

# **DNS**

# **Name resolution**

# **and beyond**

**Xavier Belanger**

xavier@belanger.fr

June 2024

# This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.

<http://creativecommons.org/licenses/by-sa/4.0/>



## You are free to:

- **Share** — copy and redistribute the material in any medium or format
- **Adapt** — remix, transform, and build upon the material for any purpose, even commercially.

The licensor cannot revoke these freedoms as long as you follow the license terms.

## Under the following terms:

- **Attribution** — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.
- **ShareAlike** — If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.
- **No additional restrictions** — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

# About the speaker

- Working in IT since 1999.
- Professionally managing DNS servers since 2001.
- Currently working as the IT Security Architect at the University of North Carolina Wilmington.

Opinions expressed during this presentation belongs to the author, and do not represent opinions or views of my past or current employers, nor any other organization, group or individual.

*“But the DNS is a lot like chess.*

*It's a simple game in terms of the rules,  
but phenomenally complex in the way  
it can be played.”*

Geoff Huston - [potaroo.net](http://potaroo.net) - March 2021

# Why do we need DNS?

- Computers can talk to each other by using the Internet Protocol (IP), relying on IP addresses.
- Human beings are not very good at working with IP addresses.
- Some servers are used to host many services for different usages, each one needs a different name (the server may still use only one IP address).
- Some services are hosted by more than one server.
- Hence, we need a service to resolve names to IP addresses (and vice-versa): the Domain Name System.

# History

- A long time ago, names and addresses were managed manually by Jon Postel.
- With the expansion of Arpanet, a better system was needed. Postel asked Paul Mockapetris to evaluate five different solutions. Mockapetris came with another idea instead, the Domain Name System.
- DNS became official with RFC 882 and 883 in November 1983; many more have been added since.

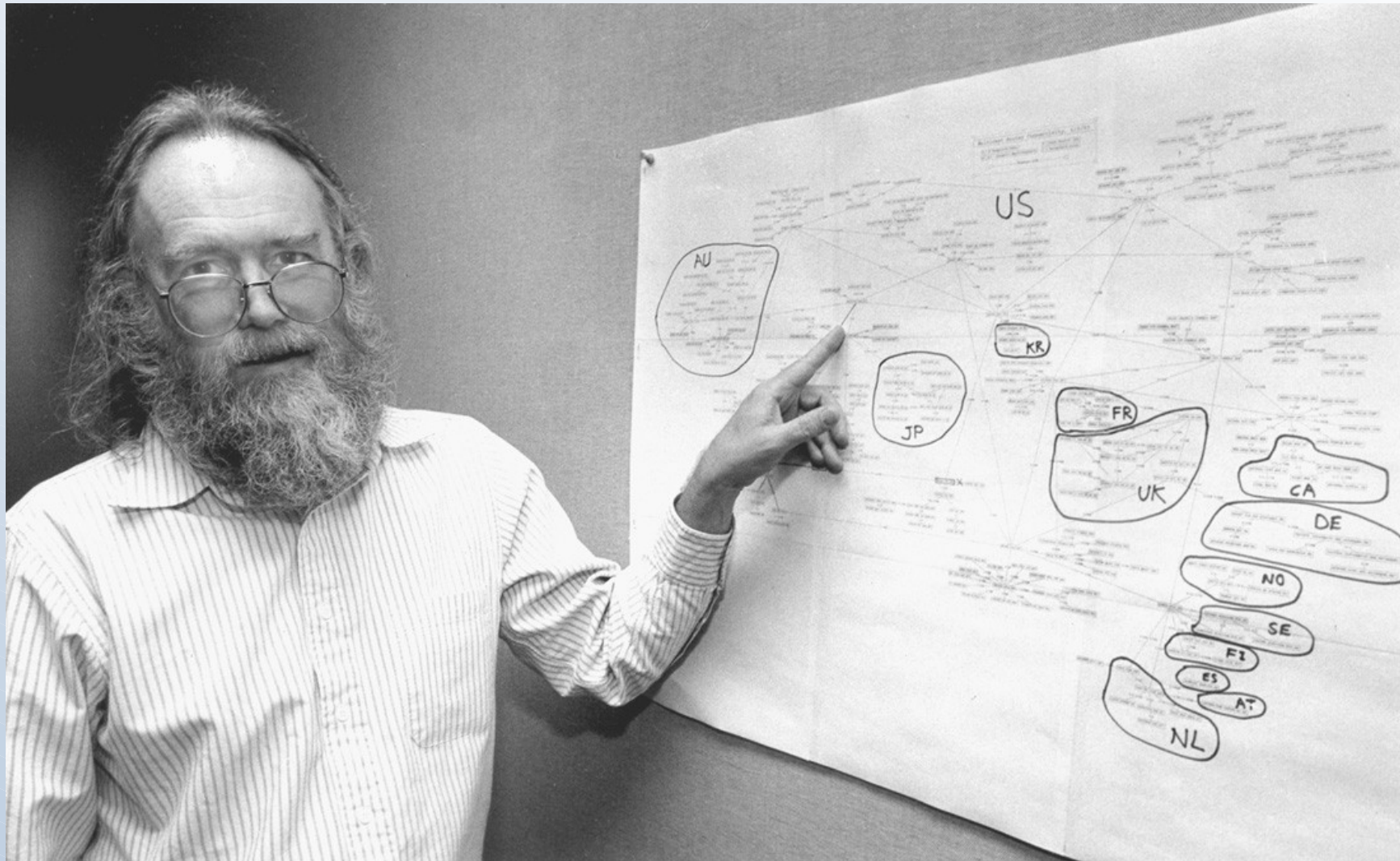
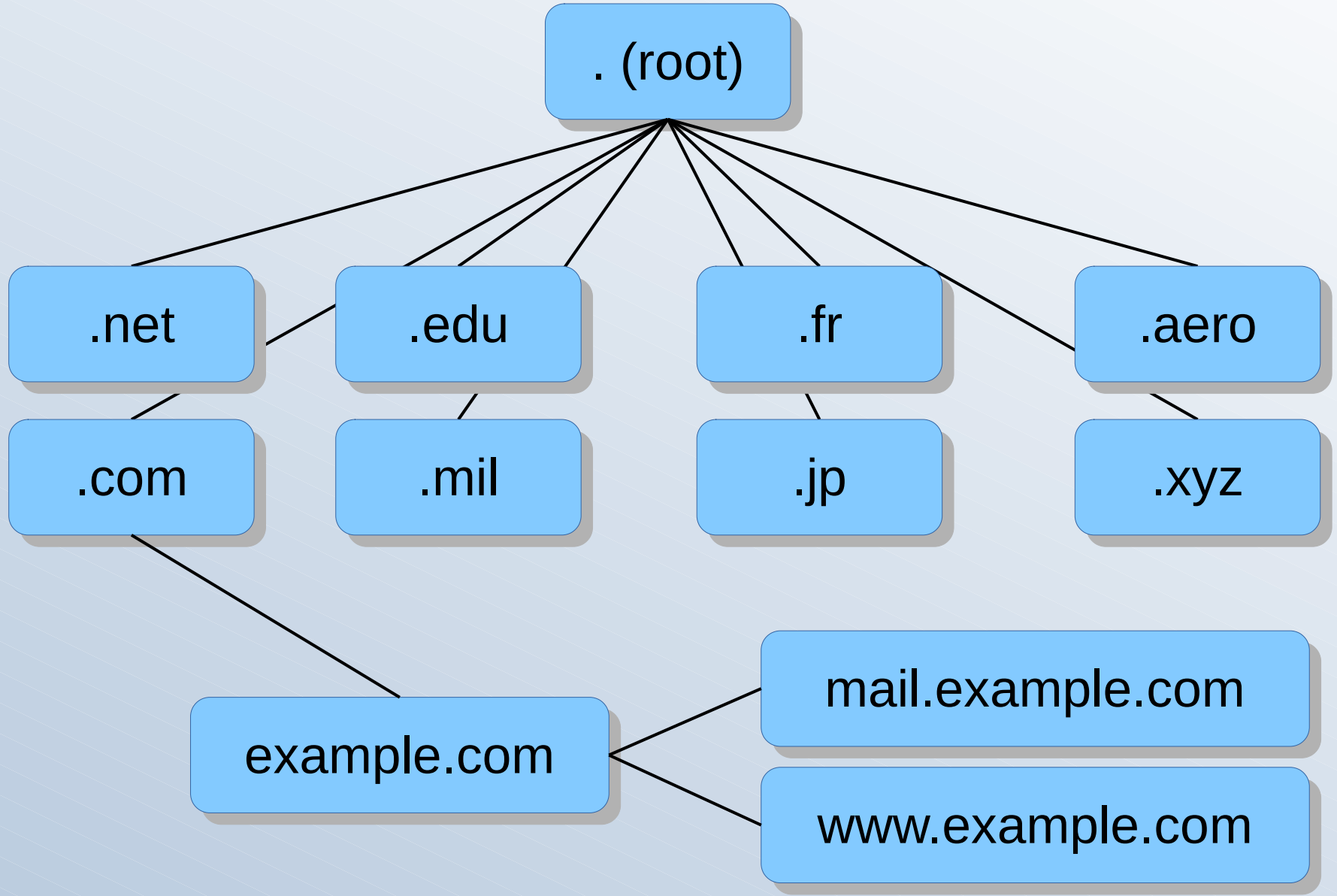


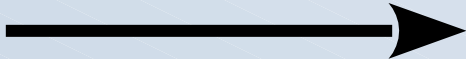
Photo by Irene Fertik, USC News Service. Copyright 1994, USC. Permission granted for free use and distribution, conditioned upon inclusion of the above attribution and copyright notice. - <http://www.postel.org/pr.htm>





# Internationalization (i18n)

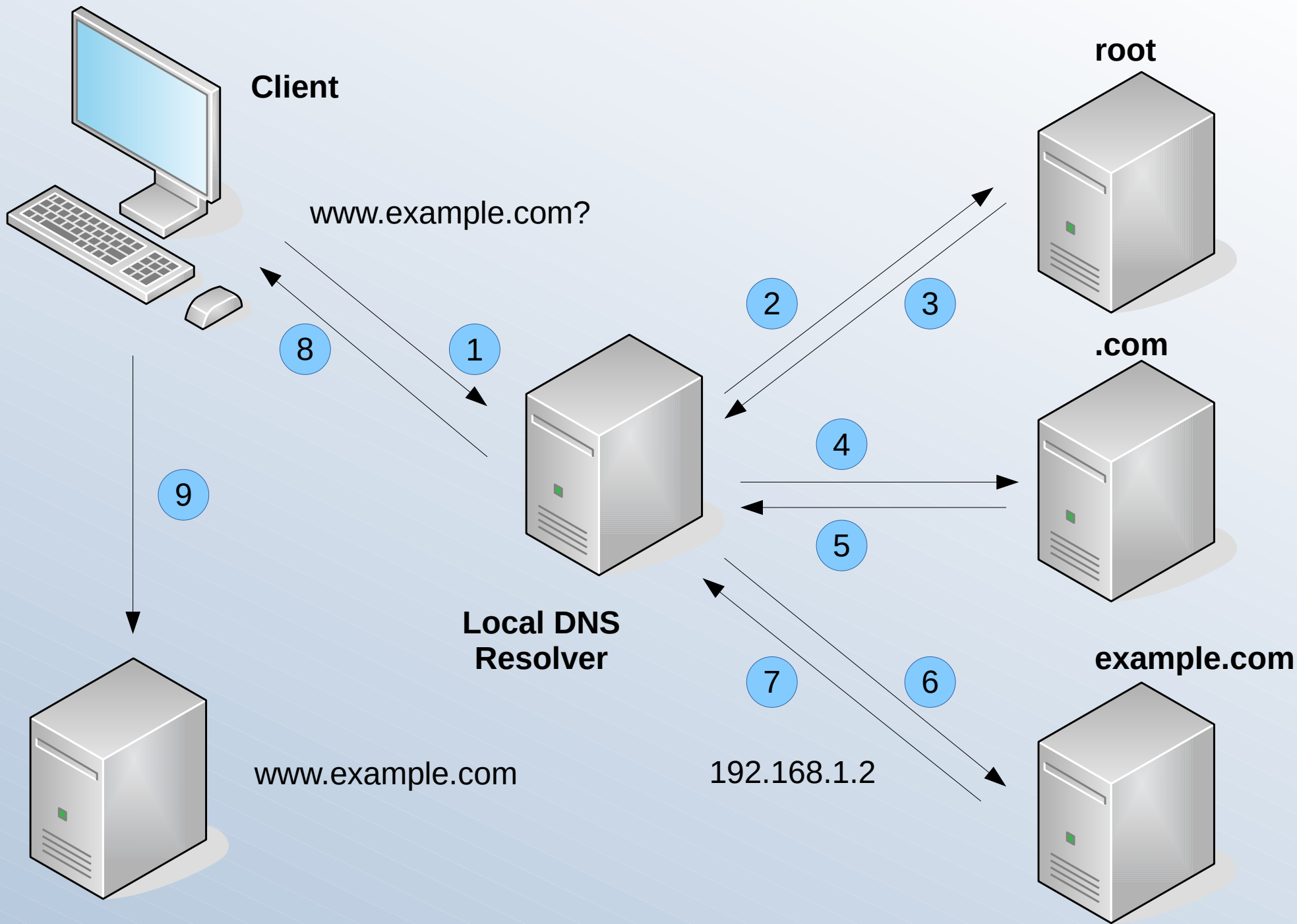
Domains and records are not limited to the ASCII character encoding anymore. Arabic, Chinese, Russian and other alphabets can be used. Punycode is used to “translate” those names into ASCII.

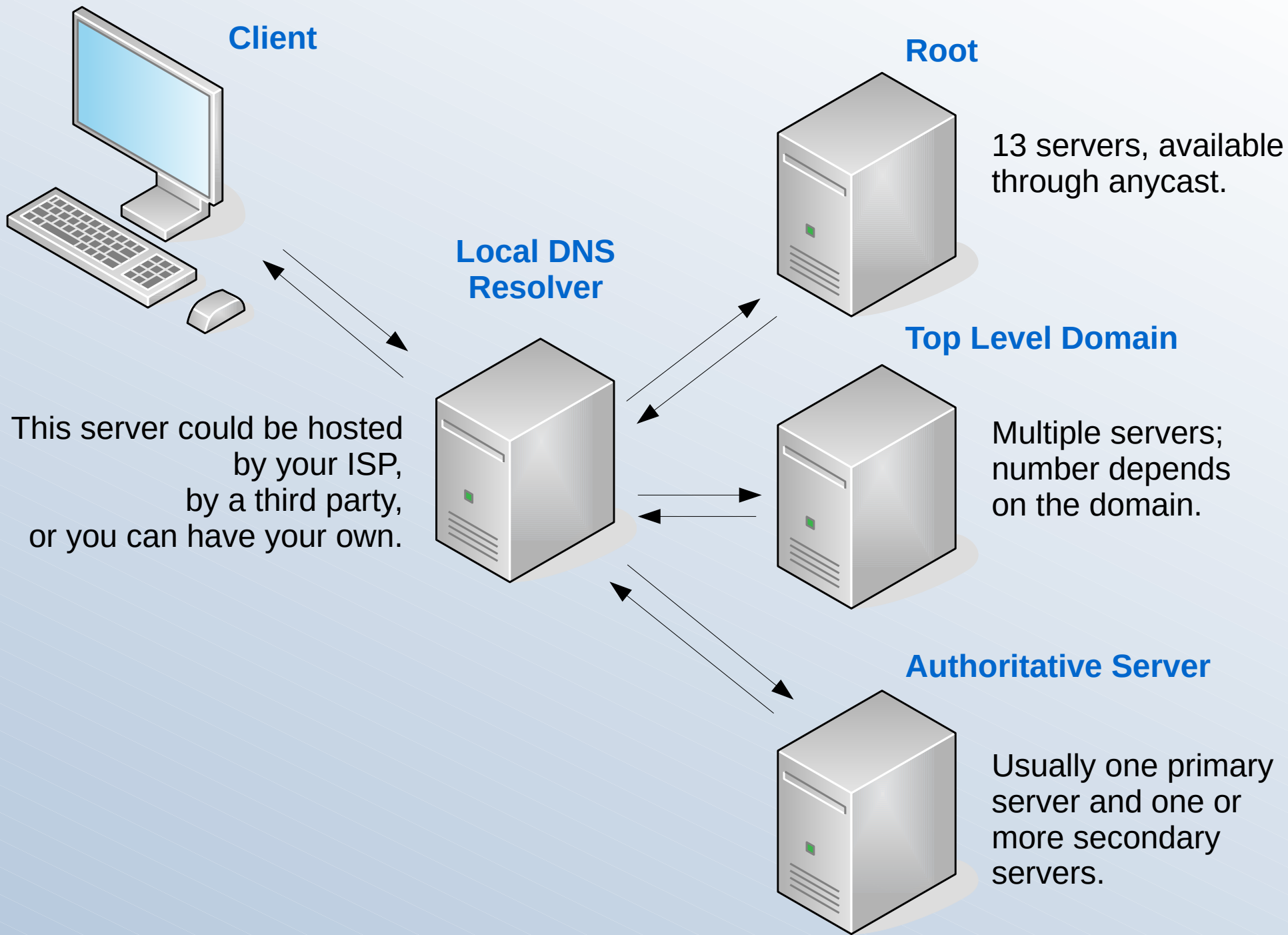
日本語 .jp  xn--wgv71a119e.jp

(Japan Registry Services Co.)

# How does DNS work?

- Your computer (the client) asks the local DNS resolver: *“I need to reach www.example.com do you know its IP address?”*
- If the local resolver doesn't know, it will ask the root. With the response the resolver will ask other servers (authoritative ones), and eventually get a response.
- All of this in few milliseconds on average (heavily relying on caching).
- DNS traffic uses UDP/53 and in some specific circumstances TCP/53.





**Client**

**Root**

13 servers, available through anycast.

**Local DNS Resolver**

This server could be hosted by your ISP, by a third party, or you can have your own.

**Top Level Domain**

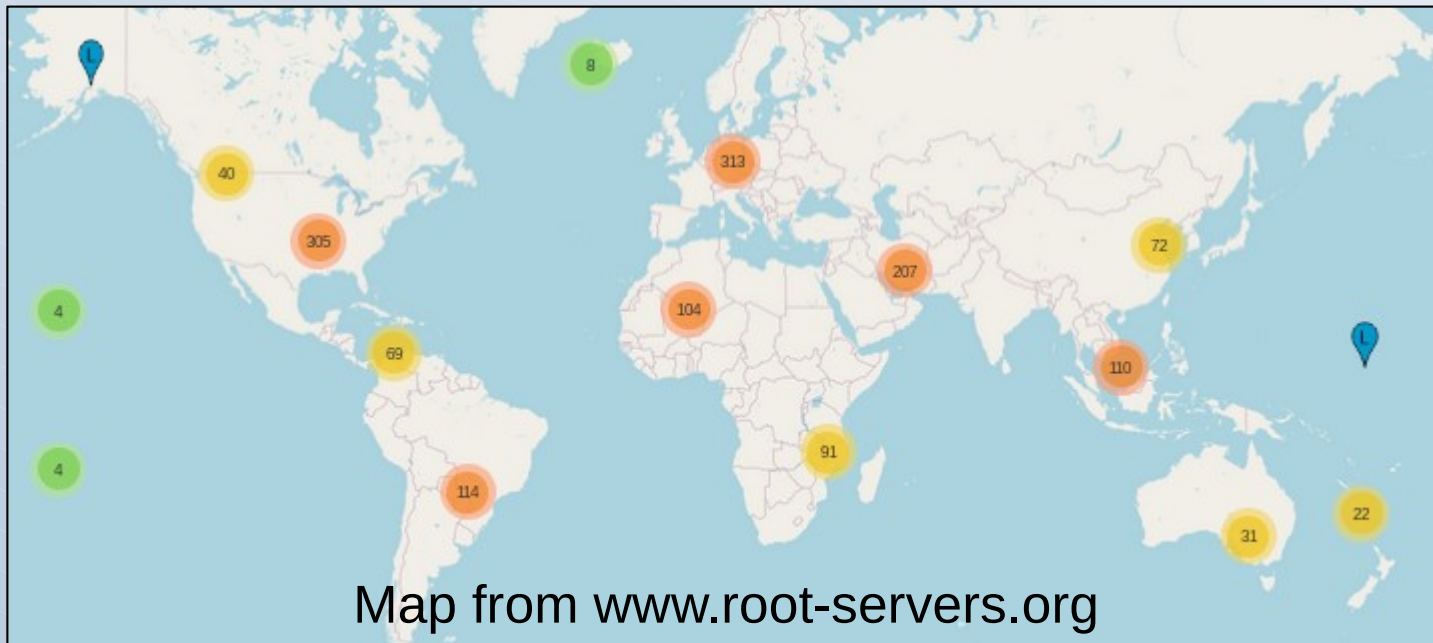
Multiple servers; number depends on the domain.

**Authoritative Server**

Usually one primary server and one or more secondary servers.

# The Root Servers

- There is 13 root name servers are operated by 12 organizations (Verisign, Cogent, NASA, University of Maryland, ICANN, DoD, ISC, NetNod,...)
- Multiple instances (~1,800), spread across the globe, using anycast.



# DNS Resource Record Types

- SOA and NS
- A, AAAA, PTR and CNAME
- MX, TXT
- And many, many more. The full list is managed by IANA (Internet Assigned Numbers Authority).

# Essential Records

- The Start of Authority (SOA) is the essential record to define a zone with the name of the domain itself, the primary name server, a contact email address and various time values (in seconds) used by the other servers:
  - refresh
  - retry
  - expire
  - negative caching
- Name Server (NS) records will list all the DNS servers for a zone.

```
$ dig SOA example.com
```

```
example.com. 86400 IN SOA ns1.example.com.  
hostmaster.example.com. (  
2023021501 ; serial  
86400      ; refresh (1 day)  
3600      ; retry (1 hour)  
2419200   ; expire (4 weeks)  
86400     ; minimum (1 day)  
)
```



```
$ dig NS example.com
```

```
example.com.      86400 IN NS ns1.example.com.
```

```
example.com.      86400 IN NS ns2.example.com.
```

```
ns1.example.com.  86400 IN A 10.1.0.10
```

```
ns2.example.com.  86400 IN A 10.2.0.30
```

# Hosts and Addresses Records

- The hostname for an IPv4 address is registered with an 'A' record; for an IPv6 address, an 'AAAA' record is used.
- The pointer from an IP address to a hostname is handled by a PTR record (*in-addr.arpa* and *ip6.arpa* special domains).
- CNAME records are used to create aliases for a hostname.
- If Dynamic DNS (DDNS) is used, the DHCP server can register an host hostname on its behalf.

```
$ dig A www.example.com
```

```
www.example.com.      86400 IN CNAME  web01.example.com.  
web01.example.com. 86400 IN A      10.5.4.47
```

```
$ dig -x 10.5.4.47
```

```
47.4.5.10.in-addr.arpa. 86400 IN PTR web01.example.com.
```

```
$ dig AAAA www.example.com
```

```
www.example.com.      86400 IN CNAME  web01.example.com.  
web01.example.com. 86400 IN AAAA   2001:880:281:8::1
```

# Email Records

- Servers accepting e-mails for a domain must be listed as mail exchangers in the DNS with MX records.
- Those records also provides a weight for each server; this is the priority used for e-mail delivery (low value = high priority).
- TXT records are text based. Some of those are set with a specific format to store information used to detect spam (*Sender Policy Framework - SPF, DomainKeys Identified Mail - DKIM, Domain-based Message Authentication, Reporting and Conformance - DMARC*).

```
$ dig MX example.com
```

```
example.com.      86400 IN    MX     50    mx01.example.com.  
example.com.      86400 IN    MX    100    mx02.example.com.
```

```
$ dig TXT example.com
```

```
example.com.      86400 IN    TXT  
"v=spf1 include:smtp.example.com ~all"
```

# Tools and Troubleshooting

- nslookup, dig, host, drill
- Server logs, dnstap
- Third party tools: zonemaster.net, intodns.com, mxtoolbox.com, dnsdumpster.com, ...
- Public DNS servers:  
Google Public DNS, Verisign Public DNS,  
Cisco OpenDNS/Umbrella, Cloudflare,  
Quad9,...

# DNS and OSINT

- All data published in DNS is public, and can be used as open source intelligence (OSINT).
- By checking on DNS records you can get a sense how an organization is structured, what resources are exposed, what services are used.

# Limits and Issues

- Clients can be redirected to a rogue DNS server and get wrong or malicious answers.
- One can try to poison a DNS server, and then affect all the clients.
- DNS can be used for DoS/DDoS attacks.
- Servers operators can see all DNS queries and trace network activity; they can also block access to some domain names.
- 2008: Kaminsky Attack (DNS poisoning)
- 2015: NSA MORECOWBELL



# DNS and Malware

- Since malware relies on DNS, it is possible to trace malware activity by checking DNS traffic (new domain names, generated names).
- Some fraudulent activities (phishing, malware delivery) are using DNS fast-flux to hide the original source and to improve reliability.
- DNS traffic can be used to exfiltrate data.

# Domain Generation Algorithms

- In order to reach a command and control (C&C) rendezvous point, some malwares are using randomly generated names, looking like gibberish. This technique is known as “Domain Generation Algorithms” (DGA).
- DGA names are generally used in large numbers and difficult to track and block.

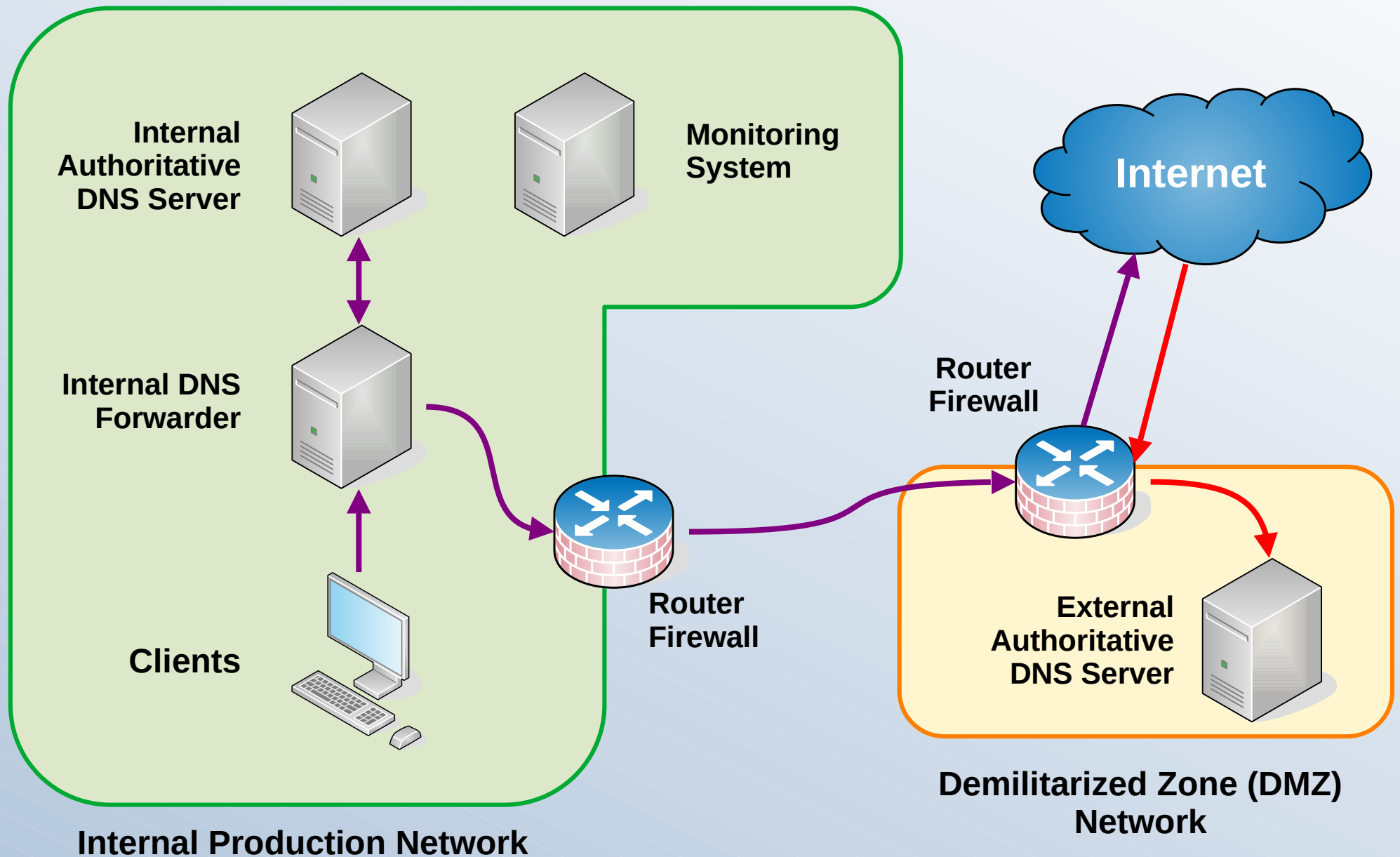
# DNS Exfil

- DNS queries can be used to exfiltrate data.
- Data need be converted into valid characters to build a DNS query (base64, hexadecimal).
- Regular size limits for queries still apply, but if no monitoring and blocking is in place, the malicious traffic could be running for a long time.

# Some Solutions

- Proper server configuration and management.
- Separate authoritative servers and resolvers.
- DNS filtering (inbound and outbound).
- DNS monitoring (server level, network level).
- Response Rate Limiting.
- Zone Transfer Integrity (TSIG).
- Keep your servers and clients up-to-date.

# DNS Architecture Example



# DNSSEC

- DNSSEC is an extension of the DNS protocol, providing cryptographic tools to validate the information from a signed zone.
- Defined in multiple RFCs, summarized in RFC 9364.
- Records are signed by the server operator with a private key; clients can check the records validity by using a public key.
- DNSSEC require more dedicated administration, and automation.

# DNSSEC Keys

- The root provide a special key, the DNS Root trust anchor, that is used to validate other keys.
- A signed zone will use two sets of keys:
  - a Zone Signing Key (ZKS), used to sign the data within the zone.
  - a Key Signing Key (KSK), used to sign the ZKS; used as a secure entry point for the zone.
- Keys should be rotated on a regular basis (key rollover).

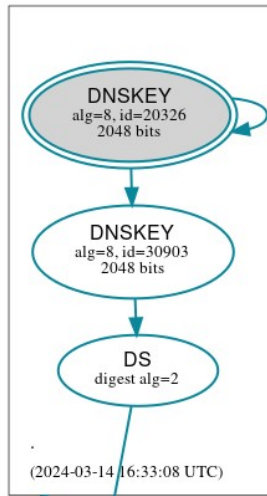
```
$ dig +dnssec SOA example.com
```

```
example.com.      86400 IN SOA ns1.example.com. hostmaster.example.com. (
2023021501 ; serial
86400      ; refresh (1 day)
3600       ; retry (1 hour)
2419200    ; expire (4 weeks)
86400      ; minimum (1 day)
)
```

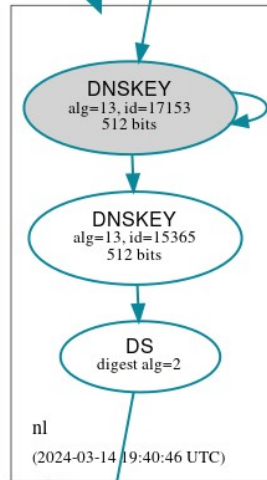
```
example.com.      86400 IN RRSIG SOA 7 2 86400 (
20230615141724 20230116141724 4502 example.com.
VCyDbhpxWgF5eRHRQt9o4fFtuN4PEzNvnUs+VRnOr+us
0zl/de9NqFxcbwP7HMuhJhyJr fFBMnFUFJq6ye98Dywc
exaqEmpdc8KL8PlllFQwB/jUe6LPQpHKAx10HUdqyOUT
ci2+qCWlc85oTwGixh4FoBuFe3lIf//Vzx/100E= )
```



root



.nl



nlnetlabs.nl

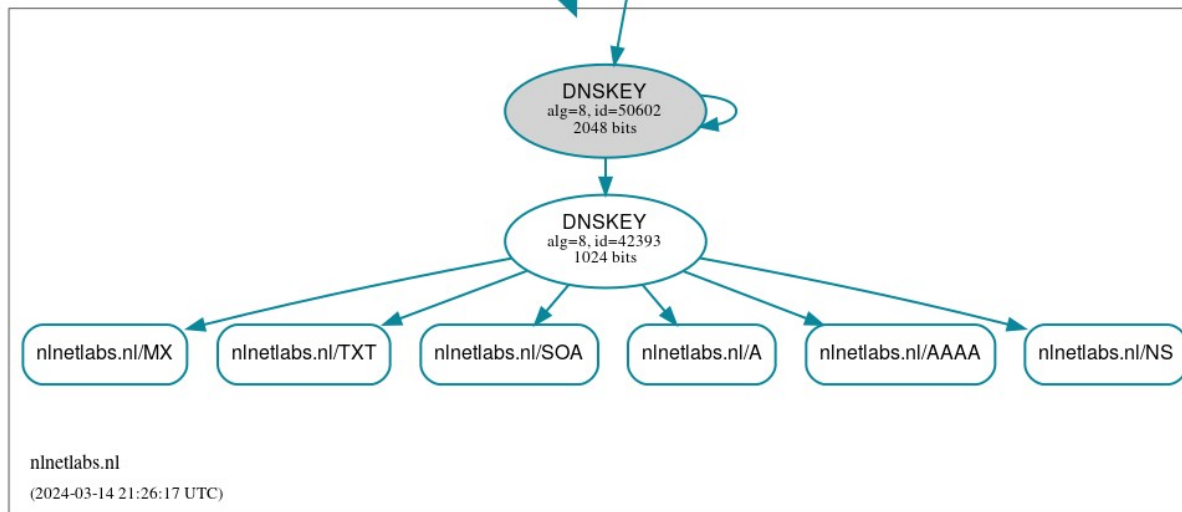


Diagram from  
dnsviz.net

# SSHFP Record

A SSHFP record stores the fingerprint of a SSH server; this can be used to check the validity of that fingerprint at the first connection attempt.

```
$ ssh -o "VerifyHostKeyDNS ask" server.example.com
```

```
The authenticity of host 'server.example.com (192.168.1.2)'  
can't be established.
```

```
RSA key fingerprint is
```

```
e9:8b:c4:5b:8f:bc:98:07:5f:20:ff:c4:23:7f:cb:aa.
```

```
Matching host key fingerprint found in DNS.
```

```
Are you sure you want to continue connecting (yes/no)?
```

# DANE

- DNS-based Authentication of Named Entities.
- Now you can provide security certificates (X.509, used with TLS) with DNS. This can be used to secure websites and email servers.
- Different options are possible, from confirming the information on a Certificate Authority (CA) to providing a full certificate.
- DANE doesn't have a widespread support yet, and require DNSSEC.

# CAA Record

- Certification Authority Authorization, defined by RFC 6844 (updated with RFC 8659 in 2019).
- That record list the the Certification Authorities authorized to issue security certificates for a domain.
- This doesn't involve or apply to TLS, it is mostly used to certificate issuing.
- Following a decision of the CA/Browser Forum, all certificate authorities must check CAA records starting September 2017.

# DNS over TLS

- RFC 7858, published in May 2016.
- It is a secure version of DNS, providing confidentiality between the client and the server.
- This is not an “upgrade” of the regular DNS traffic: the server and the client should both be configured to use TLS.
- A specific port is defined: TCP/853.

# DNS over HTTPS

- RFC 8484, published in October 2018.
- It's a tunneling technique to use HTTPS to encapsulate DNS traffic.
- The goal is to provide confidentiality between a client and a recursive server.
- Like for DNS over TLS, the server and the client should be configured for DoH.

# Software

- ISC BIND
- Unbound
- Microsoft DNS
- and others (Knot DNS, PowerDNS, ...).

# BIND

- Berkeley Internet Name Domain
- Developed by the Internet Software Consortium (ISC).
- Could be used as a cache and recursive server or as an authoritative server.
- Configuration file: `named.conf`, plus the zone files.
- Management tool: `rndc`
- Additional tools: `named-checkconf`, `named-checkzone`



# Unbound

- Developed by NLnet Labs (in the Netherlands).
- Could be used as a cache, recursive server.
- Configuration file: `unbound.conf`
- Management tool: `unbound-control`
- Additional tool: `unbound-checkconf`

# Microsoft DNS

- Available as a role on MS Windows Server.
- Active Directory can be used as a backend, instead of regular zone files.
- Could be used as a cache and recursive server or as an authoritative server.
- Can be managed via the GUI (DNS Manager) and via the command line (dnscmd, various PowerShell cmdlets).

# Around DNS

- **mDNS**: multicast DNS. RFC 6762, used for stand-alone networks, using TCP/5353. Also known as Avahi or Bonjour.
- **Alternative roots or systems**: namecoin, OpenNIC, .onion, Distributed Hash Tables (DHT), ...

# Getting Your Own Domain Name

- Check for the desired name availability (think about alternate names as secondary choices).
- Create an account with the Domain Name Registrar of your choice (if possible).
- Once the domain is activated, create the appropriate DNS records for your servers.
- Do not forget to renew your domain registration in advance.

# Running Your Own Internal Zone

- By using an authoritative DNS server on your home network you can use your own private zone.
- *home.arpa* is the special name defined for this purpose (RFC 8375).
- Depending on the application that you're using and what level of complexity you're looking for, you may need to run an authoritative server and a resolver server.

# Official Resources

- **Internet Corporation For Assigned Names and Numbers**  
<https://www.icann.org/>
- **Internet Assigned Numbers Authority**  
<https://www.iana.org/>
- **Root Servers**  
<http://www.root-servers.org/>
- **RFCs**  
<https://powerdns.org/dns-camel/>

# Third Party Resources

- **DNS for Rocket Scientists**  
<https://www.zytrax.com/books/dns>
- **Calomel**  
<https://calomel.org/>
- **Team Cymru**  
<https://www.team-cymru.org/>
- **Google Public DNS Servers**  
<https://developers.google.com/speed/public-dns>
- **Google Apps Toolbox**  
<https://toolbox.googleapps.com/apps/dig>

# Books

- **DNS and BIND**, 5<sup>th</sup> ed.  
Cricket Liu and Paul Albitz - O'Reilly
- **DNS and Bind Cookbook**  
Cricket Liu - O'Reilly
- **Pro DNS and BIND**  
Ron Aitchison - Apress
- **DNS Security**  
Allan Liska and Geoffrey Stowe - Syngress
- **DNS Security Management**  
Michael Dooley and Timothy Rooney - IEEE Press / Wiley
- **Managing Mission-Critical Domains and DNS**  
Mark E. Jeftovic - Packt
- **DNSSEC Mastery**  
Michael W. Lucas - Tilted Windmill Press
- **Learning CoreDNS**  
John Belamaric and Cricket Liu - O'Reilly



# Videos

- **DNS Explained (DNS Made Easy)**  
<https://www.youtube.com/watch?v=72snZctFFtA>
- **Threat Hunting via DNS (E. Conrad - SANS)**  
<https://www.youtube.com/watch?v=RdcCjDS0s6s>
- **How DNS can be abused for Command & Control (T. Wojewoda - BHIS)**  
<https://www.youtube.com/watch?v=u2VAInoe9eM>
- **DNSSEC (M. W. Lucas)**  
<https://www.youtube.com/watch?v=0Fc8ZMIspBc>

**Thank you**

**Time for questions  
and discussion**

- Presentation created with LibreOffice 7.  
<https://www.libreoffice.org/>
- Network icons/shapes from VRT Systems.  
<https://www.vrt.com.au/downloads/vrt-network-equipment>