zone is not very big, use NSEC
  - if you normally don't allow (e.g.) zone transfers to random people, or if you have a large zone to sign, use NSEC3

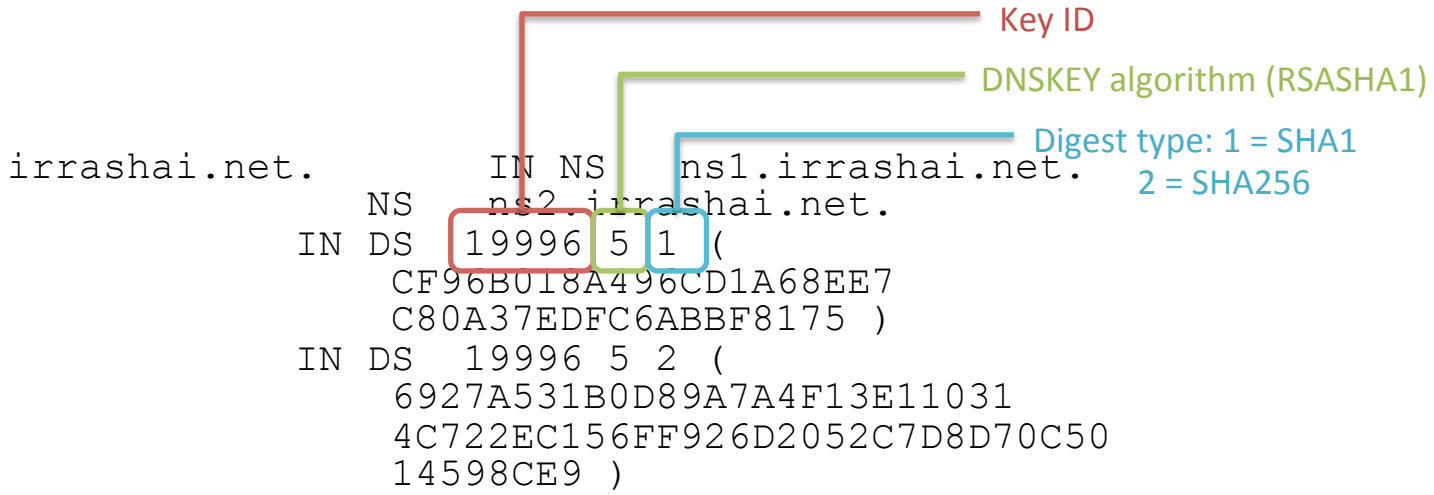
NSEC Record example

```
$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
```

Delegation Signer (DS)

- Establishes the chain of trust from parent to child zones
- Found in the parent's zone file
- In this example, irrashai.net has been delegated from .net. This is how it looks like in .net zone file



Delegation Signer (DS)

- Delegation Signer (DS) RR indicates that:
  - delegated zone is digitally signed
  - 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 in the child zone

Types of Keys

- **Zone Signing Key (ZSK)**
  - Sign the RRsets within the zone
  - Public key of ZSK is defined by a DNSKEY RR
- **Key Signing Key (KSK)**
  - Signed the keys which includes ZSK and KSK and may also be used outside the zone

Creation of Keys

- Trusted anchor in a security aware server
- Part of the chain of trust by a parent name server
- Use of a single key or both keys is an operational choice (RFC allows both methods)

Chain of Trust

- DNSSEC is based on trust
- Root is on top of the chain of trust.
  - Root servers were signed on July 15, 2010.

For ISPs - Validate

- The most effective step you can take to encourage DNSSEC uptake as an ISP is to validate responses
  - DNSSEC-signed zones are fairly new, so expect this to cause some non-zero (but manageable) amount of helpdesk load
  - Comcast is an example of a large ISP (in the US) who has taken this step

Registries / Hosting Providers – Sign your Zones

- All the zones you serve can be signed
  - think about key rollover
  - think about key compromise scenarios, and what processes you will follow when you detect them
  - think about how you can detect compromises, and monitor signatures

Key Management

- need to implement secure key storage, management procedures
- need to sign your zones
- registries need to accept DS records from users (how?)
- need to publish DS records to parents (how?)

Key Management

- DNSSEC has many parameters to consider, including:
  - key rollover schedule
  - signature duration
  - choosing appropriate TTL for the zone data
  - key size
- Those will be determined by your policy
- You must determine them for your own organisation, via a risk and operational assessment
- Don't blindly copy the policies of another organisation!

Key Management

- How do we keep the ZSK secure?
- How do we keep the KSK secure?
  - important questions
  - no simple answers here
  - requires risk analysis, consultation, maybe audit
  - again, a matter of policy
    - hybrid models possible
      - HSM for KSK, software for ZSK

Communication

- Communicate with your customers
  - explain benefits/risks of DNSSEC
- Communicate with end-users
  - demonstrate how to validate responses
  - explain operational changes (firewalls, TCP, response sizes)

Legal Aspects

Legal Aspects

- Deployment of DNSSEC involves trust in procedures and policies
  - otherwise why trust signatures?
- DNSSEC Policy and Practice Statement (DPS)
  - a public attestation of procedures and policies
  - can be used as the basis for audits

DPS

- draft-ietf-dnsop-dnssec-dps-framework-04
  - (work in progress, locate using Google)
- DPS for the Root Zone KSK Operator
  - https://www.iana.org/dnssec/
- Also review published DPS documents from TLDs who have already deployed DNSSEC

DPS

- .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

Questions