Do not shun DNS

Code for Venezuela: San Francisco Hackathon 2019

Ernesto Hernández-Novich
<emhn@usb.ve>

Universidad "Simón Bolívar"

UNIVERSIDAD SIMÓN BOLÍVAR

Domain Name System – DNS

. . . everyone ~~needs~~ depends on it.

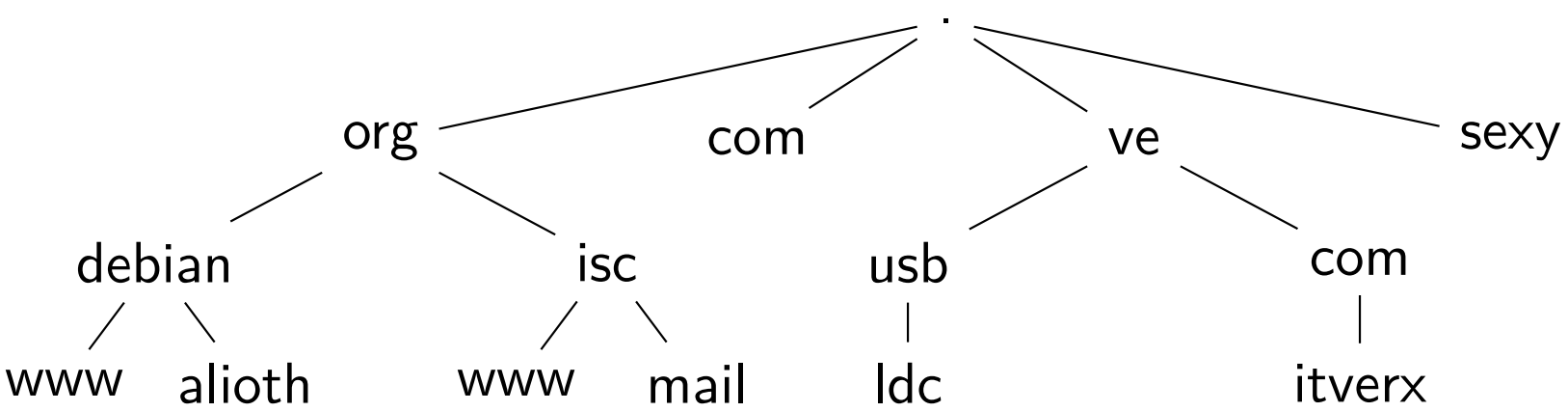
- 30+ years going strong – RFC-1034 & 1035 (1987)
- Distributed global hierarchical database
- Optimized for searching **domain names**
  - Dot separated **labels** – `www.isc.org`
  - There's a dot at the end – it's there.
  - Case-insensitive
- Users query using UDP 53 – servers interact using TCP 53

From names to IP addresses,
and back again.

UNIVERSIDAD SIMÓN BOLÍVAR

Distributed and global?

The name space is structured as a tree

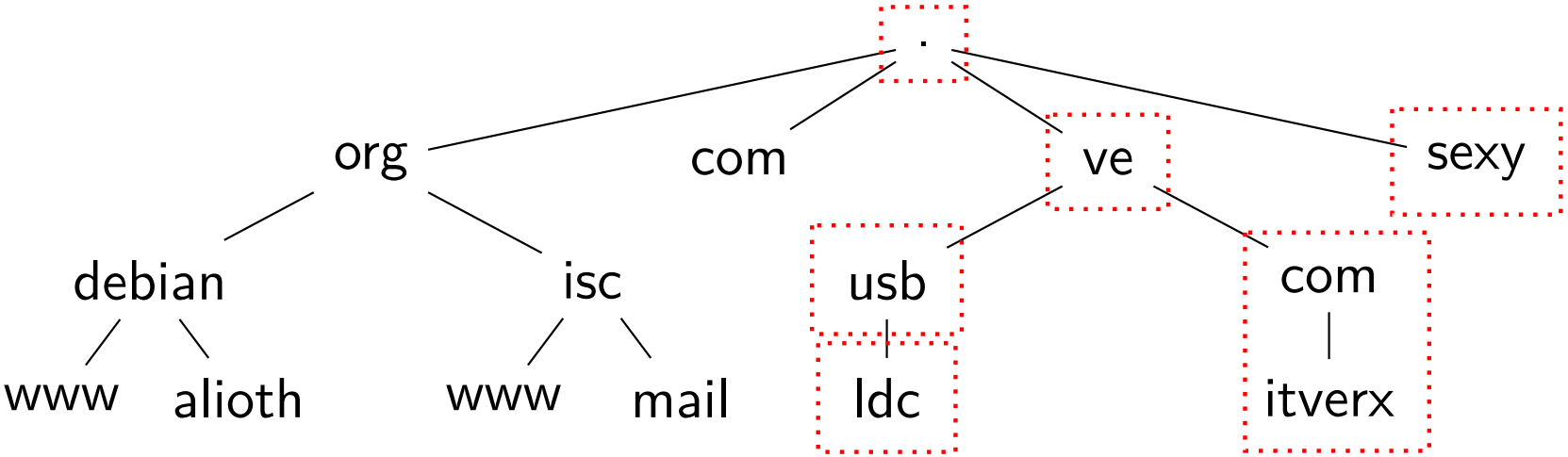


- DNS is a tree of labels – can you spot the dot now?
- Labels are unique within sibling groups.
- Top Level Domains (TLDs) – .com, .net, .edu, .org
- Country Code TLD (ccTLDs) – .ve, .us, .ch
- Generic TLD (gTLDs) – .sexy, .tattoo, .link, .juegos, .sucks, . . .

UNIVERSIDAD SIMÓN BOLÍVAR

How is it distributed?

Names, domains and zones...



- Domain Name – traverse tree from label to root.
- Domain – any given node in the tree.
- Zone – a sub-tree under autonomous administration.
- Delegation – child node's zone different from parent's

Each domain has associated information.

UNIVERSIDAD SIMÓN BOLÍVAR

What kind of information?

Records, DUH!

- **Resource Records** – RRs for short
- Name, time to live, type, class, and actual resource.
- Hgh-performance binary representation with textual representation.

```
www.usb.ve.      86400 IN CNAME usb.dsm.usb.ve.
usb.dsm.usb.ve. 86400 IN A 159.90.210.22
usb.dsm.usb.ve. 86400 IN A 159.90.210.24
```

- A name can have several RRs, all with same TTL.

UNIVERSIDAD SIMÓN BOLÍVAR

Frequent RRs

New RRs are proposed and adopted when convenient

SOA	**S**tart **O**f **A**uthority
NS	Name Servers for zone
A	IPv4 address for name
AAAA	IPv6 address for name
MX	**M**ail **Ex**changer
SRV	Generic service locator (LDAP, XMPP, ...)
TXT	Plain text information

- Zone admin **must** add RRs for relevant names – each RR is valid for its TTLs (caching!)
- Users typically want the value associated with an RR

UNIVERSIDAD SIMÓN BOLÍVAR

How does it work?

There are Servers

- Servers in charge of a **zone**
  - One or more **authoritative** servers per zone.
  - They are listed as "official servers" for the zone.
  - One is the actual master – holds the database
  - The rest are replicas – they get notified and transfer changes.
- Delegated zones
  - Superior zone points to inferior (we call it glue).
  - The `root` zone points to servers for `com`, `ve`, `sexy`, `...`
  - Zone `ve` points to servers for `usb.ve`, `itverx.com.ve`, `...`
  - Zone `usb.ve` points to servers for `ldc.usb.ve`.

UNIVERSIDAD SIMÓN BOLÍVAR

Querying the database

This happens in your device

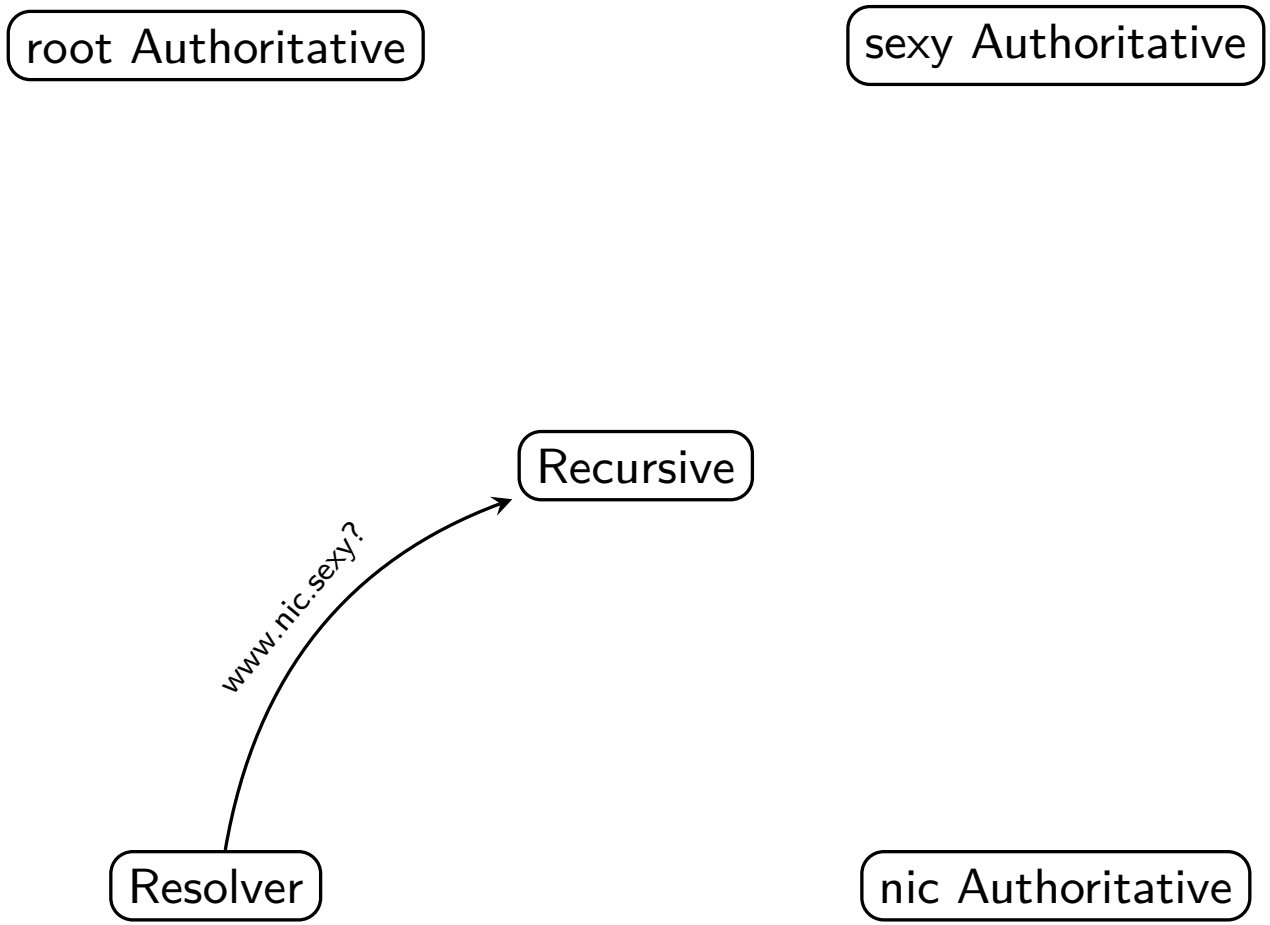
- There's a stub resolver on every OS – it's a library.
- Programs ask for a name to be "resolved" – desired RRs are returned, if they exist.
- Library needs access to a recursive server.
  - Receives queries for any zone.
  - Follows the delegation chain to find authoritatives.
  - It caches the answer for future reference.
- What recursive is going to be used?
  - Usually provided via DHCP from your ISP or network administrator.
  - Can always be changed to a better one – for many notions of "better"

But why ~~would~~ should you change it?

UNIVERSIDAD SIMÓN BOLÍVAR

A recursive query

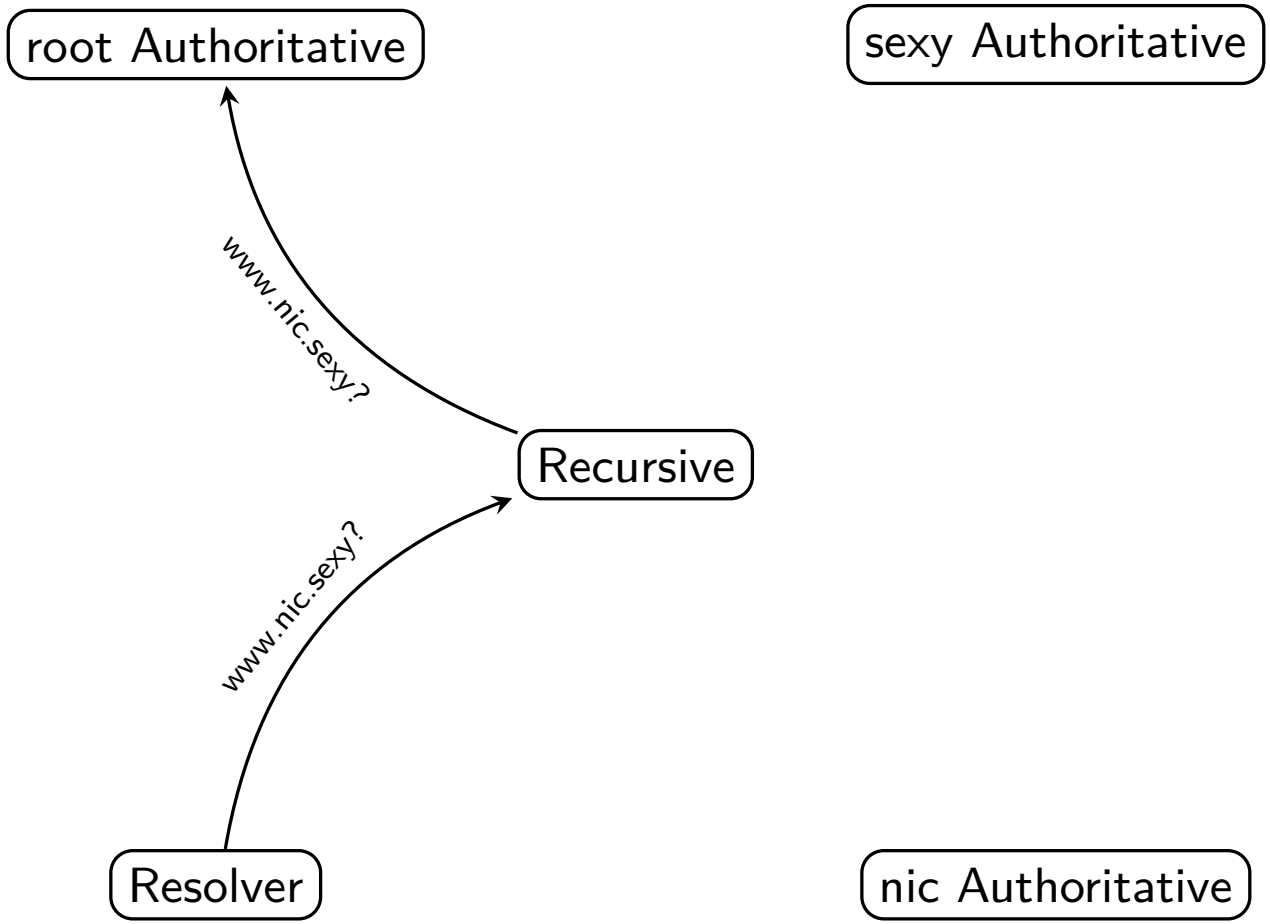
What's `www.nic.sexy` IP address?



UNIVERSIDAD SIMÓN BOLÍVAR

A recursive query

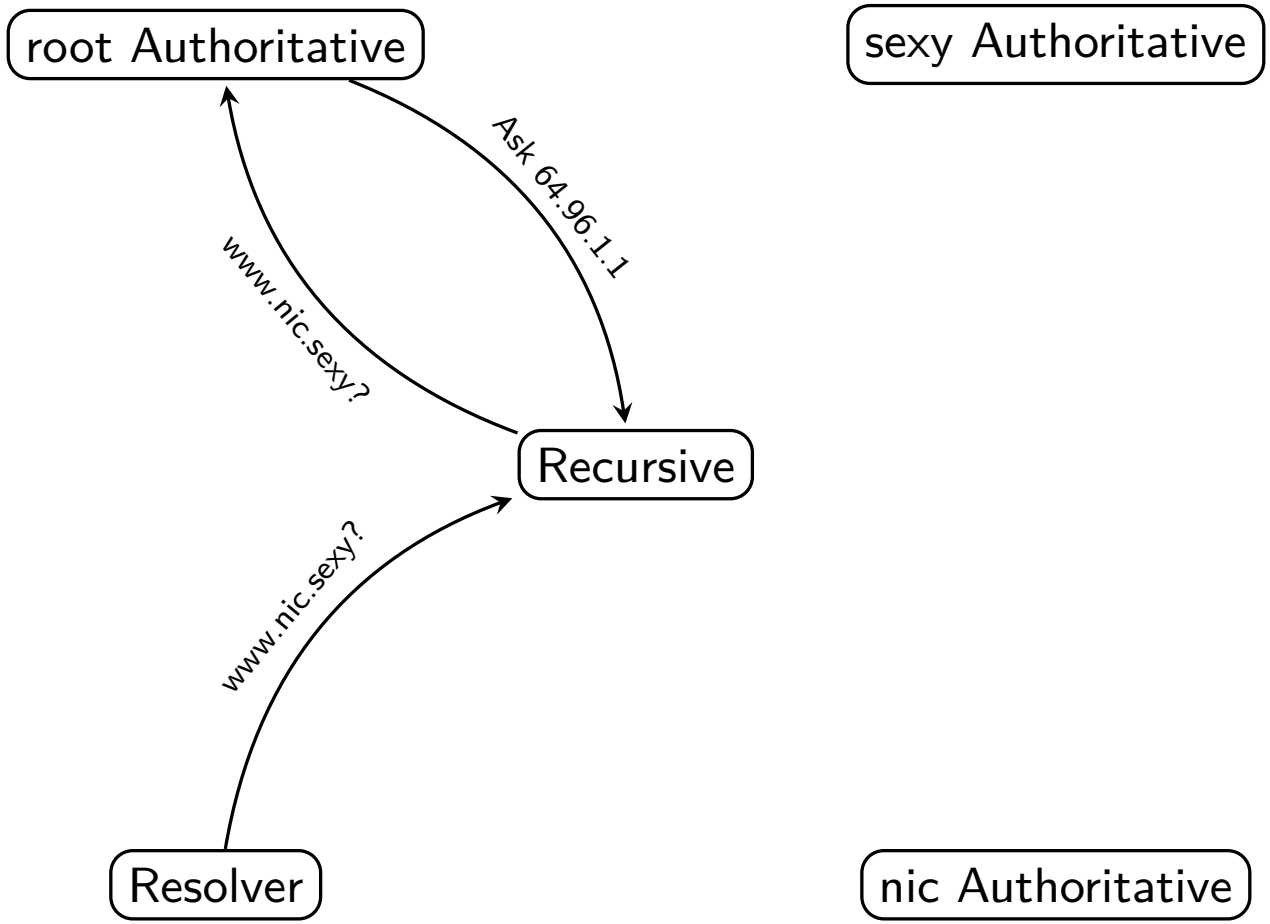
Recursive alwways queries `root`



UNIVERSIDAD SIMÓN BOLÍVAR

A recursive query

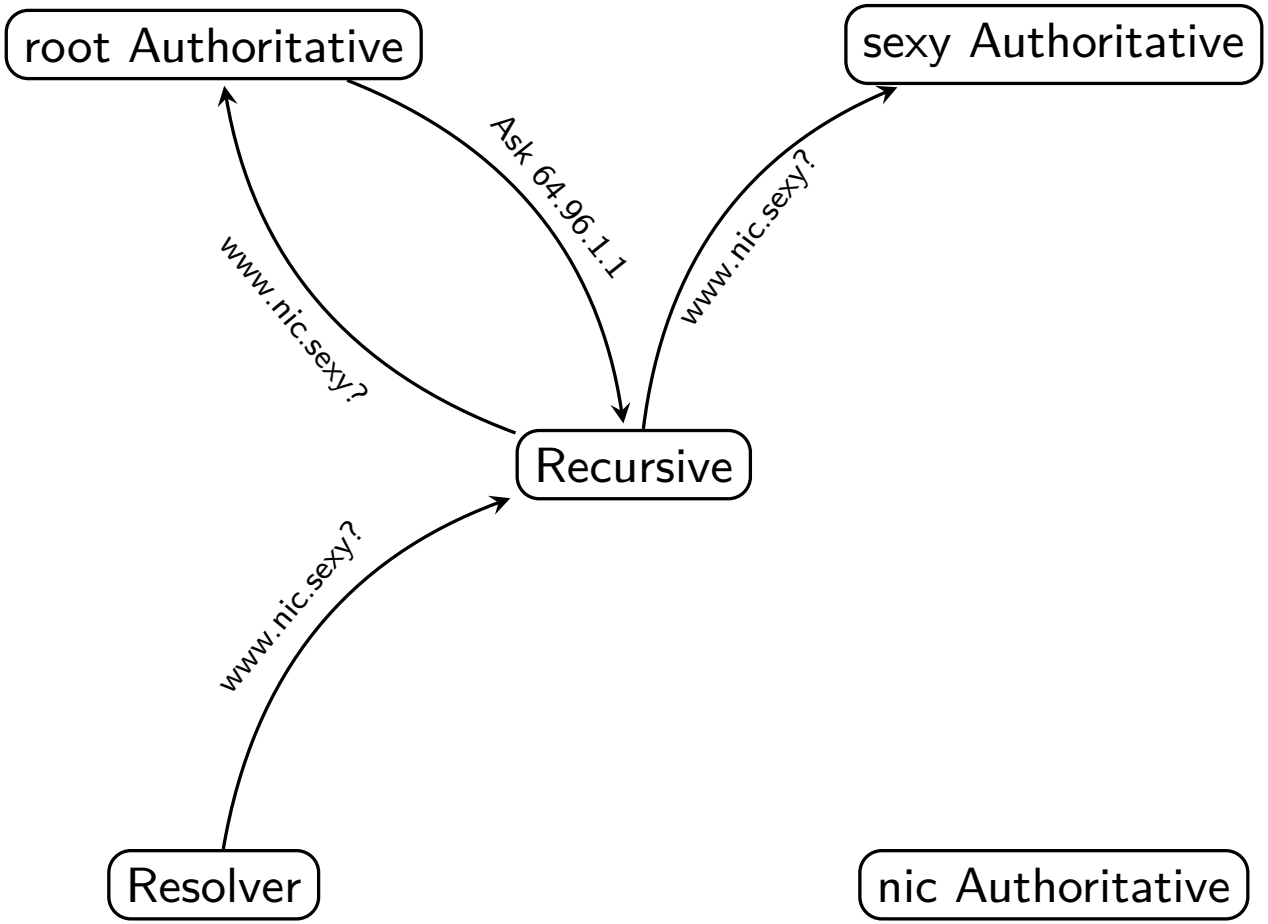
`root` refers to `sexy` authority



UNIVERSIDAD SIMÓN BOLÍVAR

A recursive query

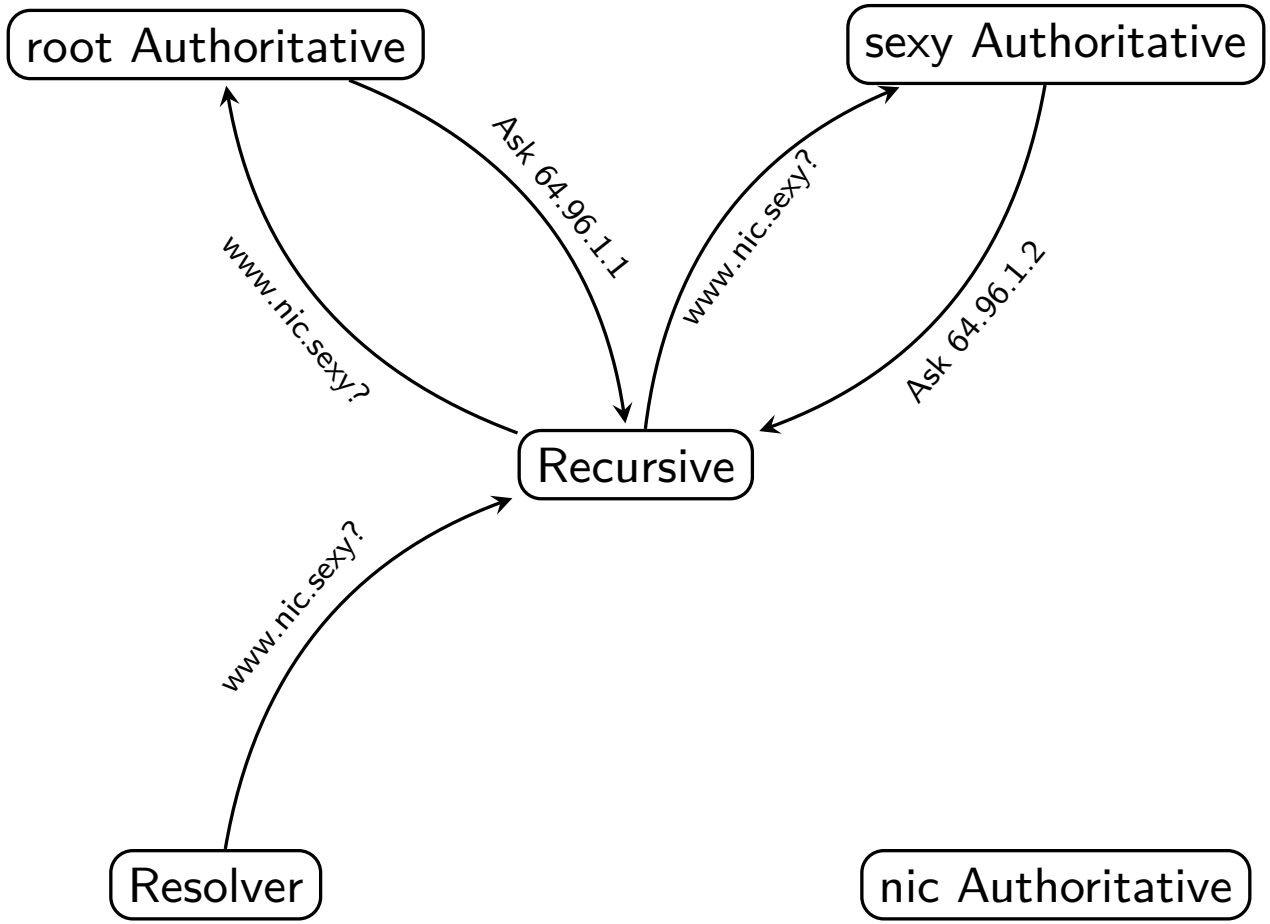
Recursive queries `sexy`



UNIVERSIDAD SIMÓN BOLÍVAR

A recursive query

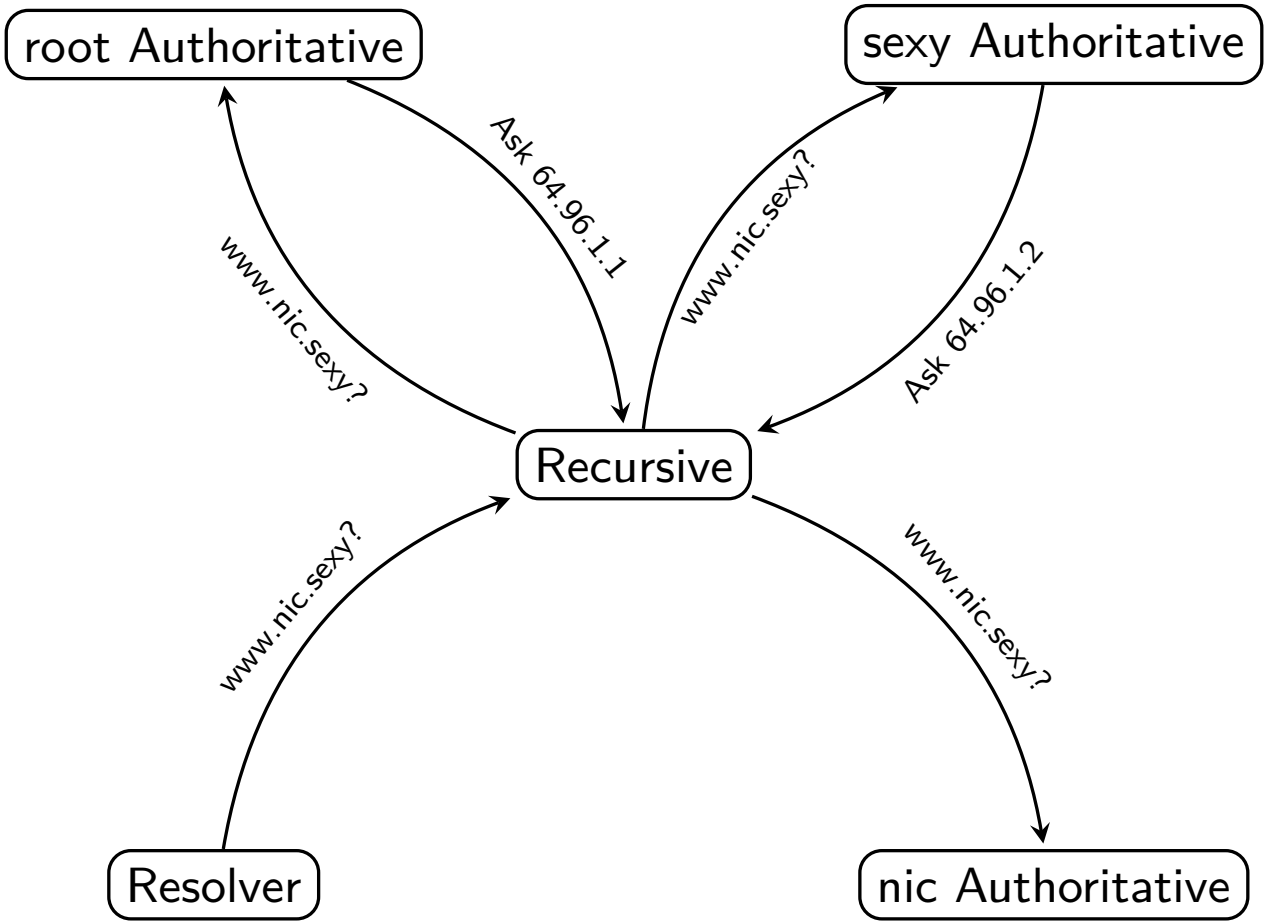
`sexy` referral to `nic.sexy`



UNIVERSIDAD SIMÓN BOLÍVAR

A recursive query

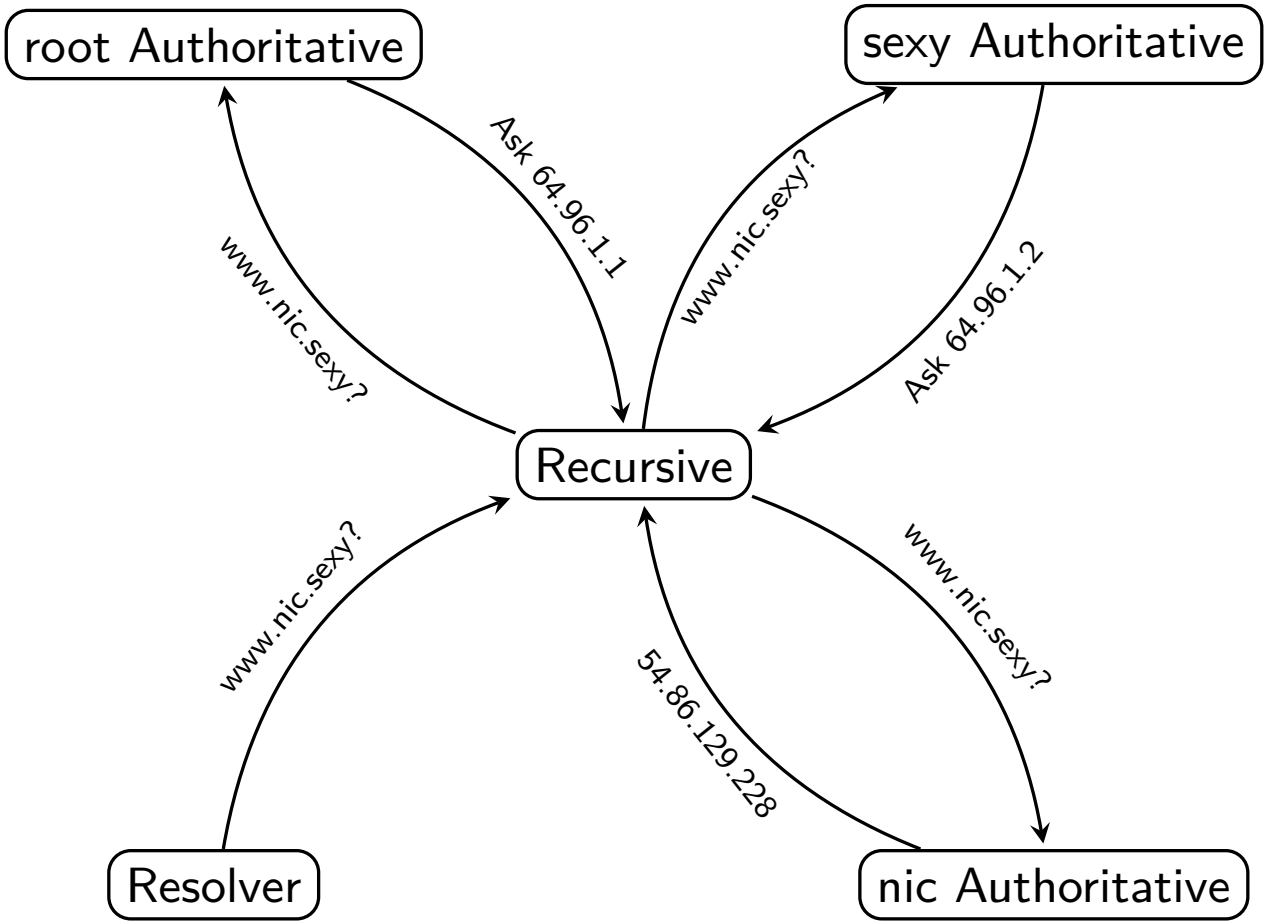
Recursive queries `nic.sexy`



UNIVERSIDAD SIMÓN BOLÍVAR

A recursive query

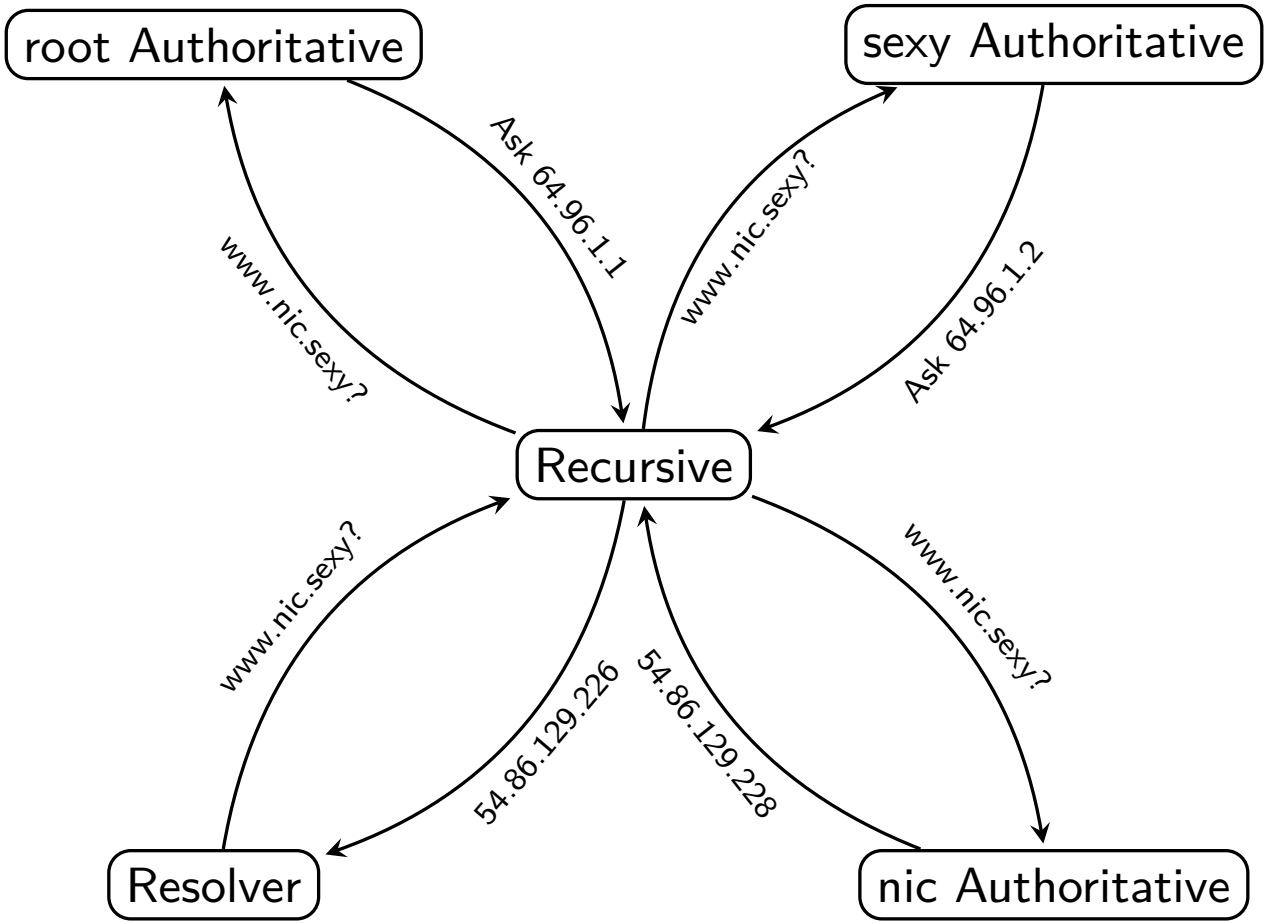
`nic.sexy` answers with authority



UNIVERSIDAD SIMÓN BOLÍVAR

A recursive query

Recursive relays answer to your device



UNIVERSIDAD SIMÓN BOLÍVAR

A recursive query

Keep in mind...

- There could be many answers for each query –
  more than one NS per zone, more than one IP per name...
- Queries happen over UDP – packet loss, congestion
- Recursive caches every answer locally –
  further queries are satisfied from the cache.
- Answers expire after the TTL – recursive purges cache.
- Failed queries are **also** cached.

Recursive servers are crucial.

UNIVERSIDAD SIMÓN BOLÍVAR

Things can go very bad

“Total, la gente, ¿qué sabe?” – Les Luthiers

- Most users couldn’t care less about DNS –
  invisible, hasn’t got a flashy UI, no BlueTooth®...
- But everyone **depends** on DNS – no DNS, no Internet for you!
- Most failures were due to poor network administration
  - Missing, incorrect, or contradicting RRs.
  - Not having replicas or having out of sync replicas.
  - Not enough authoritatives nor recursive servers.
  - Incomplete or incorrect delegation.

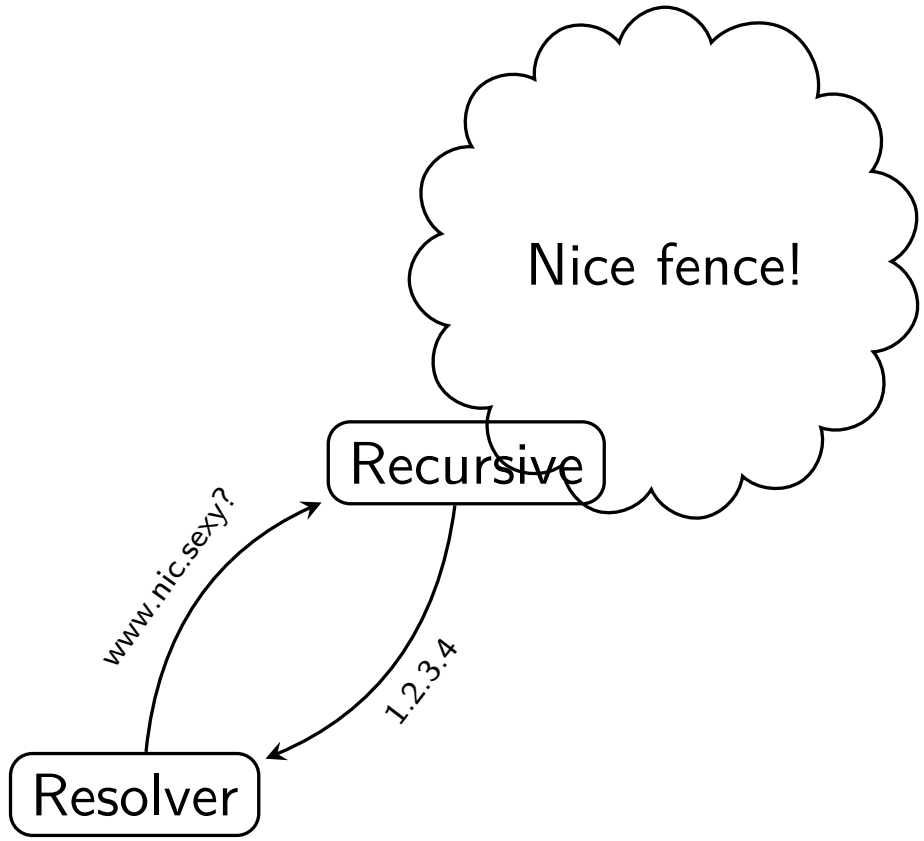
But as a user, you rely on recursives,
and you should be very, **VERY** afraid...

UNIVERSIDAD SIMÓN BOLÍVAR

Who controls the recursive?

Do they have a clue? Are they trustworthy?

- You get answers from the recursive – you have **no way** to know if it traversed the hierarchy correctly and completely.
- Is the answer **true and legitimate**?
- Is the answer **complete?**
- Is the negative answer **proof** of absence?



People: use a different recursive!

UNIVERSIDAD SIMÓN BOLÍVAR

Never use “bad people’s” recursives

. . . they could be trying to trick me or plain stupid

- “Use Google’s!” – 8.8.8.8
- “Use CloudFlare’s!” – 1.1.1.1
- “Use OpenDNS’!” – said the hipster. . .
- “Use your Access Point’s!” – expanding brain meme
- “Setup your own recursive!” – expanding brain meme

Anything but “those people’s” recursive, right?

UNIVERSIDAD SIMÓN BOLÍVAR

Using a different recursive is not enough

. . . because you have to go through the ~~fence~~ swamp

- Let's say you expanded your brain
  1. Your device uses your alternate recursive.
  2. A filter within the ISP detects the query – easy peasy.
  3. The filter forges an answer or behaves as authority.
  4. Your query never reached the recursive – brain explodes.
- That's a classical "man in the middle attack" (MITM).
  - Forging answers so you're directed somewhere else.
  - Making it look like a failure or unavailable site.

The fence between client and external recursives is hostile.

UNIVERSIDAD SIMÓN BOLÍVAR

But using mine is safe, right! Right?

. . . yeah, right, except when rarely ever

- So you setup your local network recursive with Linux or FreeBSD.
- Someone in the network could be doing exactly the same – can't deny nor confirm it was me.
- But let's assume I'm not doing that for the sake of argument
  1. Your device queries the recursive.
  2. The recursive queries the root – a filter within the ISP notices it.
  3. The filter follows the query sequence until the end.
  4. The filter changes the last answer to whatever ~~I want~~ it wants.
- Your cache's been poisoned
  - Forging answers, making you go elsewhere.
  - Making it look like failure or "blocking".
  - . . . but this time is worse – it will have a long TTL within your cache

UNIVERSIDAD SIMÓN BOLÍVAR

DNSSEC – Domain Name System Security

Extension to traditional DNS

Guarantees

- Answers you receive are **authentic**
- Answers you receive are **unmodified**
- Negative answers really mean a non-existent name.

Does not guarantee

- Query/answer **confidentiality**
- DoS protection

UNIVERSIDAD SIMÓN BOLÍVAR

DNSSEC - General idea

Some assembly required. Batteries not included.

- Asymmetric criptography
- Zone admin signs RR with the private key.
  - Zone cryptographically signed as a whole.
  - Each RR has its own signature.
- Public key included with the zone.
- Recursives are able to **check** answers using the public key.
- The chain of delegation builds a chain of trust.
- This requires a lot of new RRs.

UNIVERSIDAD SIMÓN BOLÍVAR

And here they are

Good system design included ways to extend DNS itself

DNSKEY	Zone Public Key
RRSIG	Digital signature for RR
NSEC	Next name – to verify non-existance
DS	Delegation signature – child zone fingerprint

UNIVERSIDAD SIMÓN BOLÍVAR

Zones grow complex

. . . but trust increases

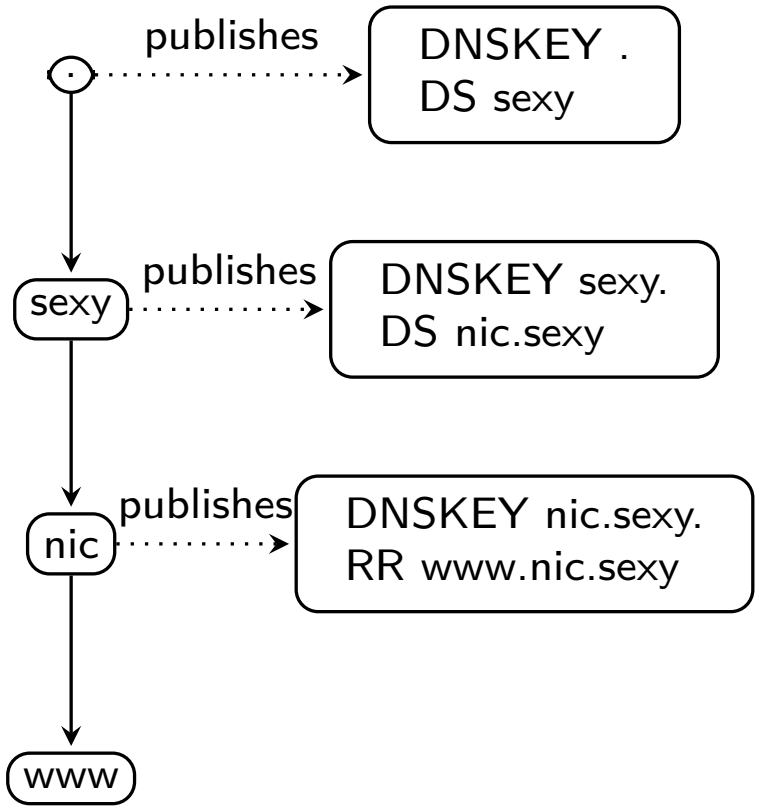
- One or more **DNSKEY** for different purposes
  - **KSK** – strong private key to sign other keys
  - **ZSK** – strong private key to sign the zone
  - The **KSK** signs the **ZSK**.
- One or more **RRSIG** for each RR set per name – signature plus validity.
- One or more **NSEC** per name
  - Points to the next signed name
  - Jump over "holes" – identifies non-existing names.
  - It's a circular list starting and ending at the SOA.
- One or more **DS** from parent to child zone.
  - Adult supervision required – parent zone must cooperate.
  - Parent zone confirms valid keys for child zone.

UNIVERSIDAD SIMÓN BOLÍVAR

The chain of trust

Built on top of delegation, rooted on ICANN's trust

- Root zone (.) hols DS records for every TLD using DNSSEC – publishes a DNSKEY to check those DS.
- Each TLD zone (`sexy`) holds DS for delegated zones – publishes its own DNSKEY to check those DS.
- Final zone (`nic.sexy`) publishes a DNSKEY.
- RR for (`www.nic.sexy`) is signed (RRSIG) – checked with `nic.sexy`'s DNSKEY



A recursive can perform those checks.

UNIVERSIDAD SIMÓN BOLÍVAR

What's the end result?

We can demand `dig` to validate DNSSEC – heavily edited for human consumption

```
$ dig +dnssec www.nic.sexy
...
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 2, AUTHORITY: 5, ADDITIONAL: 1
...
; EDNS: version: 0, flags: do; udp: 512
...
;; ANSWER SECTION:
www.nic.sexy.   3599  IN  A 54.86.129.228
www.nic.sexy.   3599  IN  RRSIG A 5 2 3600
 20190423204035 20190324194943 15813 nic.sexy.
 ZfSyTWXRrArz8LKXLeCP5eXau
 4j4eDAdHmHxrQJKqIJG+5H2BKs84dLN ZxfLssf+Ua5Lp8Q2+5v7iU41
 J0F2egmGKMstpm+nQ8Xbhmc3sxZsmluW VpzcOeQs7FCVX5VNE307tKV7A4kHcavkb61LEO/sD7Z8faJK2
```

- Authentic answer – flag **ad**.
- Untampered answer – flag **do**.
- RRSIG shows date validity ranges.
- NS RRSIGs omitted – more confirmation the name is valid.

UNIVERSIDAD SIMÓN BOLÍVAR

What if we query an unexistent name?

Make sure it does not exist

- "Certifiably unexistent" – zone holds "existing" names only.
- Names can be ordered (discrete math sends his regards!) – there is a "hole" where names are missing.
- NSEC (Next Secure) RRs fill these holes – point to the next **existing** name.
  - Querying and unexistent name using DNSSEC will answer with predecessor and successor names – yes, an information leak!
  - NSEC3 improves by using hashes instead of plain names.

UNIVERSIDAD SIMÓN BOLÍVAR

Got it! What do we need to do?

Be aware, be able to explain, be able to guide

- Stop relying on third-party resolvers, specially those than can be compromised (like those within CAN'TV).
- Deploy your own resolvers, forcing DNSSEC.
- Deploy your applications on domains served using DNSSEC.
  - Choose a TLD that supports DNSSEC.
  - Choose a Registrar that supports DNSSEC.
  - Use a DNS service provider supporting DNSSEC.
- Write your applications to look for names using DNSSEC and be very loud when unavailable – let the user now the name is not safe to use.
- Be wary of using any network service (webpage, API, e-mail) operating on a domain that is not DNSSEC-validated.
  - Doesn't matter if the service is TLS-encrypted.
  - Hijacking is easier with «Let's Encrypt» and no-DNSSEC.

UNIVERSIDAD SIMÓN BOLÍVAR

I need DNSSEC in my life

I need more cowbell

- I want **my zones** to be DNSSEC signed –
  deploy DNSSEC-aware **authoritative** servers.
  - BIND 9.10 or later – found in any Linux/FreeBSD.
  - Have several replicas in multiple geographic regions –
    protect replication using TSIG signing.
  - **Complexity** – key management, delegation, on-the-fly signing, ...
  - Pay for the service – ask the guy talking.
- I want my queries to **other zones** to enforce DNSSEC–
  deploy DNSSEC-aware **resolvers** on **your side** of the fence.
  - You can use BIND too, but **unbound** is way easier.
  - Enforce DNSSEC on the recursive.
  - Have your devices use that recursive instead.
  - Deploy this on local network – this is the tricky part.

UNIVERSIDAD SIMÓN BOLÍVAR

How about my non-tech savvy loved ones?

. . . 'cause they still think 8.8.8.8 is a good idea

- Do they have an Access Point at home/office?
  - Install FOSS-based firmware like DD-WRT or OpenWRT.
  - Use the embedded resolver supporting DNSSEC.
  - Bonus – pump up the antennas power beyond FCC regulations.
  - Idea – have local events to "upgrade" home AP devices this way.
- DNSSEC Validator – browser plugin, close but no cigar.
- AFAIK iOS/Android don't have DNSSEC support –
  it would be useful to make Unbound run on iOS/Android

UNIVERSIDAD SIMÓN BOLÍVAR

Assorted challenges

- What are Venezuela's ISP-provided resolvers?
- Do they allow DNSSEC-enforced queries?
- Do they give back consistent answers for any given name (DNSSEC-signed or not)?
- Are they faking authority for any given name (of course some are, which ones?).

UNIVERSIDAD SIMÓN BOLÍVAR

Is that enough?

Final thoughts

- DNSSEC is deployed on the assumption that noone wants to shutdown DNS, just manipulate it.
- DNSSEC does not hide information – it prevents tampering.
- ISPs can simply drop your queries – they could block you from using port 53 altogether.
  - Newcomers – DNS over TLS and DNS over HTTPS
  - DNS over TLS relies on port 853 – easily blocked.
  - DNS over HTTPS – browser centric (Danger, Will Robinson!)
- The artesanal solution I'm about to explain, but will not write in the slides to make it hard on the censors.

UNIVERSIDAD SIMÓN BOLÍVAR

There's a LOT to read...

- DNS – RFC-1034 y RFC-1035
- DNS – RFC-4033, RFC-4034 y RFC-4035
- DNSSEC TOOLS – http://www.dnssec-tools.org/
- BIND – Best name server for DNSSEC
  https://www.isc.org/downloads/bind/
  `apt install bind9 bind9-doc bind9utils`
- unbound – Compact recursive server with DNSSEC support
  http://unbound.net
  `apt install unbound`

UNIVERSIDAD SIMÓN BOLÍVAR

How many zones are using DNSSEC?

TLDs are covered, but operators are slow

- ICANN publishes a report on DNSSEC deployment
  http://stats.research.icann.org/dns/tld_report/
- Today 2019-04-13 at 12:07 PST – yes, I was working on slides...
  - 1398 out of 1531 TLDs are signed – 91 %
  - 1387 out of 1398 TLDs hold DS – 99 %
- **Every** new gTLD is signed and holds DSs –
  no wonder: it's mandatory as per ICANN's SLA.
- **Every** classical TLD (`com`, `net`, `org`, ...) is signed and hold DSs –
  if your domain is not being signed, ask your DNS administrator, your Registrar, and ultimately your Registry

`.ve` is not signed, doesn't have provisions for DNSSEC

(and WHOIS been broken for over five years, a different story...)

UNIVERSIDAD SIMÓN BOLÍVAR

Quid quid latine dictum sit, altum videtur.

¡Gracias!

emhn@usb.ve

emhn@uniregistry.link
@iamemhn

UNIVERSIDAD SIMÓN BOLÍVAR