DNSSEC Tutorial [1]

Frederico Augusto de Carvalho Neves
<tutorial-dnssec@registro.br>

Registro.br

May 17, 2010

[1] version 1.5.2 (Revision: 5662)
The last version of this tutorial can be found at: ftp://ftp.registro.br/pub/doc/tutorial-dnssec.pdf

Objectives

- Introduction to DNSSEC concepts
- Show practical examples of DNSSEC usage with BIND
- Encourage the use of DNSSEC

Agenda

1. DNS
   - Concepts
   - Architecture
   - Vulnerabilities
2. DNSSEC
   - Concepts
   - Resource Records
   - Operation
   - Keys
   - DNS vs DNSSEC
   - Softwares
3. Development
   - Configurations
   - DNSSEC with BIND
   - Validation Test
   - Practical summary
4. References

Part I

DNS and DNSSEC concepts

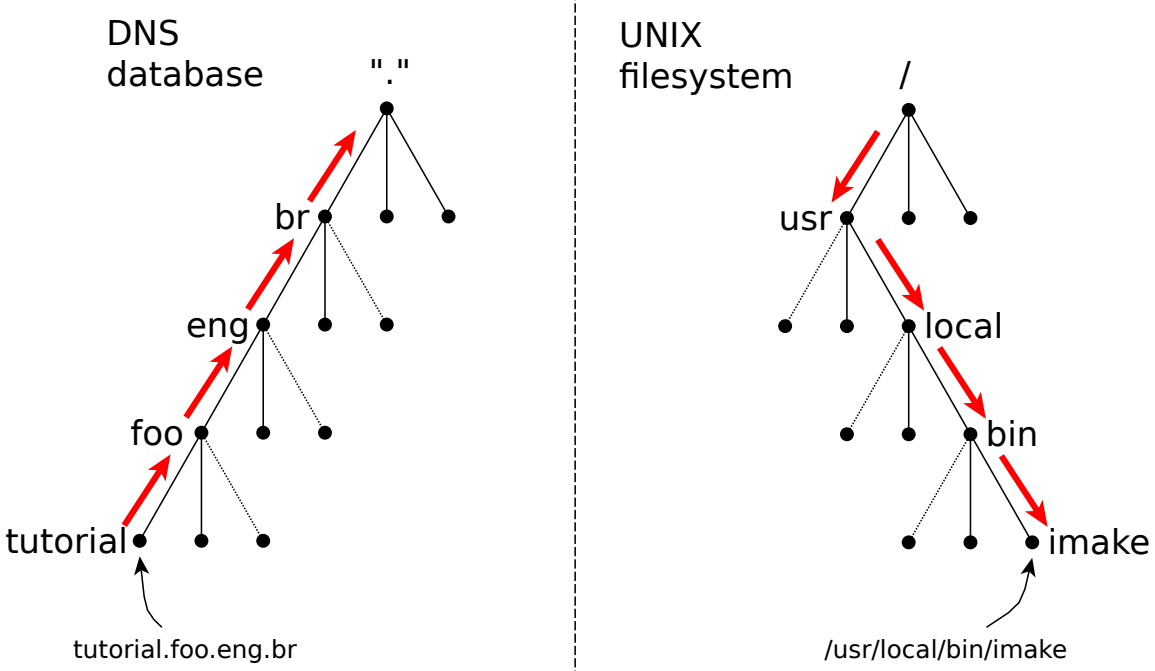
DNS - Domain Name System

The domain name system is a distributed database. This allows for local control on the segments of the global database, although the data in each segment are available to all the network through the client-server schema.

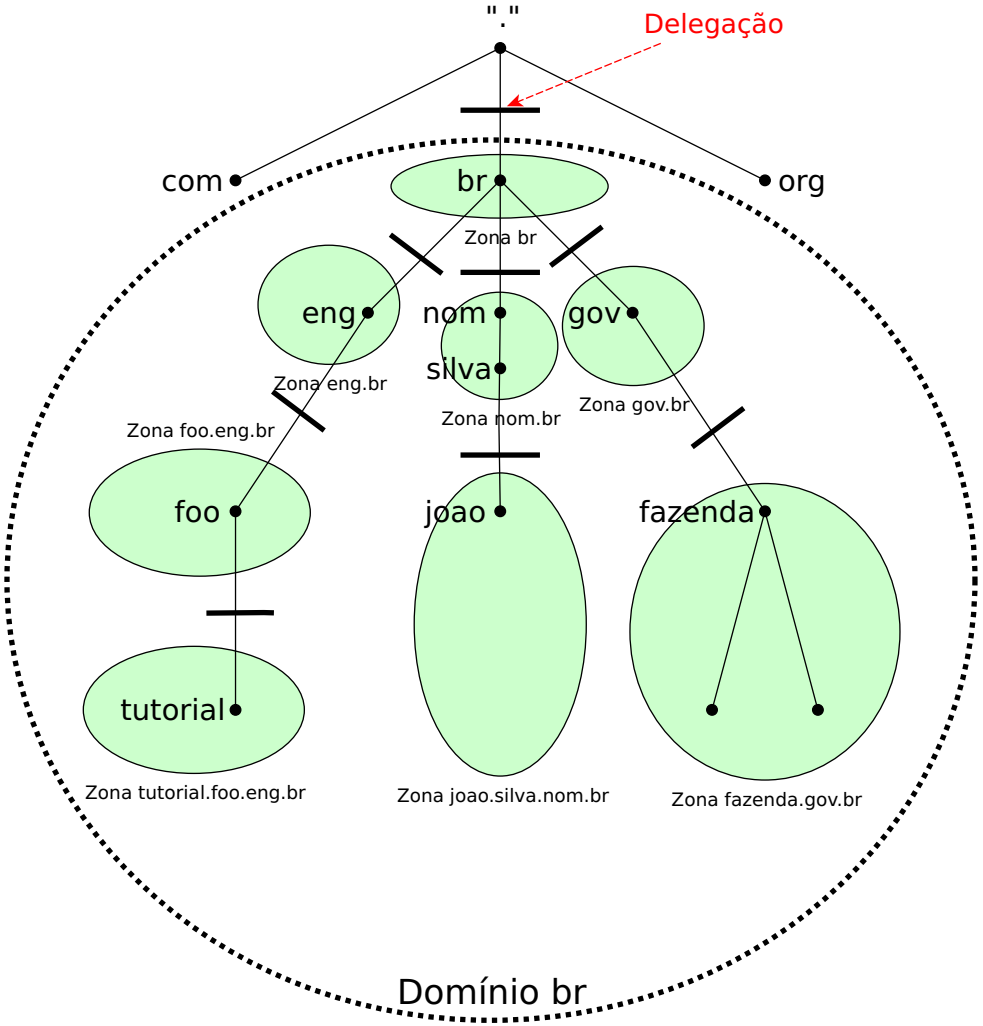
- Hierarchical architecture, data arranged in an inverted tree
- Efficiently distributed, descentralized system and with caching
- The main purpose is to resolve domain names into addresses IP and vice-versa

exemplo.foo.eng.br	$\longleftrightarrow$	200.160.10.251
www.cgi.br	$\longleftrightarrow$	200.160.4.2
www.registro.br	$\longleftrightarrow$	2001:12ff:0:2::3

Hierarchy


DNS
database
"."
br
eng
foo
tutorial
tutorial.foo.eng.br
UNIX
filesystem
/
usr
local
bin
imake
/usr/local/bin/imake

Domain Vs Zone



Delegation

Indicates a transfer of responsibility in the administration of that starting point in the DNS tree

Resource Records

The data associated with the domain names are contained in **Resource Records** or **RR**s

- They are divided into classes and types
- Currently there is a wide variety of types
- The set of resource records with the same domain name, class and type is called **RRset**

Common Resource Records

SOA	Indicates where the zone *authority* begins
NS	Indicates a *name server* for the zone
A	Name to address mapping (IPv4)
AAAA	Name to address mapping (IPv6)
MX	Indicates a *mail exchanger* for a name (e-mail server)
CNAME	Alternative name mapping (nickname)

Zone resource records example

Zone file - Has the RRs for a particular domain, where each domain has a zone file.

```
foo.eng.br. IN SOA ns1.foo.eng.br. hostmaster.foo.eng.br. (
            1        ; serial
            3600     ; refresh
            3600     ; retry
            3600     ; expire
            900 )    ; minimum
foo.eng.br.                IN NS ns1.foo.eng.br.
foo.eng.br.                IN NS ns2.foo.eng.br.
tutorial.foo.eng.br.       IN NS ns1.tutorial.foo.eng.br
tutorial.foo.eng.br.       IN NS ns2.tutorial.foo.eng.br
ns1.foo.eng.br.            IN A 200.160.3.97
ns2.foo.eng.br.            IN A 200.160.10.251
ns1.tutorial.foo.eng.br.   IN A 200.160.3.97
ns2.tutorial.foo.eng.br.   IN A 200.160.10.251
exemplo.foo.eng.br.        IN A 200.160.10.251
```

Server Types

registro.br
Registro de Domínios para a Internet no Brasil

Recursive Server

Upon receiving requests for name resolution, makes requests to the authoritative servers and, depending on the response received, continues to make requests to other authoritative servers until it gets a satisfactory answer

Authoritative Server

Upon receiving requests for name resolution, answers with the IP address if it is authoritative for the queried name or with a referral if the name can be found in their delegations or with a negative response if the server doesn't know anything about the queried name.

Address request example

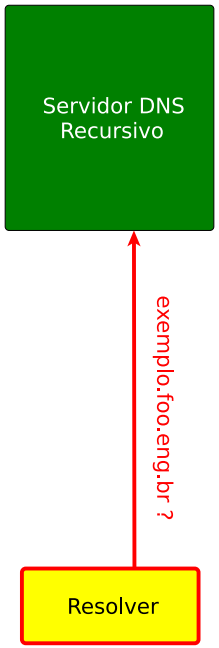
Cache is empty or without br, eng.br, foo.eng.br, exemplo.foo.eng.br information

Resolver

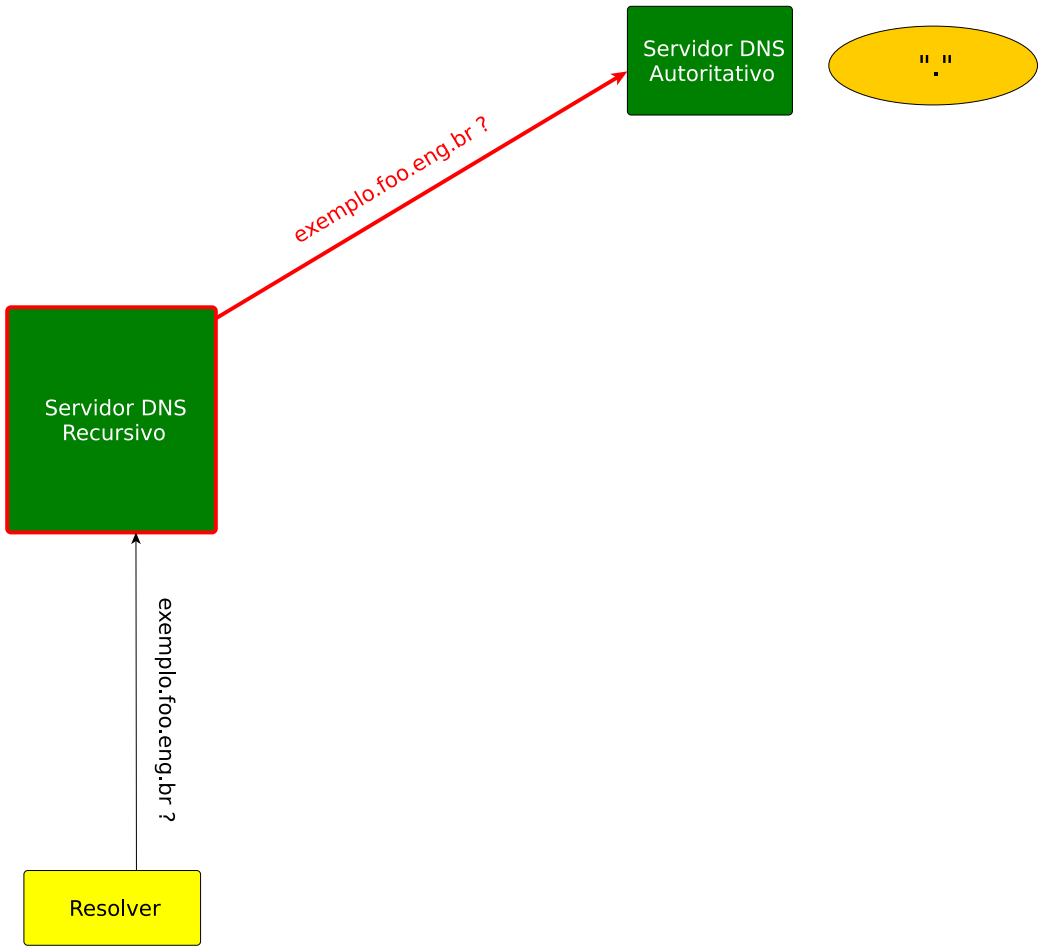
Client application that handles DNS name resolution for various other applications

Address request example

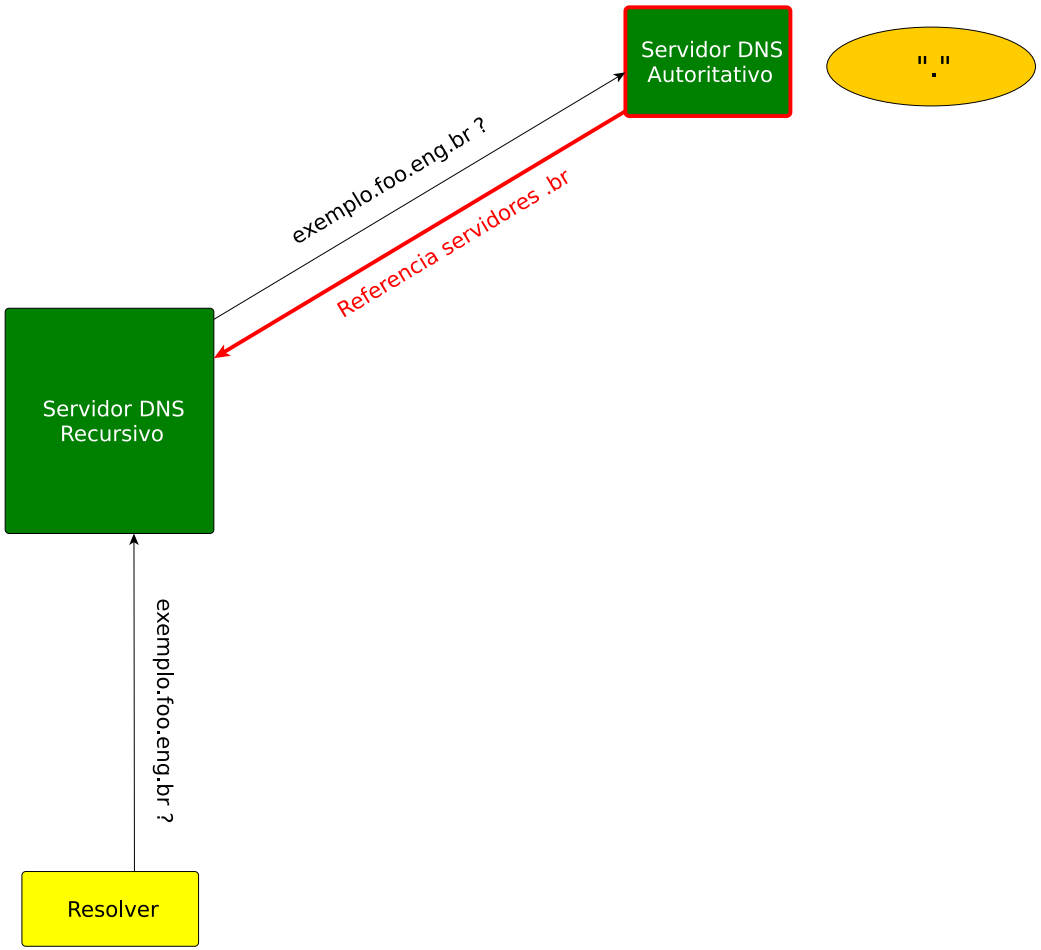
Cache is empty or without br, eng.br, foo.eng.br, exemplo.foo.eng.br information



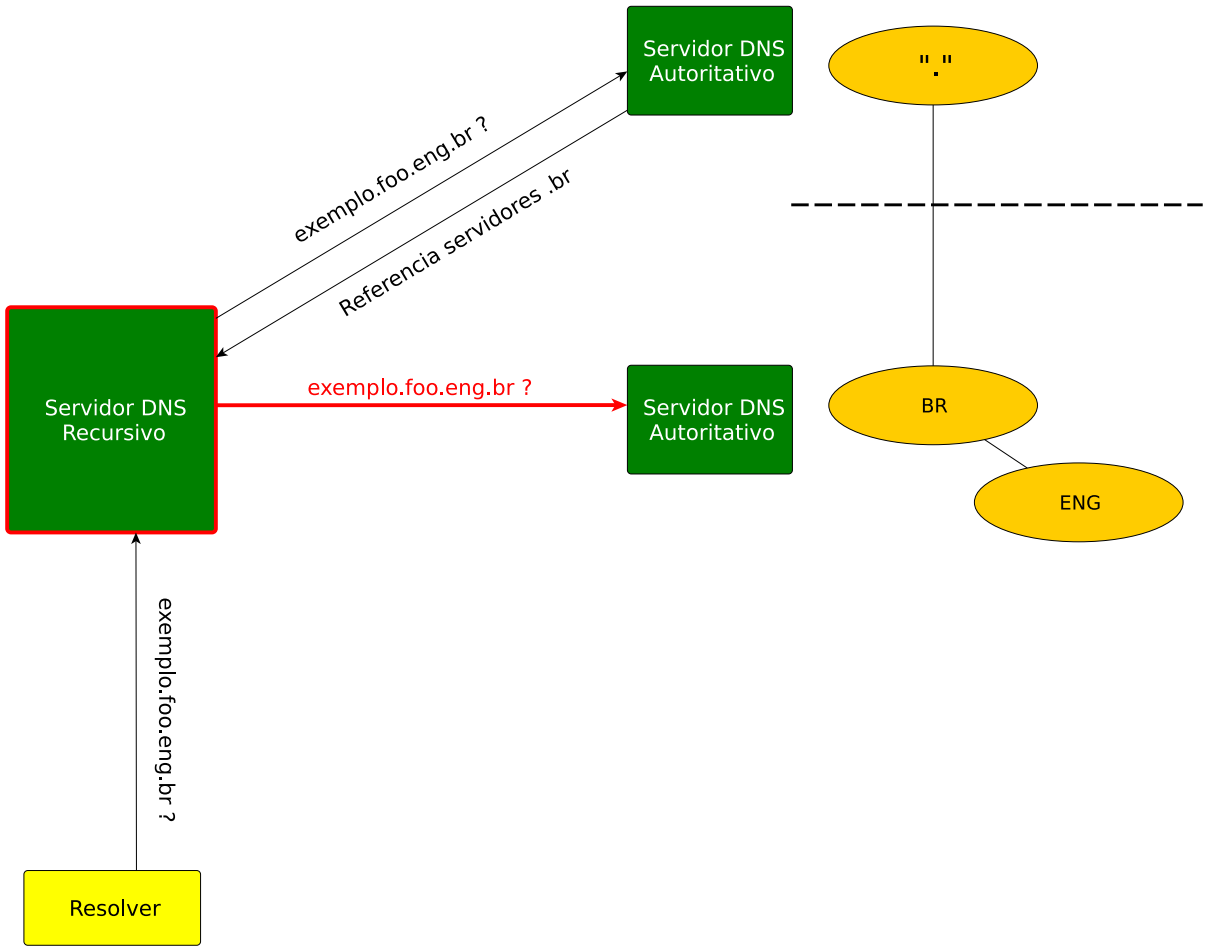
Address request example



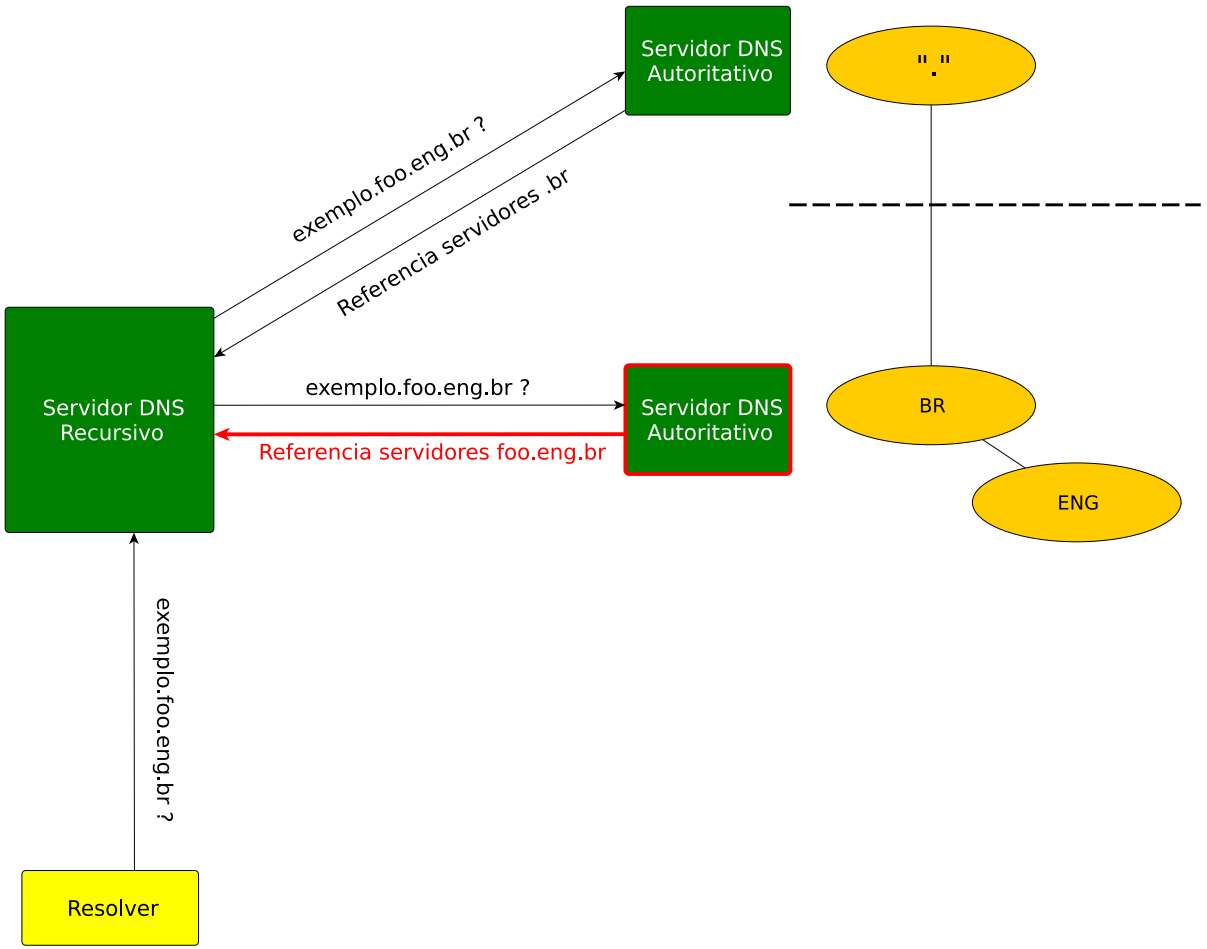
Address request example



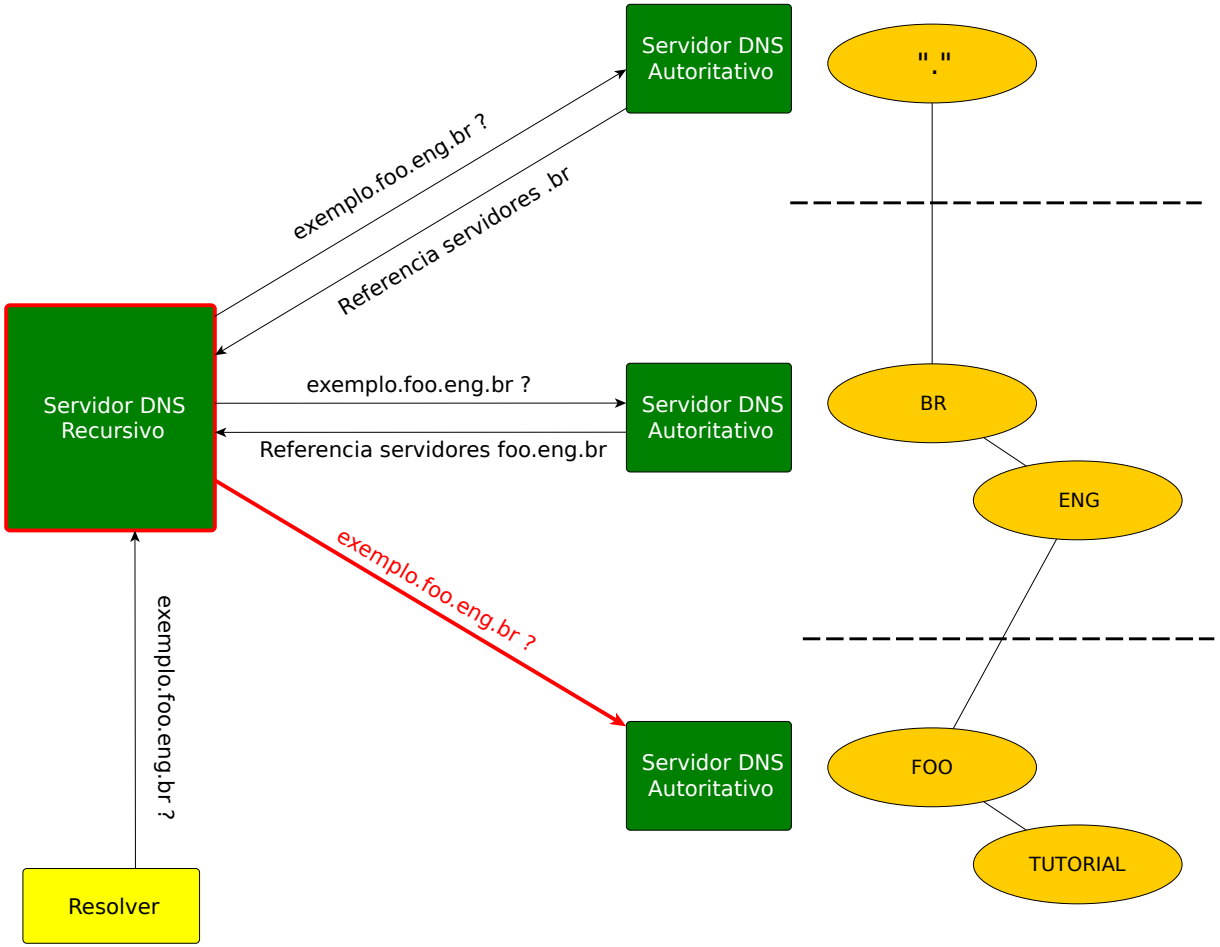
Address request example



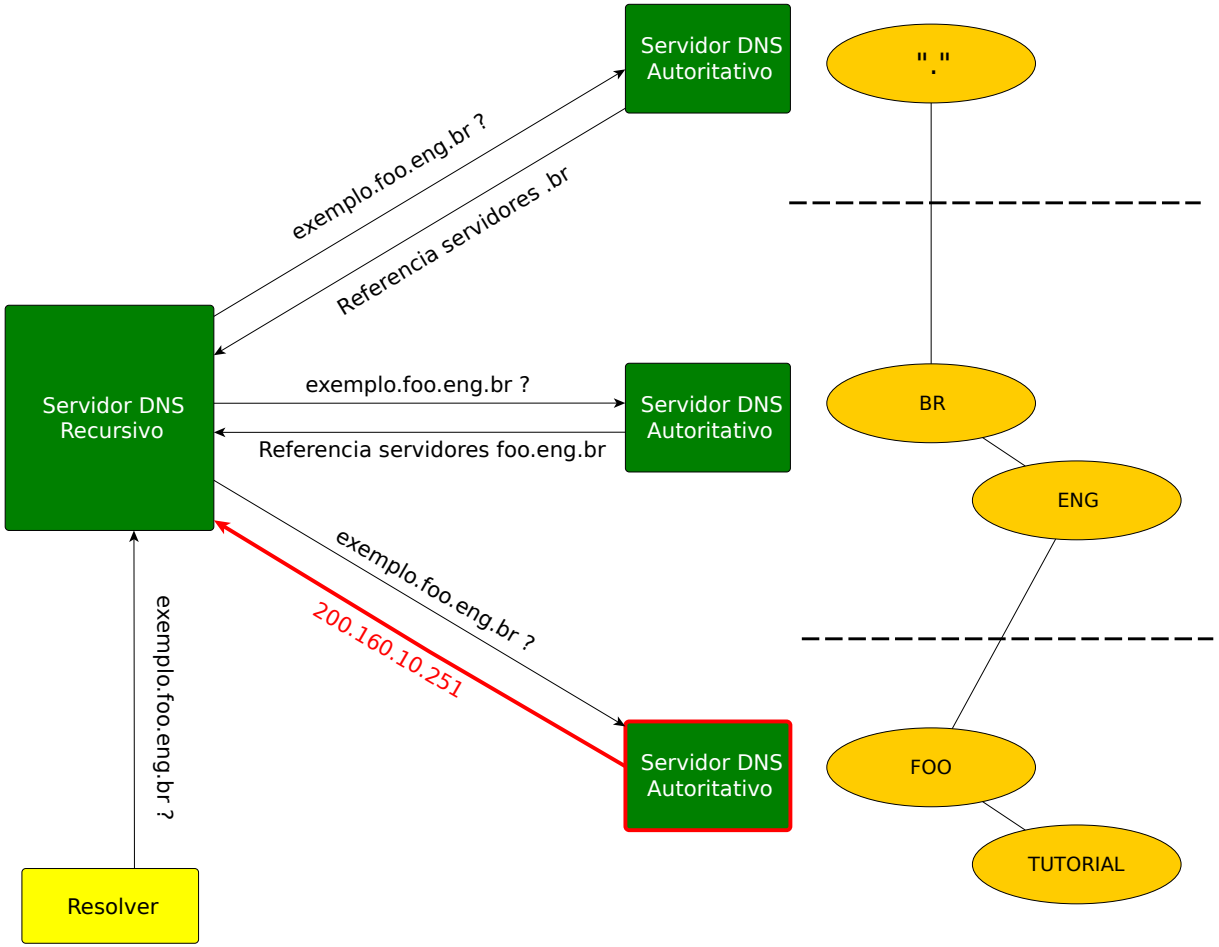
Address request example



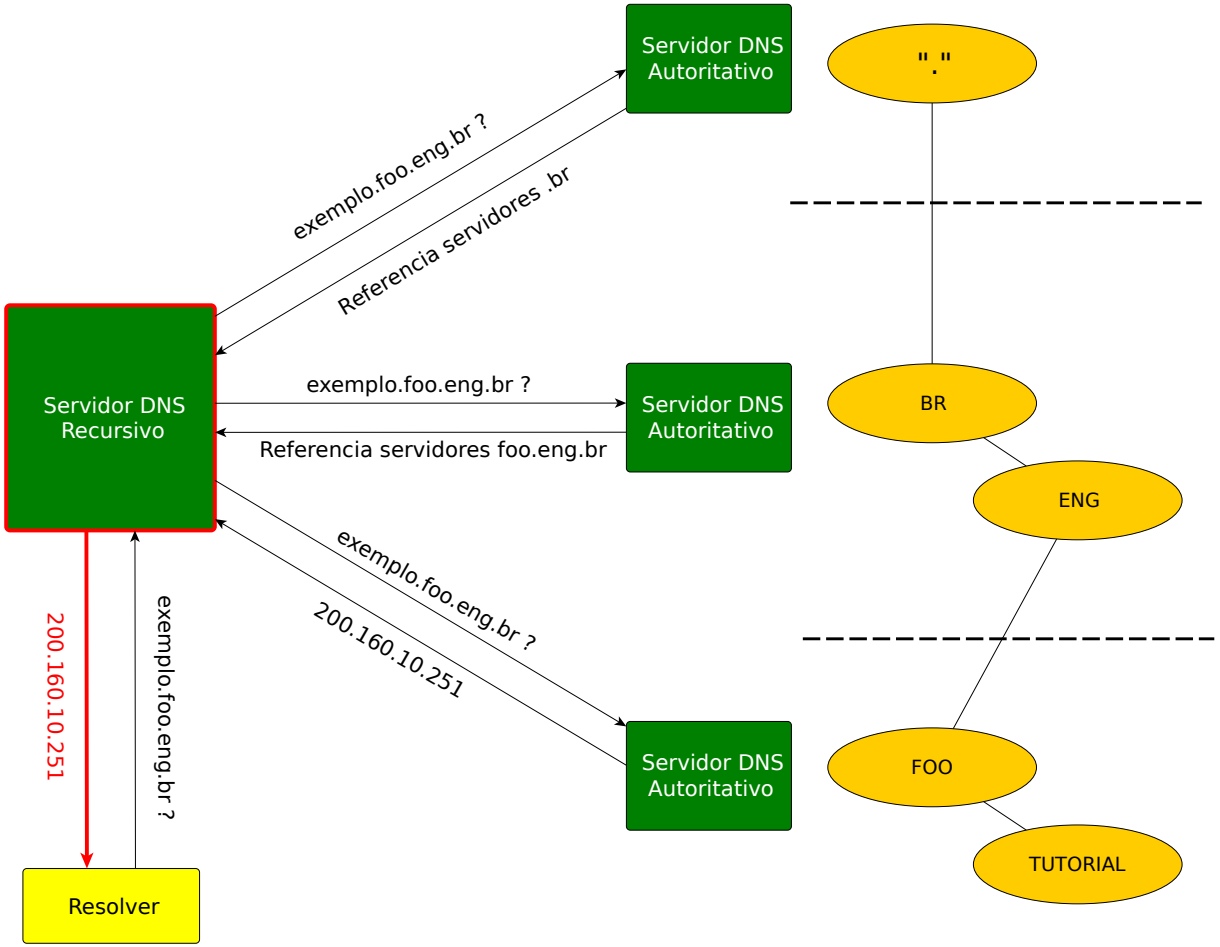
Address request example



Address request example

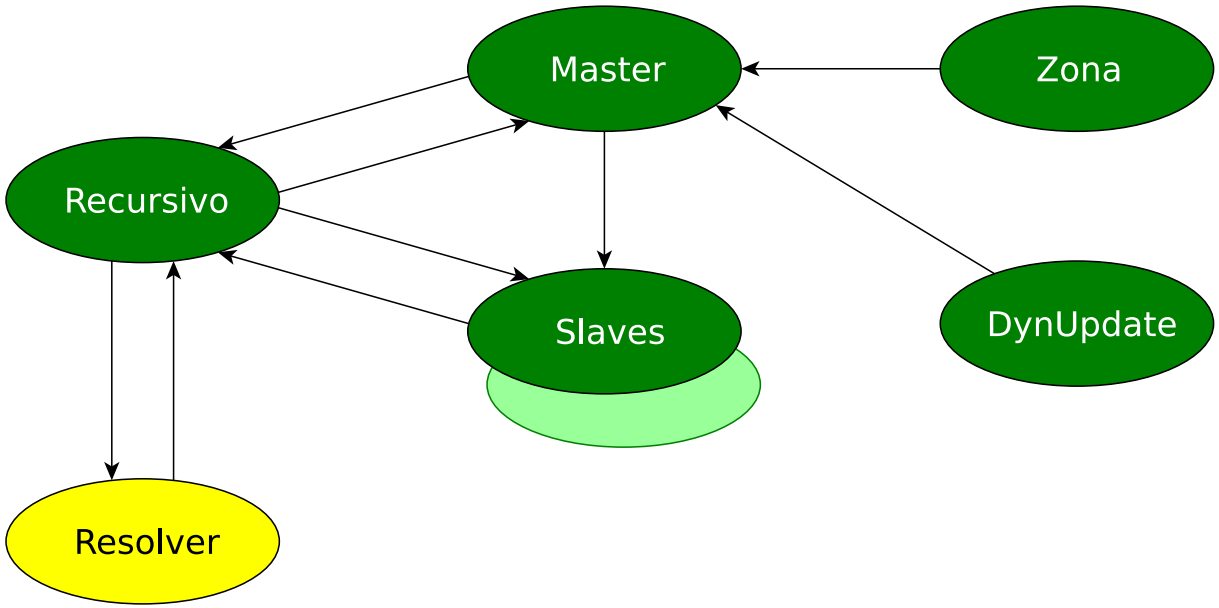


Address request example



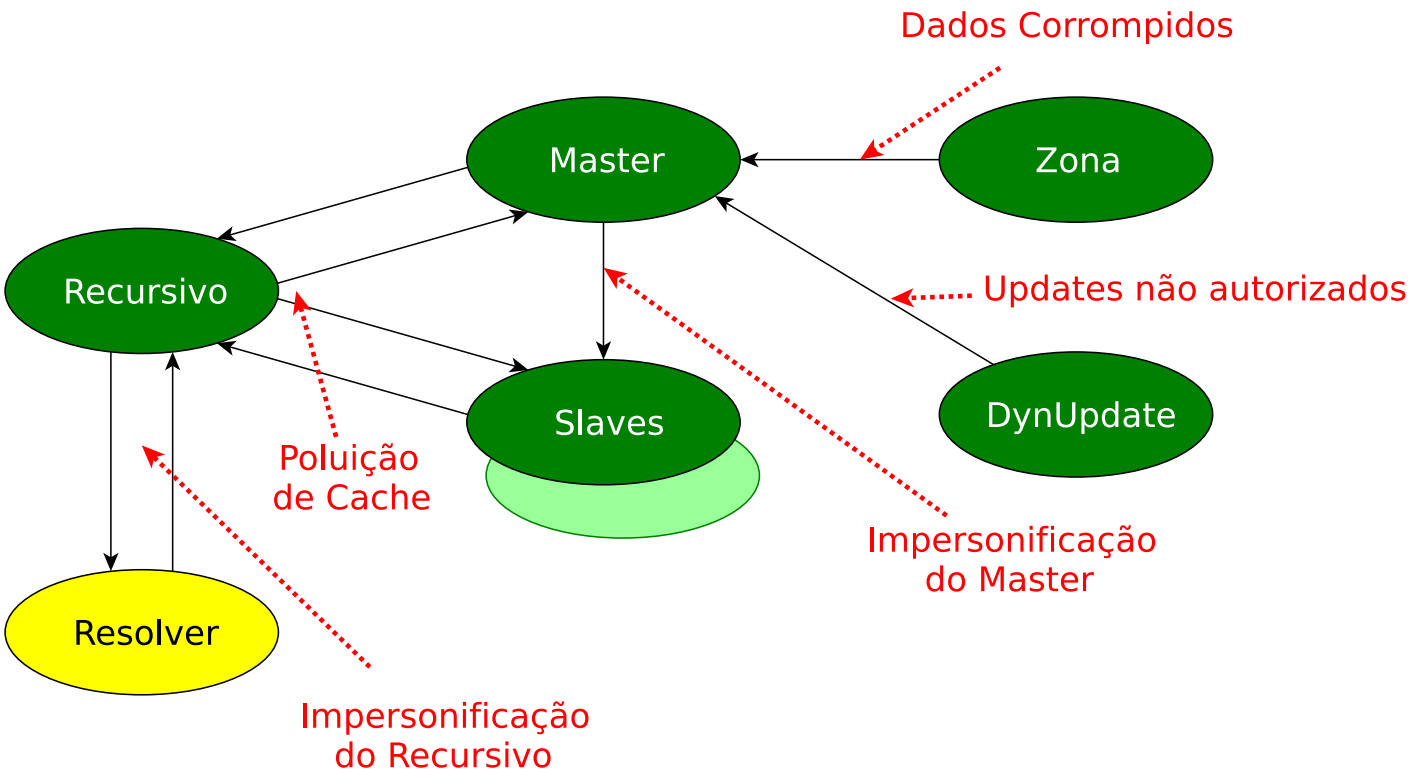
Address request example

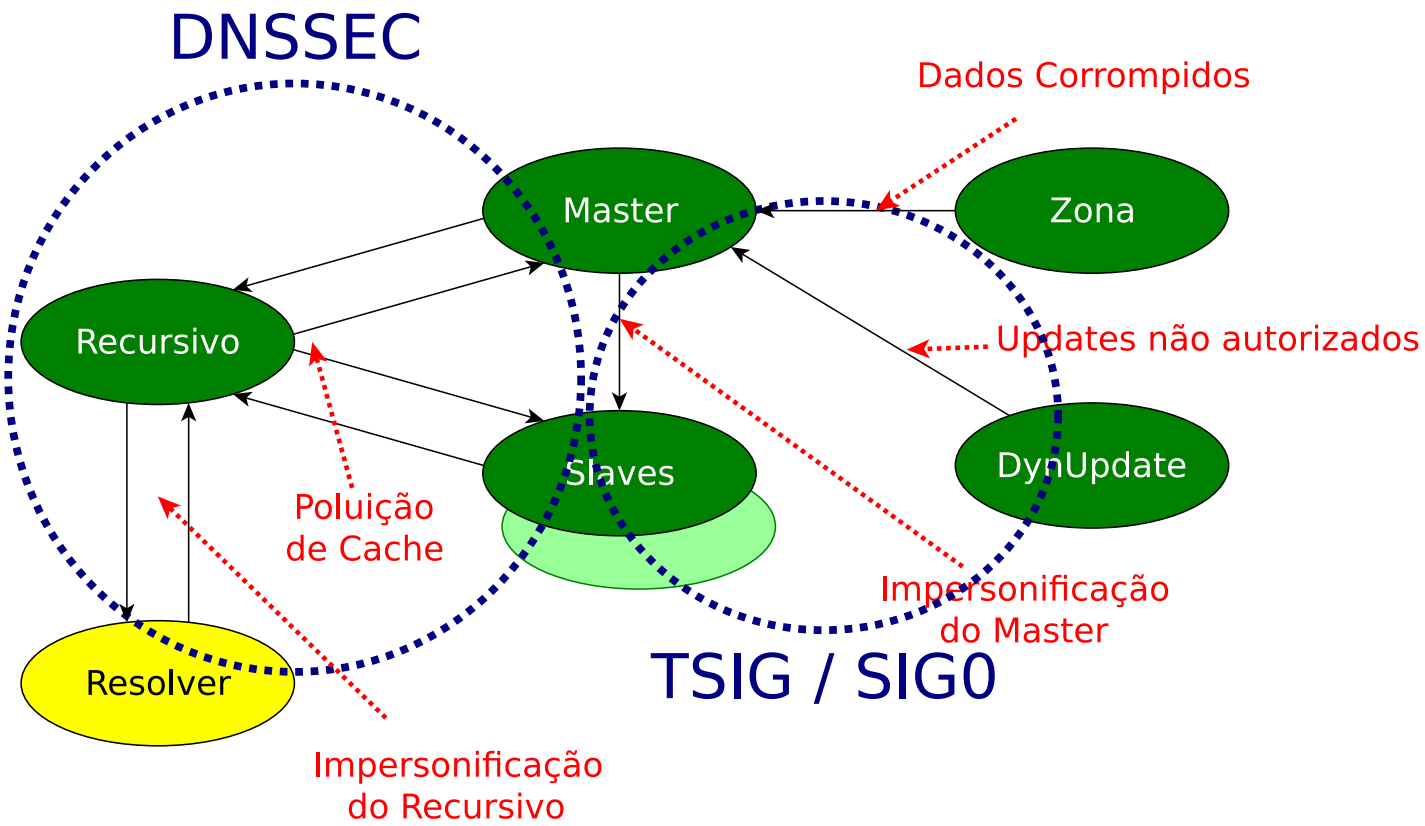
Data Flow



1. Resolver queries recursive server
2. Recursive queries Master or Slave authoritative server
3. Master server has the original zone data (zone file)
4. Slave server receives the zone data from Master (AXFR or IXFR)

Vulnerabilities

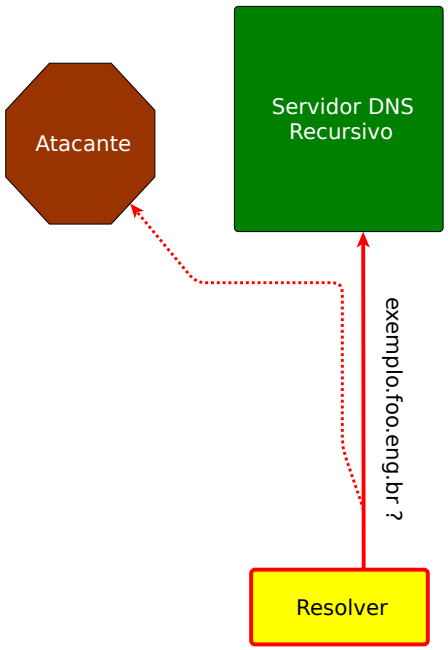



DNSSEC
Dados Corrompidos
Master
Zona
Recursivo
Updates não autorizados
Slaves
DynUpdate
Poluição
de Cache
Impersonificação
do Master
TSIG / SIG0
Resolver
Impersonificação
do Recursivo

Attack example 1

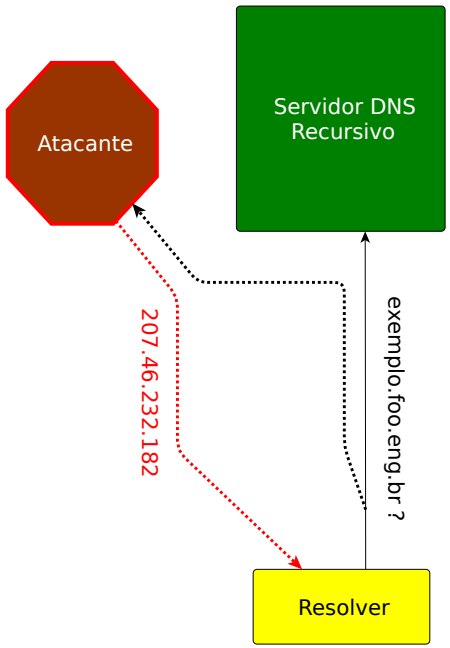
Man-in-The-Middle

Attack example 1
Man-in-The-Middle



Attack example 1
Man-in-The-Middle

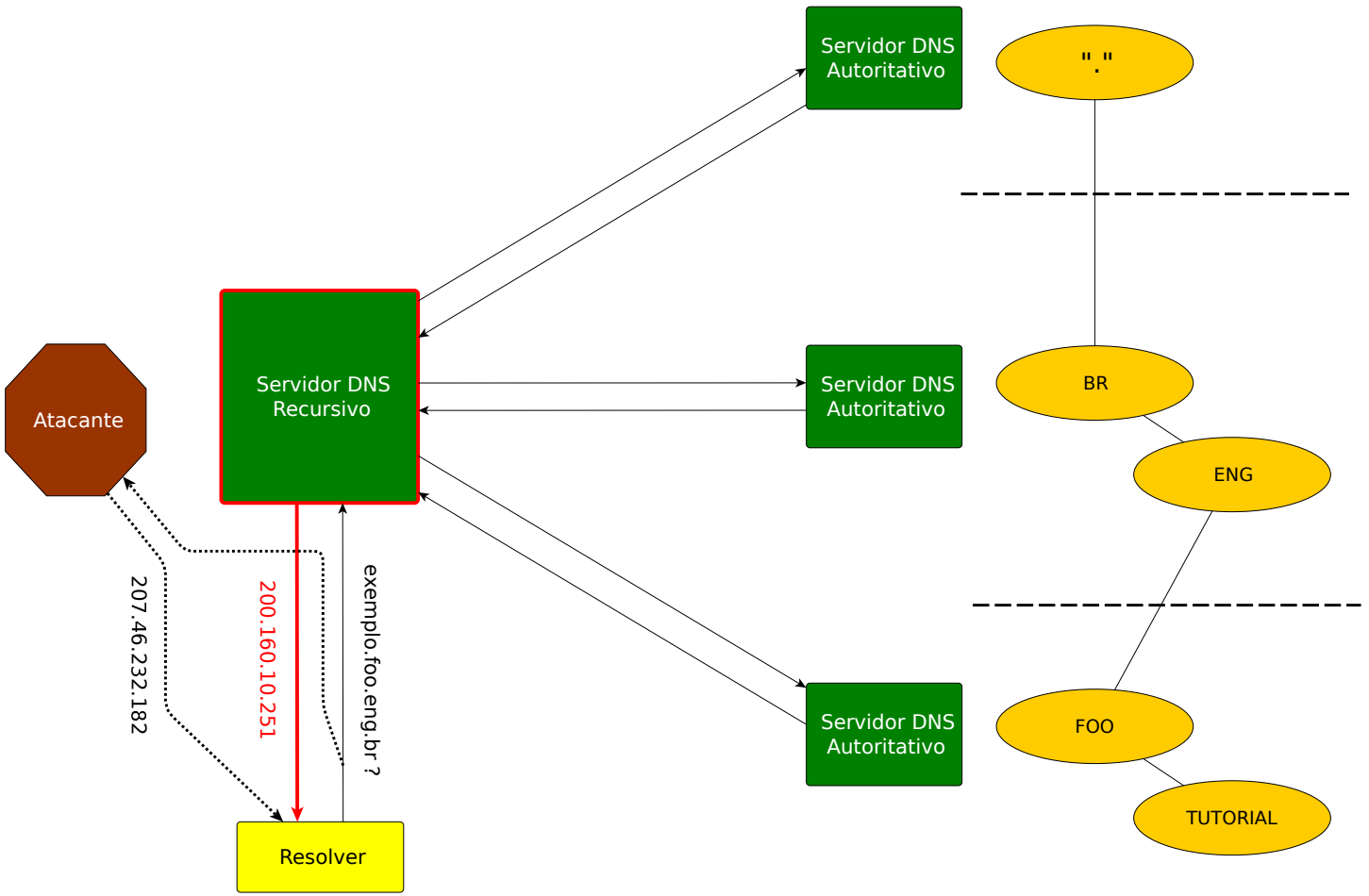
The attacker answers faster, spoofing the recursive IP address



Attack example 1

Man-in-The-Middle

The attacker answers faster, spoofing the recursive IP address



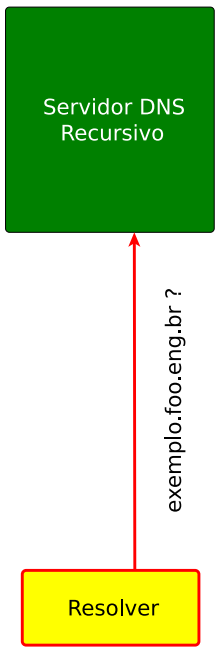
Attack example 2

Cache Poisoning

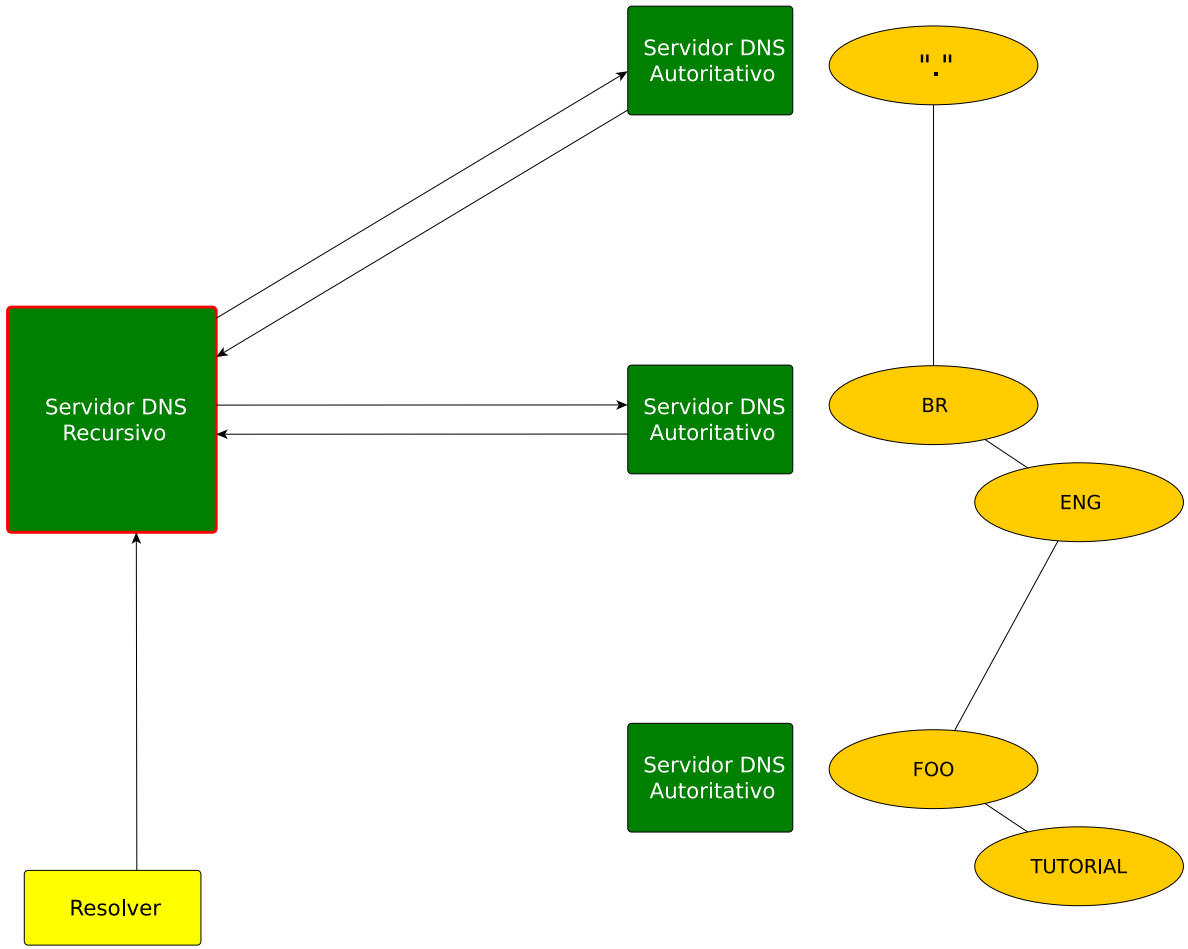
Resolver

Attack example 2

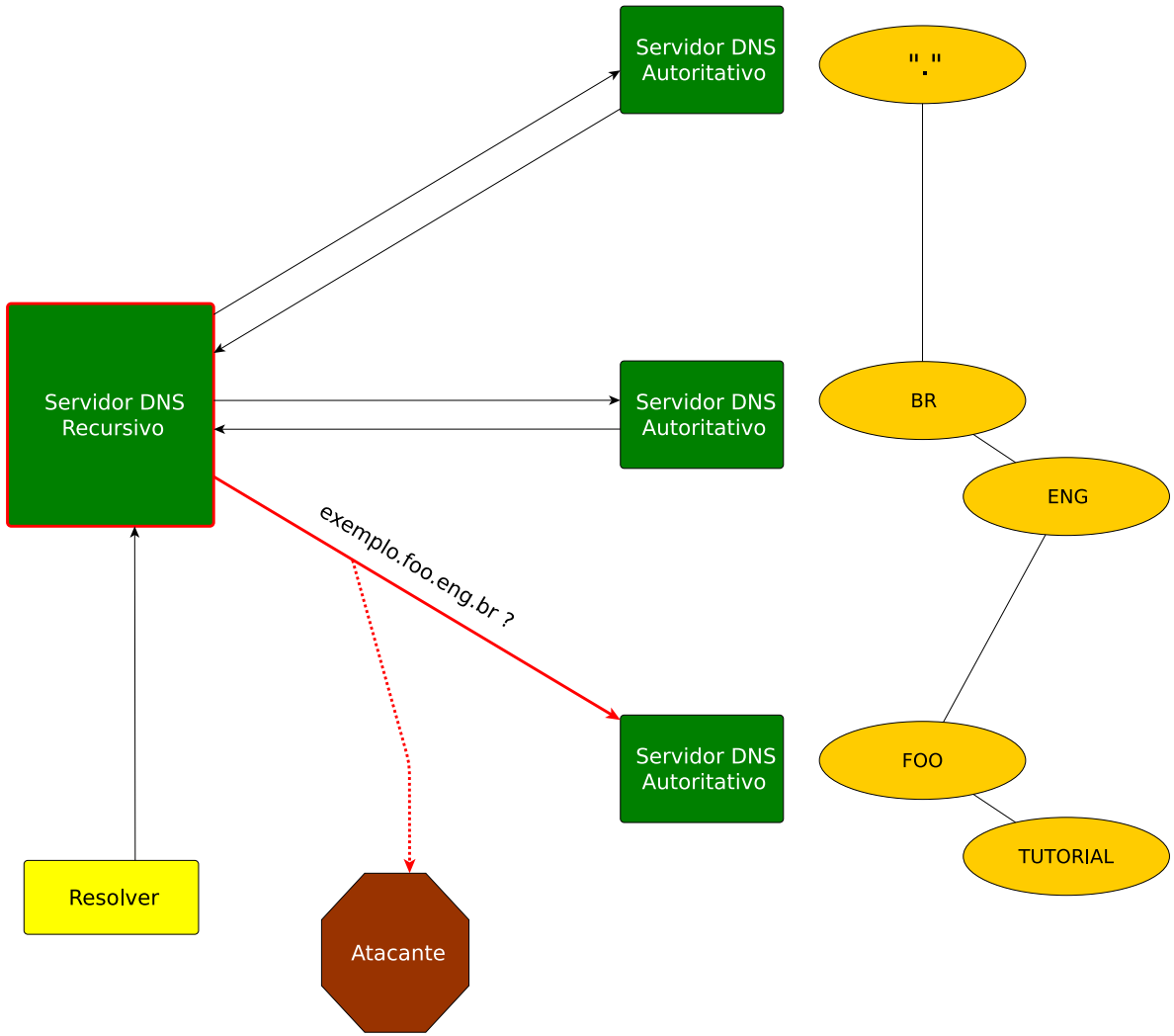
Cache Poisoning



Attack example 2
Cache Poisoning

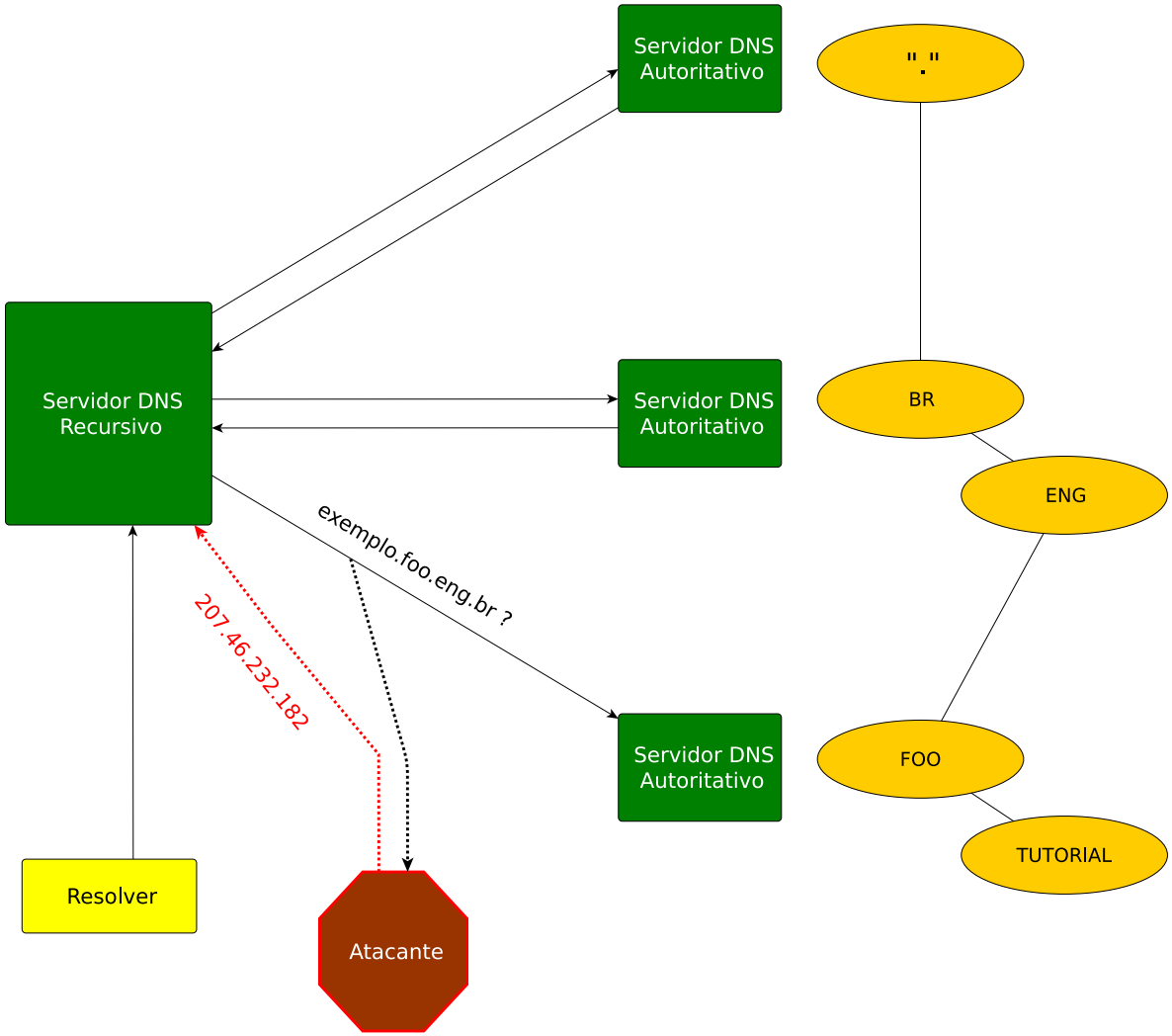


Attack example 2
Cache Poisoning



Attack example 2
Cache Poisoning

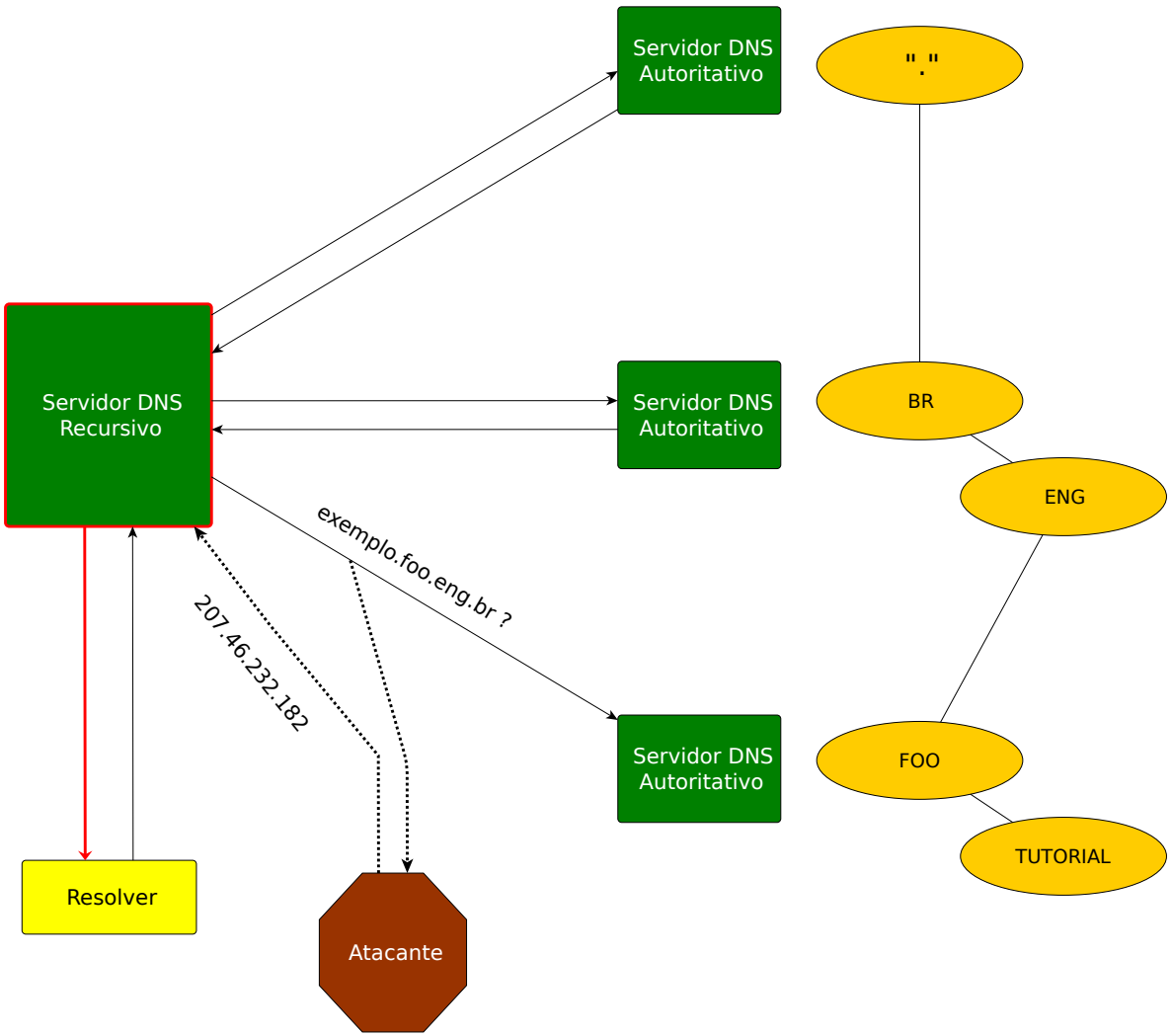
The attacker answers faster, spoofing the authoritative IP address



Attack example 2

Cache Poisoning

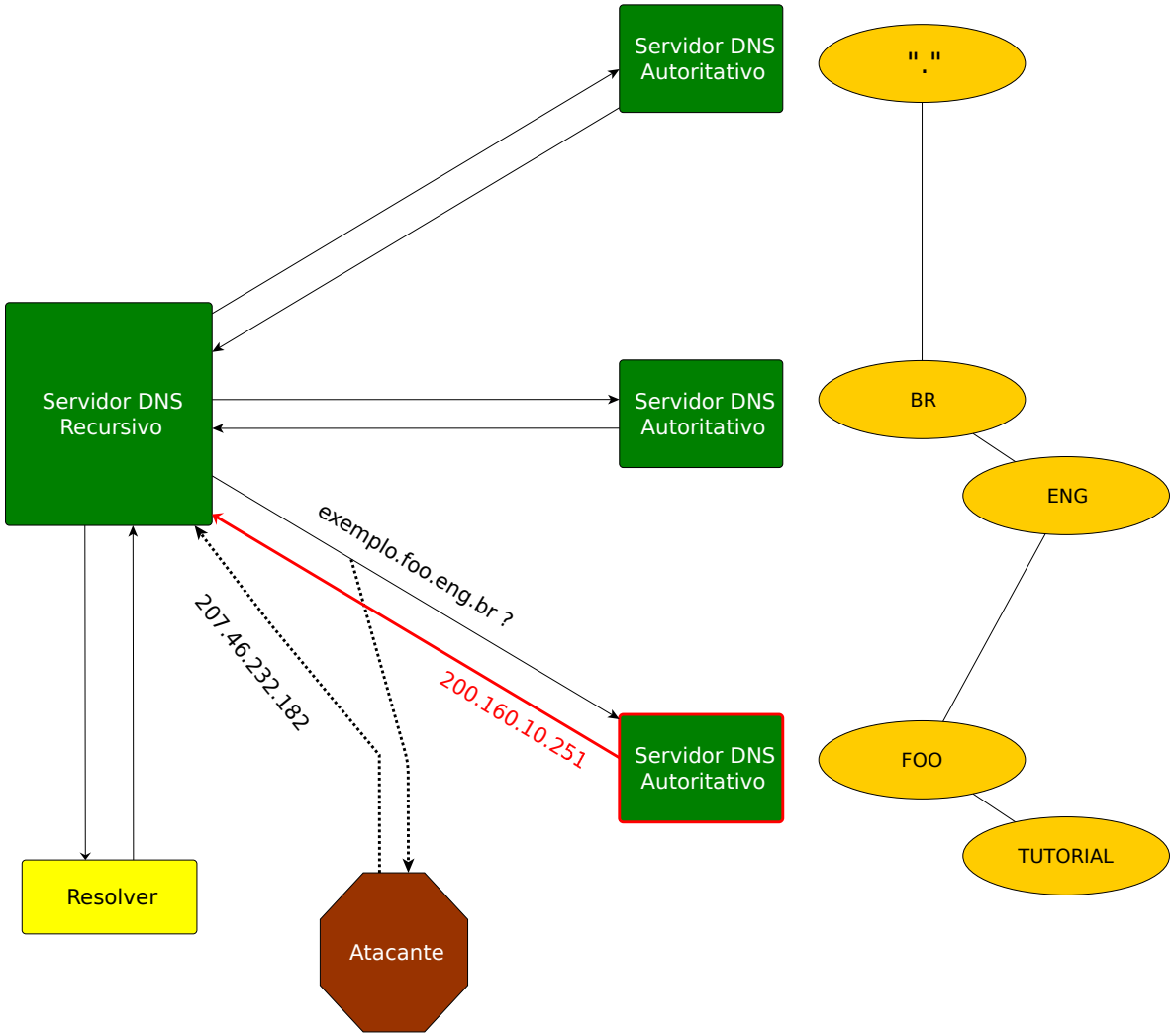
The attacker answers faster, spoofing the authoritative IP address



Attack example 2

Cache Poisoning

The attacker answers faster, spoofing the authoritative IP address



Target Enviroments

- Ethernet (not bridge 802.1d)
- Ethernet Wireless (802.11)

Target Enviroments

- Ethernet (not bridge 802.1d)
- Ethernet Wireless (802.11)

Attention: Be very careful in conferences!

registro.br
Registro de Domínios para a Internet no Brasil

TSIG

Transaction Signatures – RFC 2845

- Authentication of AXFR, IXFR and Dynamic Updates
- Authentication of the cache forwarder server
- All transactions are signed with a shared key

registro.br
Registro de Domínios para a Internet no Brasil

TSIG

Transaction Signatures – RFC 2845

- Authentication of AXFR, IXFR and Dynamic Updates
- Authentication of the cache forwarder server
- All transactions are signed with a shared key

DNSSEC

- Provides security on domain name resolution
- Works as a way of authenticity verification
- Verification occurs before other security applications (SSL, SSH, PGP, etc...)

DNSSEC

Domain Name System SECurity extensions

- An extension to the DNS protocol
  (plain old DNS still works the same way)
- Increases security to the Internet user
  (fixes some flaws in the DNS protocol)
- Current version is called DNSSEC bis with NSEC3 optional

DNSSEC features

What does it guarantee?

- Authenticity
- Integrity
- The non-existence of a domain name or type

DNSSEC features

What does it guarantee?

- Authenticity
- Integrity
- The non-existence of a domain name or type

What it does NOT guarantee?

- Confidentiality
- Protection against Denial Of Service attacks (DOS)

Adoption of DNSSEC around the world

registro.br
Registro de Domínios para a Internet no Brasil

World Wide DNSSEC Deployment

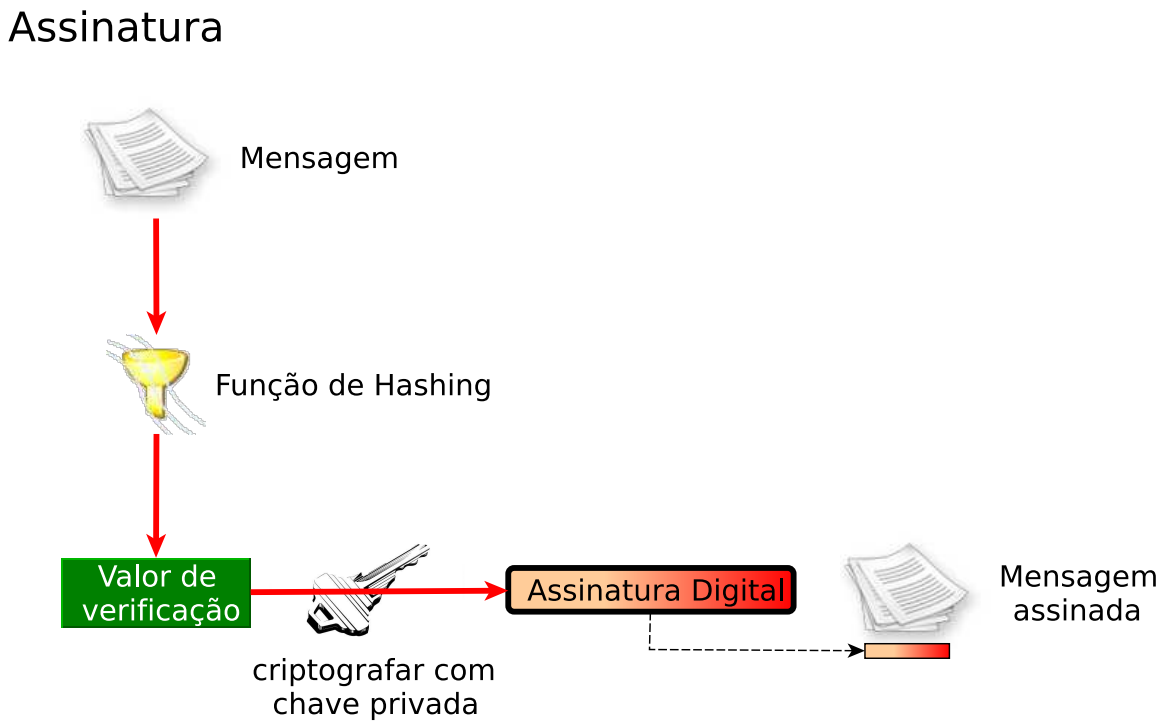
See also DNSSEC Theory and World Wide Deployment by Paul Wouters, November 21, 2007, SecTor

This map was created by Paul Wouters

Source: http://www.xelerance.com/dnssec/

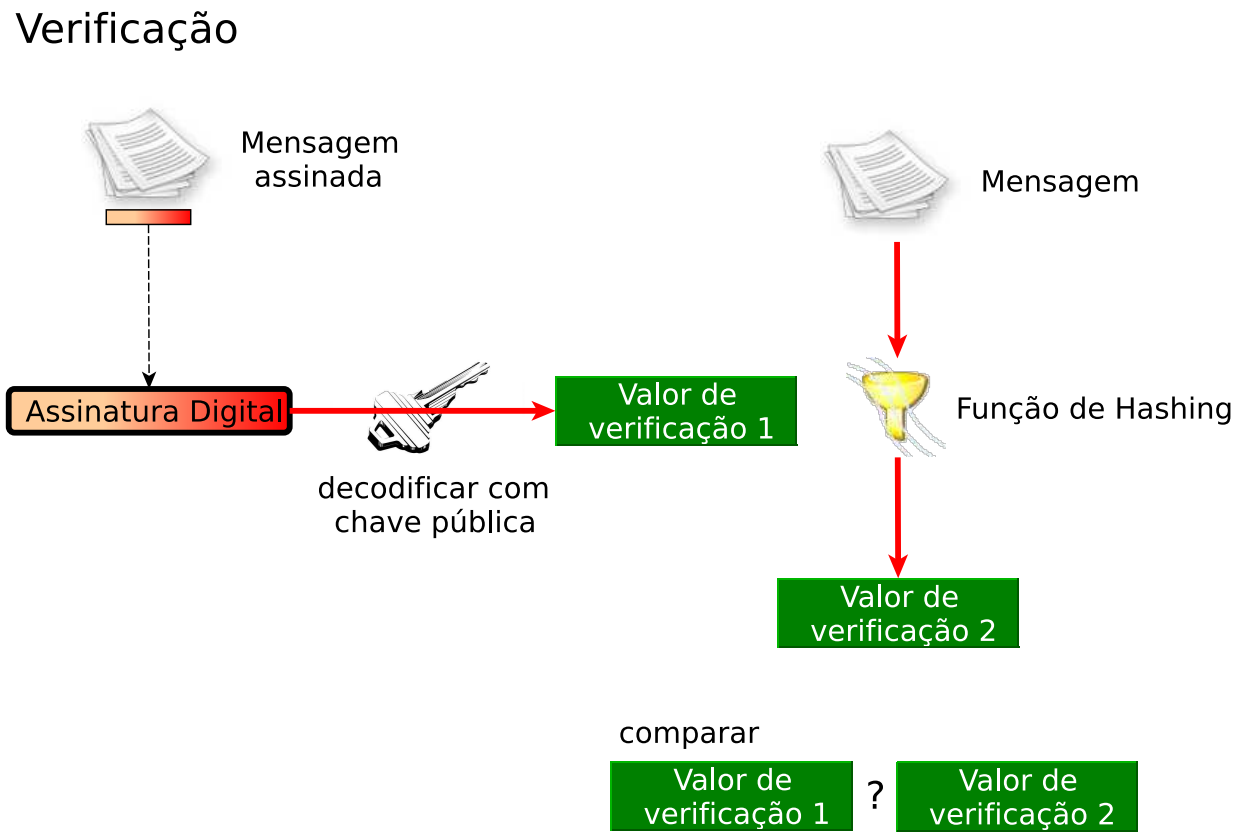
nic.br

Assymmetric Keys



DNSSEC works with the concept of assymetric keys
– Public key and private key

Assymmetric Keys



DNSSEC works with the concept of assymetric keys
– Public key and private key

New Resource Records

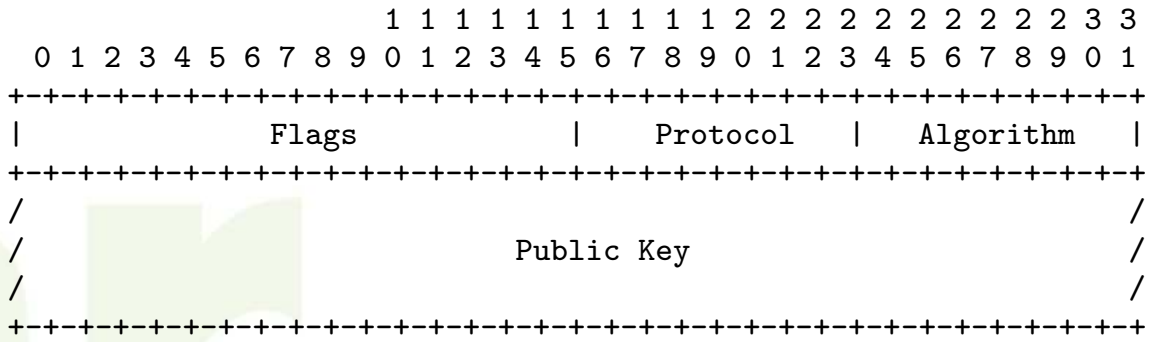
registro.br
Registro de Domínios para a Internet no Brasil

DNSKEY	Public key
RRSIG	Signature of a RRSET (only on authoritative records)
DS	Delegation Signer (Pointer to the chain of trust)
NSEC	Points to the next name in a zone and indicates what RRTYPEs are present in the current name

nic.br

DNSKEY

Resource Record that stores a zone's public key

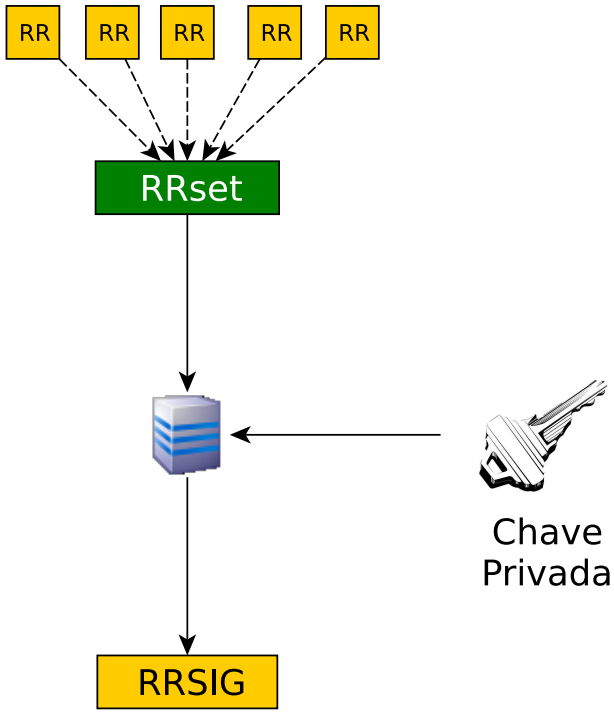


Example

```
foo.eng.br.          900 IN DNSKEY 256 3 5 (
                            AwEAAeZPN2yMs9q6kgYjFUblEwjCnWWcPq+TGcJrD5ga
                            XXAbP5MAqIkgZ5J4TU1mmpL1A8gMfd/wUmBkVipXR8FK
                            HRajBZSRfgeKnKaQtrxNZ32Ccts2F6Ylv9WaLXtiqebg
                            OZtuJFpQr6pnIt/FoOI+I7BUSNrX28VTq4jXu/qTrmM/
                            ) ; key id = 62745
```

RRSIG

- Resource Record that holds the signature of a RRSET created by a (DNSKEY)
- Has a validity period with determined by an inception date and an expiration date



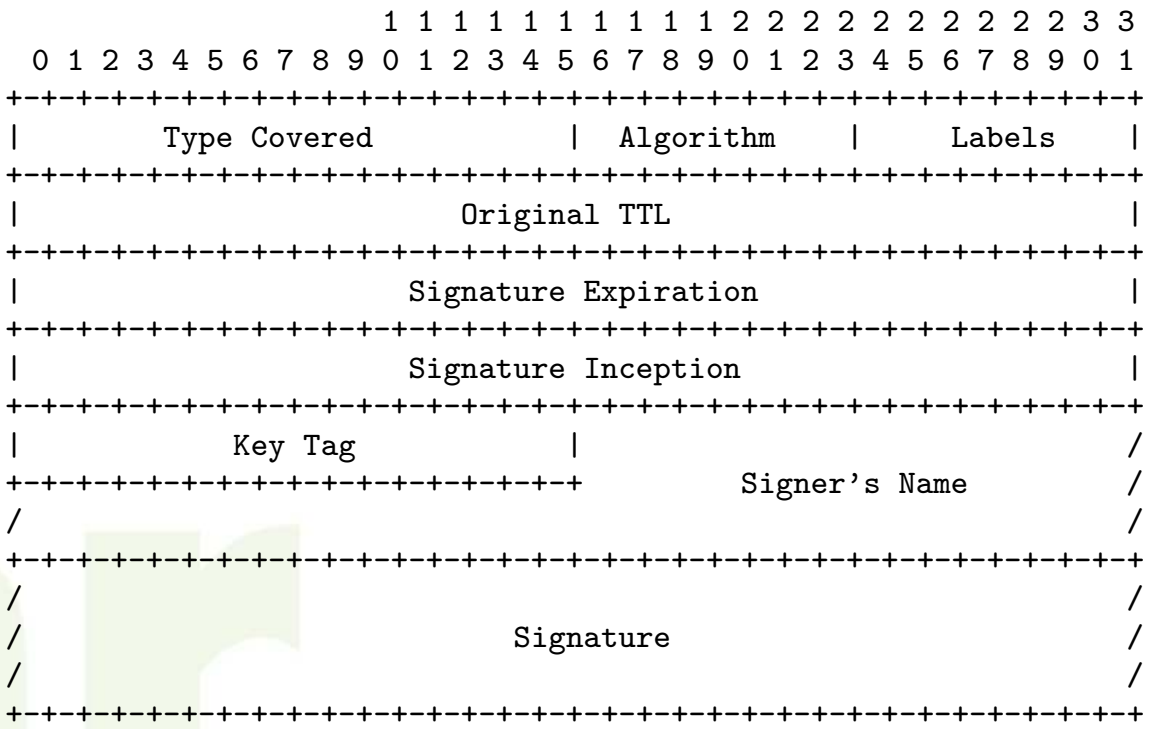
RRSET examples:

```
foo.eng.br.       IN NS ns1.foo.eng.br.
foo.eng.br.       IN NS ns2.foo.eng.br.
```

```
ns1.foo.eng.br.   IN A 200.160.3.97
```

```
ns2.foo.eng.br.   IN A 200.160.3.97
```

RRSIG



Exemplo

```
foo.eng.br.          900 IN RRSIG SOA 5 3 900 20070617200428 (
                            20070518200428 62745 foo.eng.br.
                            glEeCYyd/CCBfzH64y0RAQf9OxYDsI4xuBNaam+8DZQZ
                            xeoSLQEEtwmp6wBtQ7G10wSM9nEjRRhbZdNPNKJMp2PE
                            lLLgLI+BLwdlz0t8MypcpLOaTm9rc7pP7UR5XLzU1k8D
                            m6ePW1bNkId7i0IPSghyoHM7tPVdL2GW51hCujA= )
```

Contains a hash of a DNSKEY record

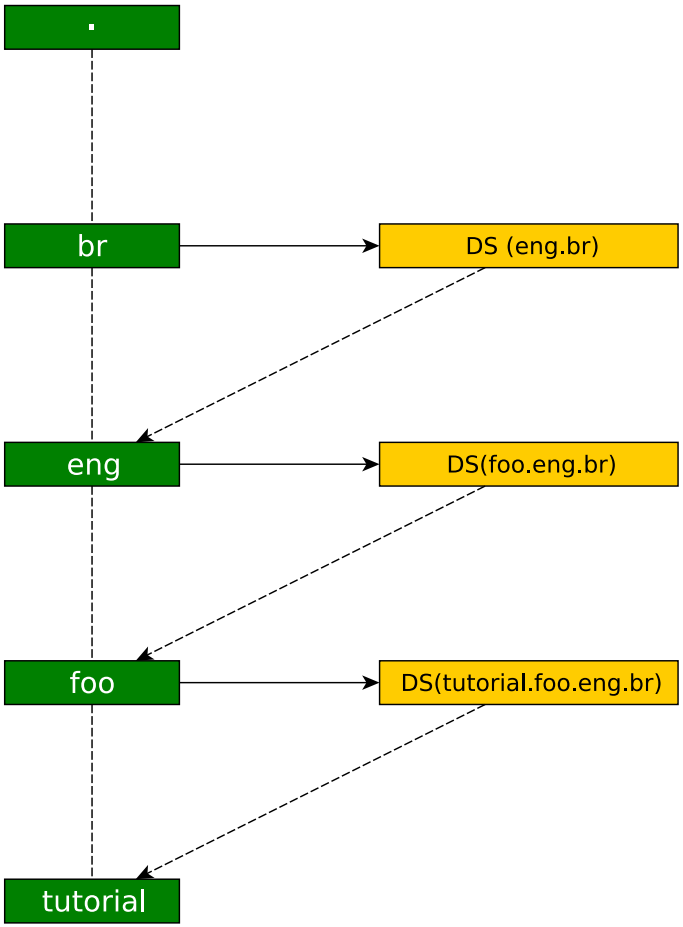
Used to guarantee the existence of a chain of trust between a domain and its sub-domains.

Indicates:

- that a delegated sub-zone is signed
- which key is being used the delegated sub-zone

The DS record is under the authority of the parent zone

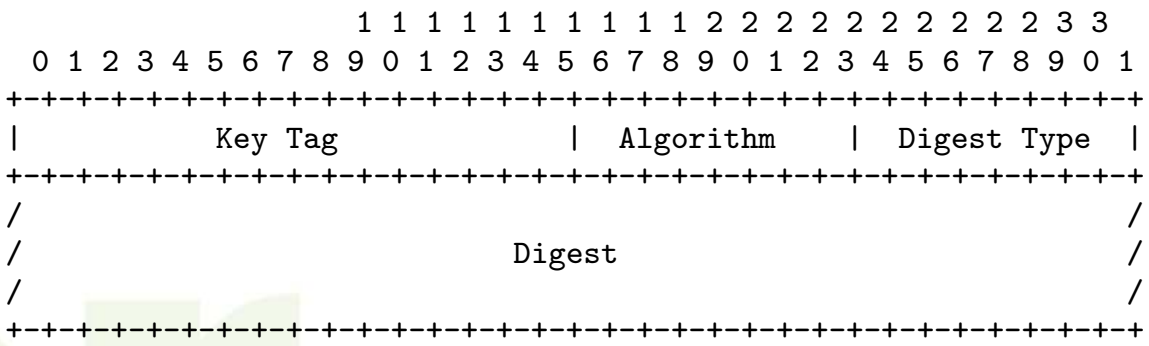
- The NS records are only “hints”, and aren’t authoritative on the parent
- The DS record should **NOT** appear on the child zone



Chain of Trust

The DS record is a pointer to a chain of trust, that guarantees the authenticity of the delegations of zone until a point of trust
– an anchored key or DLV (DNSSEC Lookaside Validation – slide 76)

DS



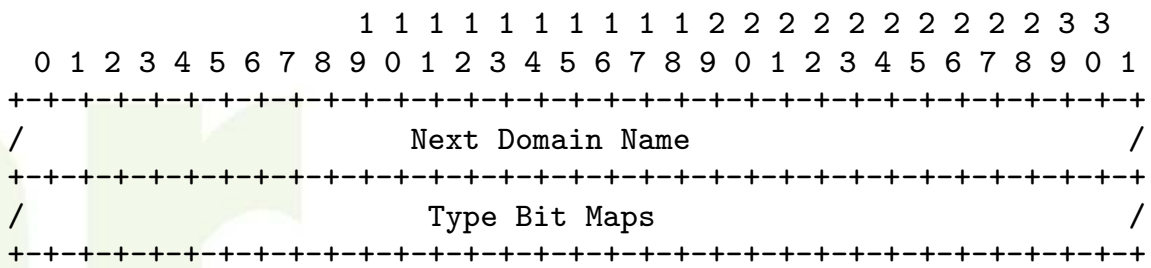
Example

```
foo.eng.br.          IN DS 817 5 1 EAEC29E4B0958D4D3DFD90CC70C6730AD5880DD3
```

NSEC - Next Secure

Authenticates a negative answer

- Next secure name
- Indicate the existing RRTYPEs for the domain name
- The last name points back to the first (SOA)



Example

```
foo.eng.br.          900 IN NSEC ns1.exemplo.foo.eng.br. NS SOA RRSIG NSEC DNSKEY
```

Proof of non-existence, pre-signed, without the need of having online keys for on-demand signatures, avoiding the possibility of DOS attacks.

NSEC - Non-existent name simulation

- Answers **NXDOMAIN**
  - One or more NSEC records indicate that the name or wildcard doesn't exist

```
$ dig @200.160.10.251 zzz.foo.eng.br SOA +dnssec
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 18301
;; flags: qr aa rd; QUERY: 1, ANSWER: 0, AUTHORITY: 6, ADDITIONAL: 1
;; QUESTION SECTION:
;zzz.foo.eng.br.          IN    SOA
;; AUTHORITY SECTION:
foo.eng.br.        0      IN    SOA      ns1.foo.eng.br. hostmaster.foo.eng.br. 1 3600 3600 3600 900
foo.eng.br.        0      IN    RRSIG    SOA 5 3 900 20070617200428 20070518200428 62745 foo.eng.br.
                                glEeCYyd/CCBfzH64y0RAQf9OxYDsI4xuBNaam+8DZQZxeoSLQEEtwmp
                                6wBtQ7G10wSM9nEjRRhbZdNPNKJMp2PElLLgLI+BLwdlz0t8MypcpLOa
                                Tm9rc7pP7UR5XLzU1k8Dm6ePW1bNkId7i0IPSghyoHM7tPVdL2GW51hCujA=
foo.eng.br.      900   IN    NSEC     ns1.exemplo.foo.eng.br. NS SOA RRSIG NSEC DNSKEY
foo.eng.br.        900    IN    RRSIG    NSEC 5 3 900 20070617200428 20070518200428 62745 foo.eng.br.
                                OCOCpFW5fR6MPhVBaUWfrP9pkIqVc+NDORi6PRwIX/p1dLmAT7NF5Rkc
                                9IfbAHZTxefoqTKqN/vPl1PqSxUzh0rl+atHblaH6yt79CTkmStota7C
                                SLYYXX5c7D93hRYJ2yk1COxQz6GG9SIp/U4qR4//TcQDHpqQ4bFs42ZsD4I=
ns2.foo.eng.br.  900   IN    NSEC     foo.eng.br. A RRSIG NSEC
ns2.foo.eng.br.    900    IN    RRSIG    NSEC 5 4 900 20070617200428 20070518200428 62745 foo.eng.br.
                                XVf7M09L4rVUD6uxa1P+EhQYohuimuwk1xzAemsn292esUhkkYz/BG7b
                                OT/L9fhzOEPYtYGFyMF4gZ1/mxwY31UmX6xVZZPYFJ7x5Kw2uTSD49FK
                                VsdUOLBCAHzO88byAm8EwLe3l+U0/q8RvPimAfpouoivUDcuWtKxs0CzLyc=
```

NSEC - Non-existent RRTYPE

- Answer **NOERROR** + no answer (ANSWER = 0)
  - The NSEC record proves that the queried RRTYPE does not exist

```
$ dig @200.160.10.251 foo.eng.br TXT +dnssec
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 60466
;; flags: qr aa rd; QUERY: 1, ANSWER: 0, AUTHORITY: 4, ADDITIONAL: 1
;; QUESTION SECTION:
;foo.eng.br.            IN    TXT
;; AUTHORITY SECTION:
foo.eng.br.    900    IN    SOA     ns1.foo.eng.br. hostmaster.foo.eng.br. 1 3600 3600 3600 900
foo.eng.br.    900    IN    RRSIG   SOA 5 3 900 20070617200428 20070518200428 62745 foo.eng.br.
                            glEeCYyd/CCBfzH64y0RAQf9OxYDsI4xuBNaam+8DZQZxeoSLQEEtwmp
                            6wBtQ7G10wSM9nEjRRhbZdNPNKJMp2PElLLgLI+BLwdlz0t8MypcpLOa
                            Tm9rc7pP7UR5XLzU1k8Dm6ePW1bNkId7i0IPSghyoHM7tPVdL2GW51hCujA=
foo.eng.br.    900    IN    NSEC    ns1.exemplo.foo.eng.br. NS SOA RRSIG NSEC DNSKEY
foo.eng.br.    900    IN    RRSIG   NSEC 5 3 900 20070617200428 20070518200428 62745 foo.eng.br.
                            OCOCpFW5fR6MPhVBaUWfrP9pkIqVc+NDORi6PRwIX/p1dLmAT7NF5Rkc
                            9IfbAHZTxefoqTKqN/vPl1PqSxUzh0rl+atHblaH6yt79CTkmStota7C
                            SLYYXX5c7D93hRYJ2yk1COxQz6GG9SIp/U4qR4//TcQDHpqQ4bFs42ZsD4I=
```

NSEC - Existing name

NSEC record is used only to prove non-existence. Does not appear on positive answers

```
$ dig @200.160.10.251 foo.eng.br SOA +dnssec +noadditional
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 6372
;; flags: qr aa rd; QUERY: 1, ANSWER: 2, AUTHORITY: 3, ADDITIONAL: 5
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;foo.eng.br.            IN    SOA
;; ANSWER SECTION:
foo.eng.br.    900     IN    SOA     ns1.foo.eng.br. hostmaster.foo.eng.br. 1 3600 3600 3600 900
foo.eng.br.    900     IN    RRSIG   SOA 5 3 900 20070617200428 20070518200428 62745 foo.eng.br.
                             glEeCYyd/CCBfzH64y0RAQf9OxYDsI4xuBNaam+8DZQZxeoSLQEEtwmp
                             6wBtQ7G10wSM9nEjRRhbZdNPNKJMp2PElLLgLI+BLwdlz0t8MypcpLOa
                             Tm9rc7pP7UR5XLzU1k8Dm6ePW1bNkId7i0IPSghyoHM7tPVdL2GW51hCujA=
;; AUTHORITY SECTION:
foo.eng.br.    900     IN    NS      ns2.foo.eng.br.
foo.eng.br.    900     IN    NS      ns1.foo.eng.br.
foo.eng.br.    900     IN    RRSIG   NS 5 3 900 20070617200428 20070518200428 62745 foo.eng.br.
                             3iLm1ROC+UeqYk0xgQGQQXkBzcKiKQRPwe+1JZlpjEzjU1Uj0HUOHefa
                             jXzMv7FlFMWYeU51Ybg49HFe67XQVlK54GeAFXWB7YS59yODLoNEBxQl
                             9QEy6g/OOnLpuKTrST8qqd5Fc/eYqN/Ag3GnfcAviZgiQhhveGH9mJHWZyc=
```

NSEC3 - DNSSEC Hashed Authenticated Denial of Existence

- Solves the problem of "Zone Walking"
- Replaces the NSEC record
- Contains the hash of a name in the zone and sorted in canonical order
- Instead of pointing to the next name in the zone, points to the next hashed name

Operation

- Authenticity and Integrity are provided by the signature of the RRSET created with a private key
- Delegated zones (child) sign the RRSETs with a private key
  - The authenticity of the key is verified by the signature of the DS record, present in the parent zone (Hash of the public key – DNSKEY –)
- The public key is used to verify the signature of a RRSET (RRSIG)
- Authenticity of the non-existence of a name or type is provided by a chain of names (NSEC), in which each name points the next in the zone, in canonical order

DNSSEC is not PKI

- There are no certificates
  (Certification Authority, Service Level Agreement, Certificate Revogation List)
- The keys don't expire
- The signatures have a validity period
  (`inception` and `expiration` on the RRSIG)
- Key handling policy is local to the zone

Anchoring a key - Recursive Server

In a recursive server with DNSSEC enabled, it is necessary to have a trusted public key anchored.

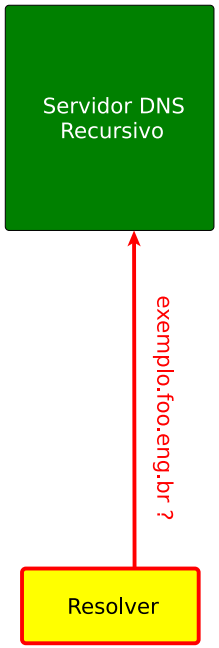
That indicates the a point of trust in the chain of trust.

Obtaining the public key for the ".br" zone

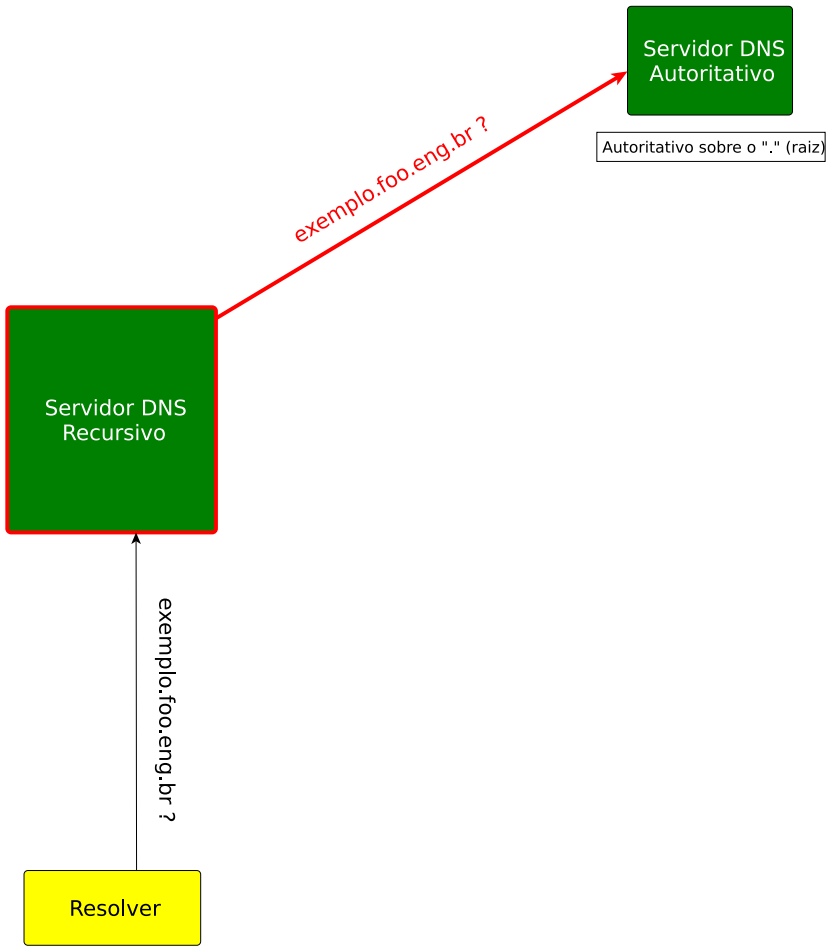
- **.BR** `https://registro.br/ksk/`

Example

- The recursive resolver already has the public key for the “.br” zone anchored

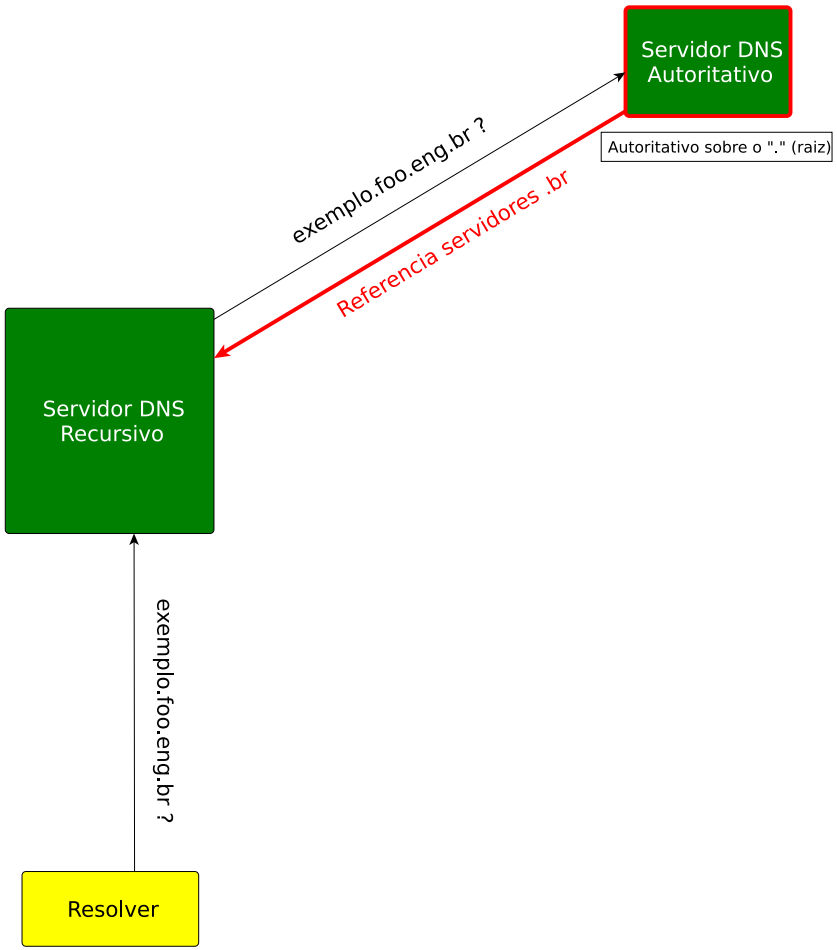


Example



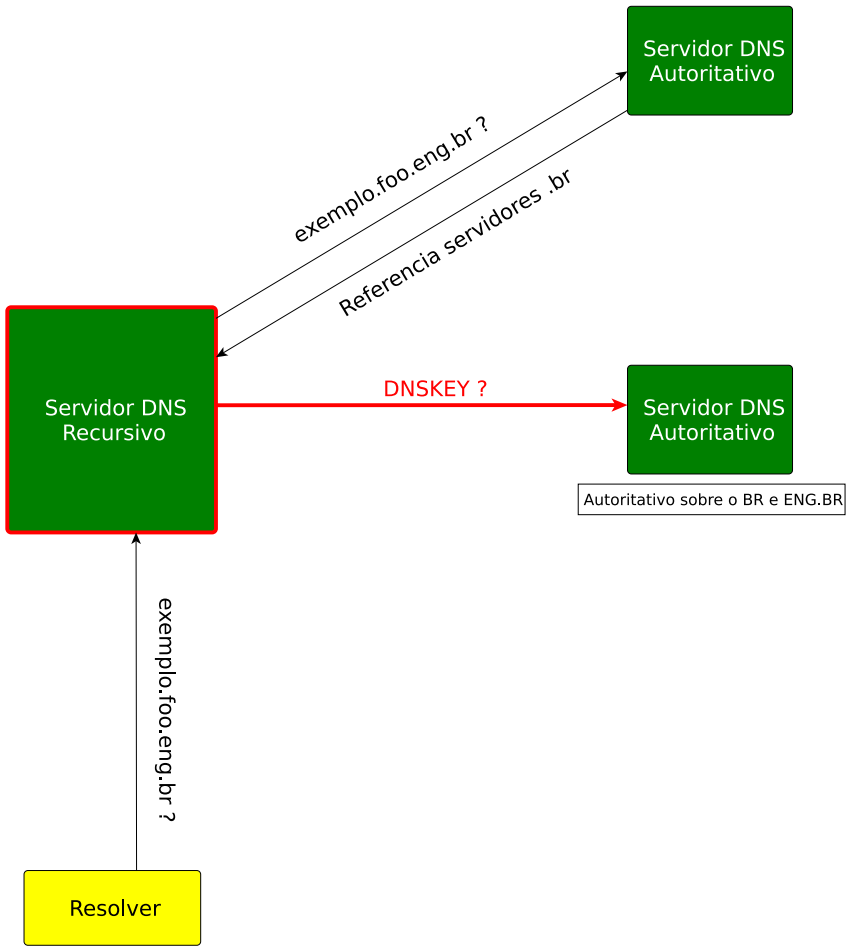
- In this simulation, we assume that the root zone is not signed.

Example



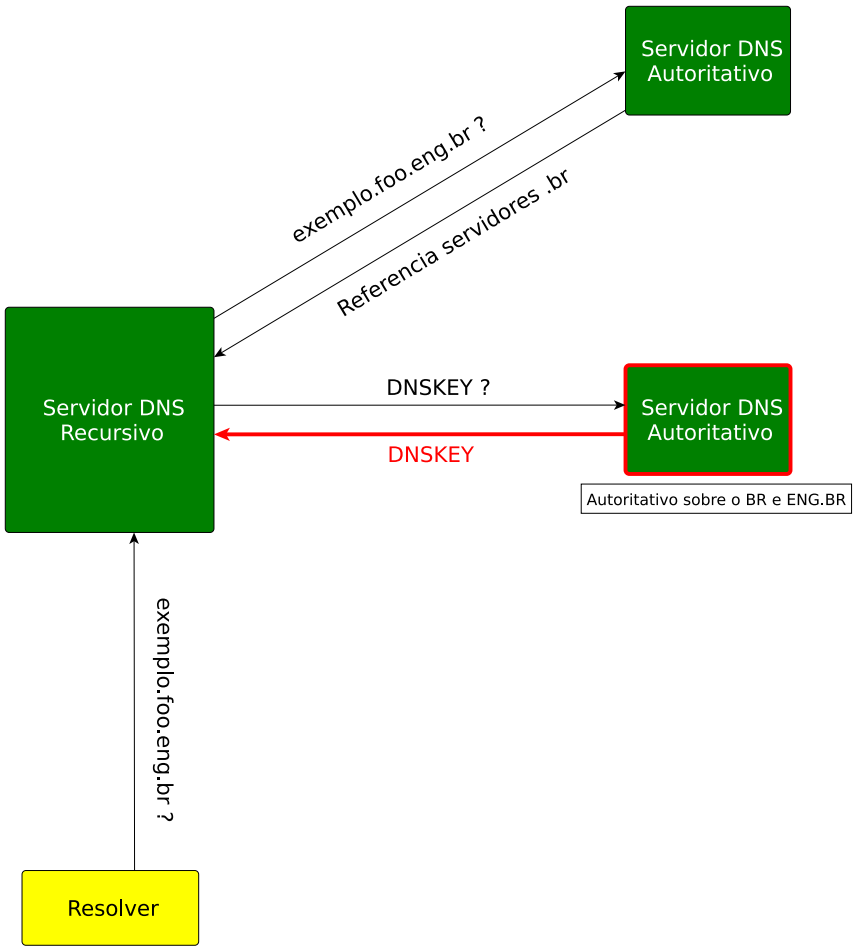
- No answer, but with a referral for the NS records on the ".br" zone.

Example



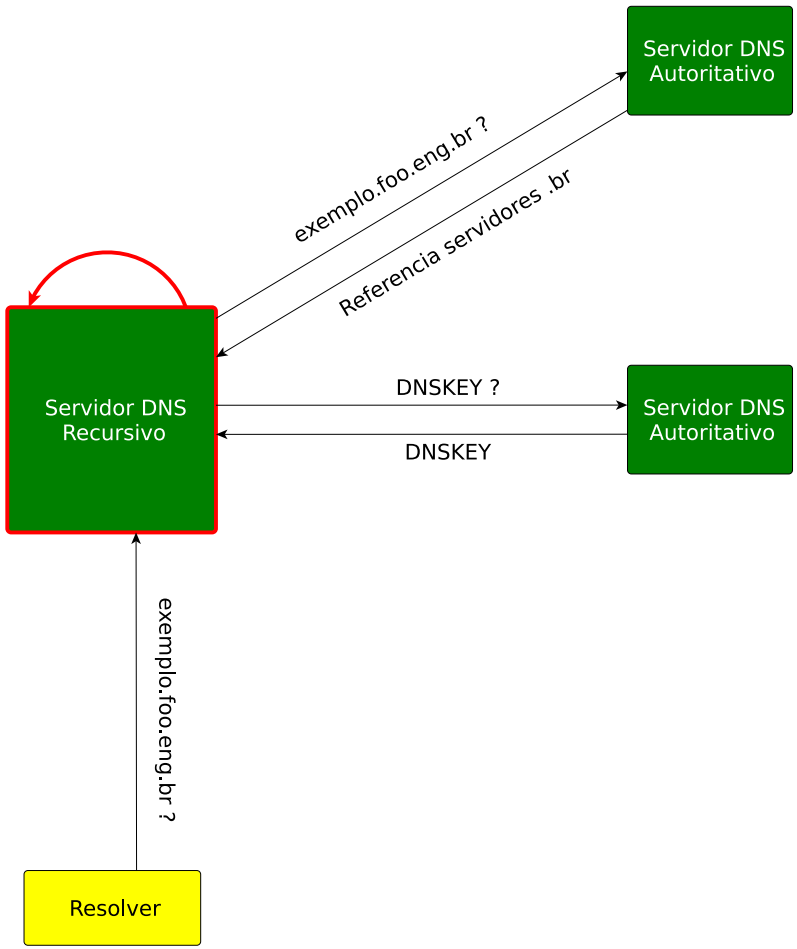
- The recursive server queries for the DNSKEY when it sees that the zone name is the same as the one on its trusted-key.

Example



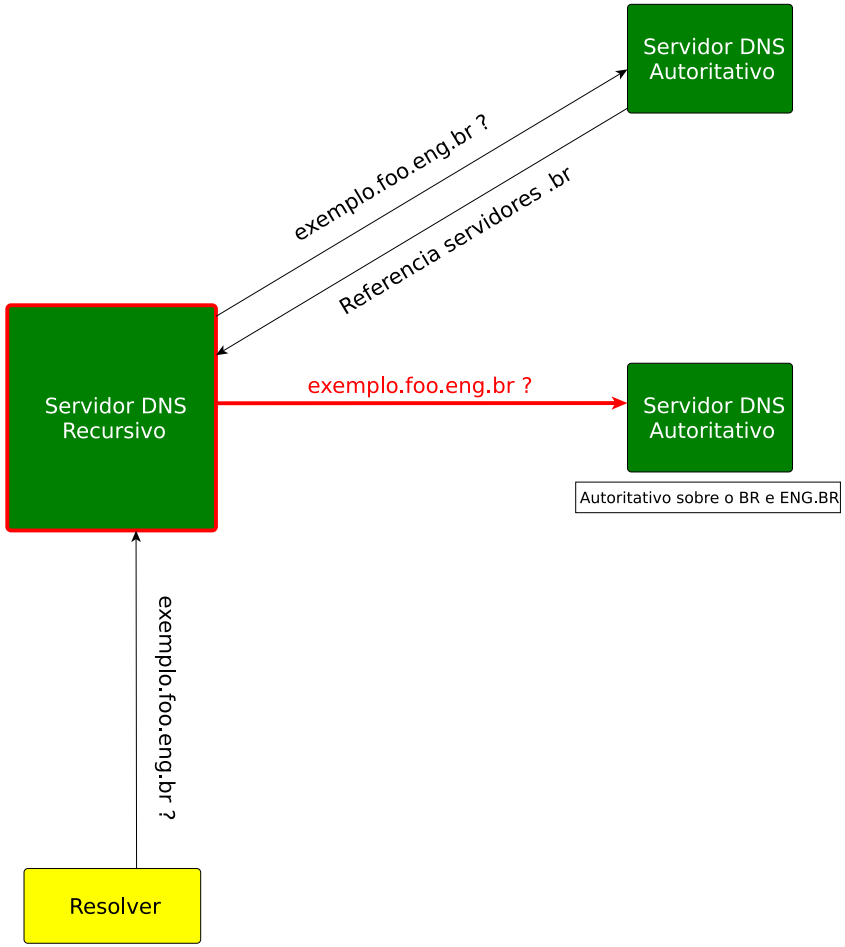
- The server responds with the DNSKEY and the RRSIG

Example

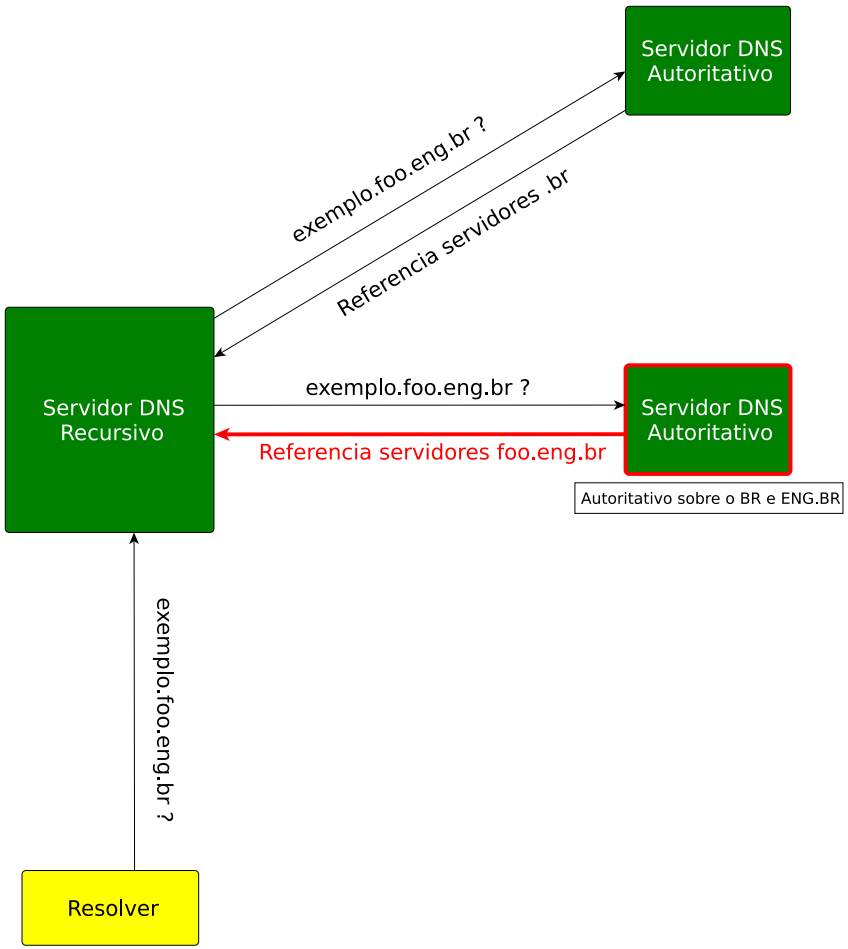


- Compares the trusted-key with the DNSKEY. In case it's valid, go on with the queries

Example


Servidor DNS
Autoritativo
exemplo.foo.eng.br ?
Referencia servidores .br
Servidor DNS
Recursivo
exemplo.foo.eng.br ?
Servidor DNS
Autoritativo
Autoritativo sobre o BR e ENG.BR
exemplo.foo.eng.br ?
Resolver

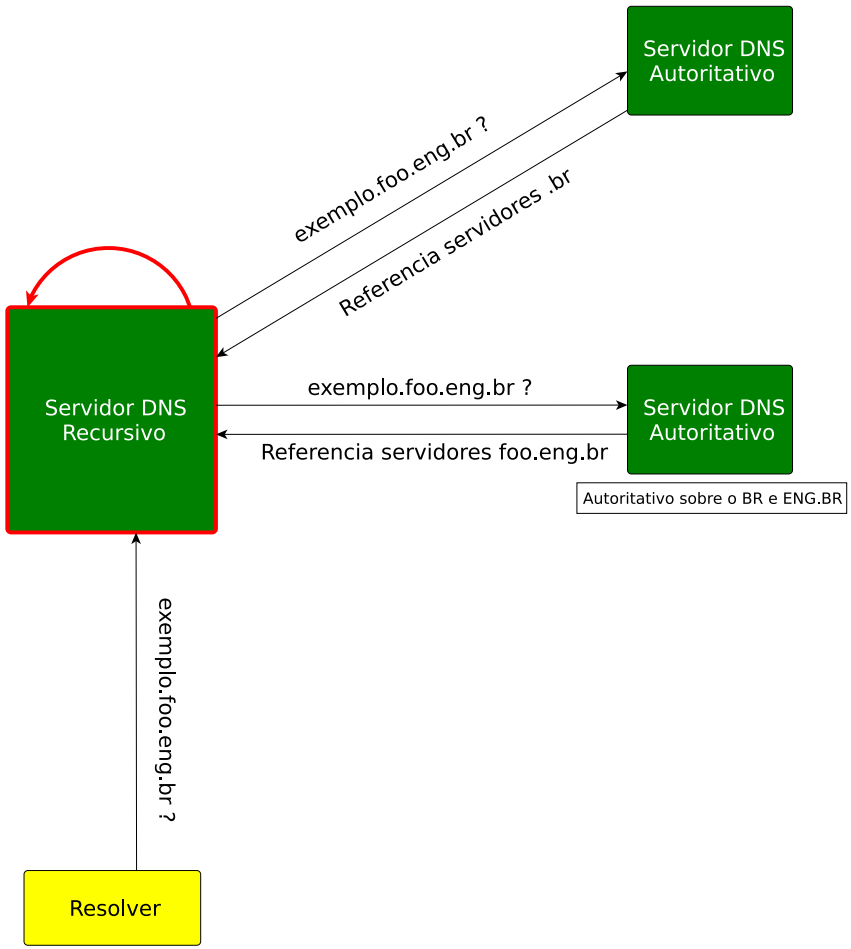
Example



- No answer but with referral:
  - NS "foo.eng.br"

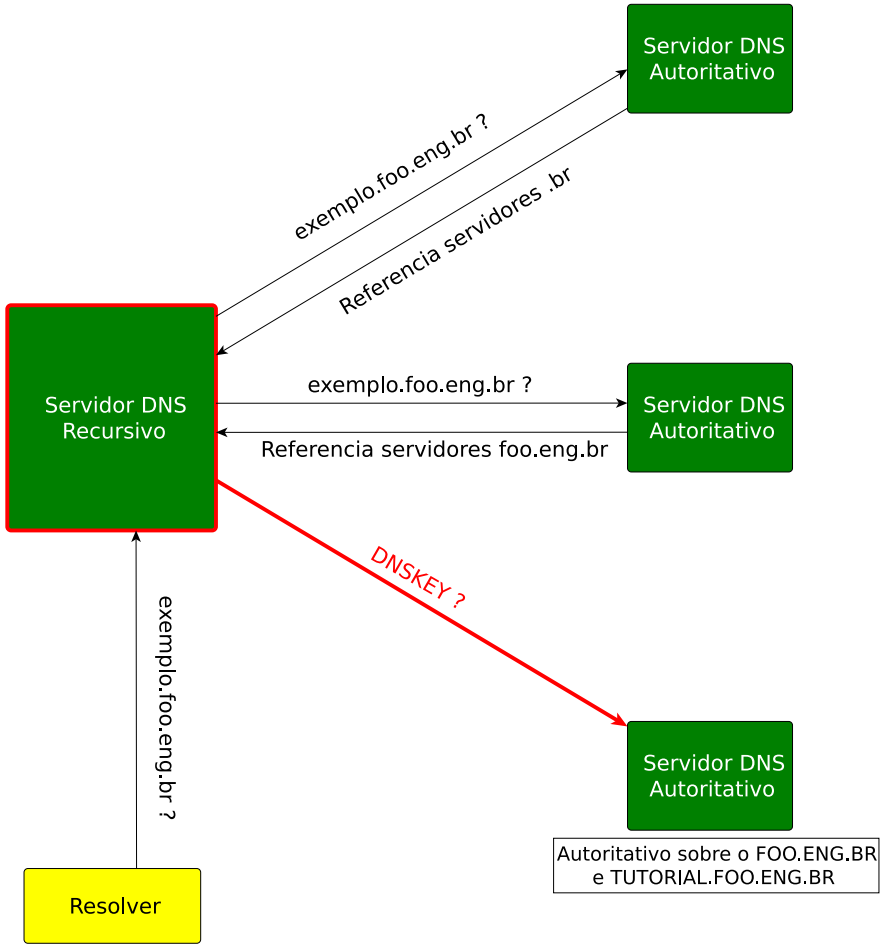
  And authority over the records:
  - DS do "foo.eng.br"
  - RRSIG do Record DS

Example

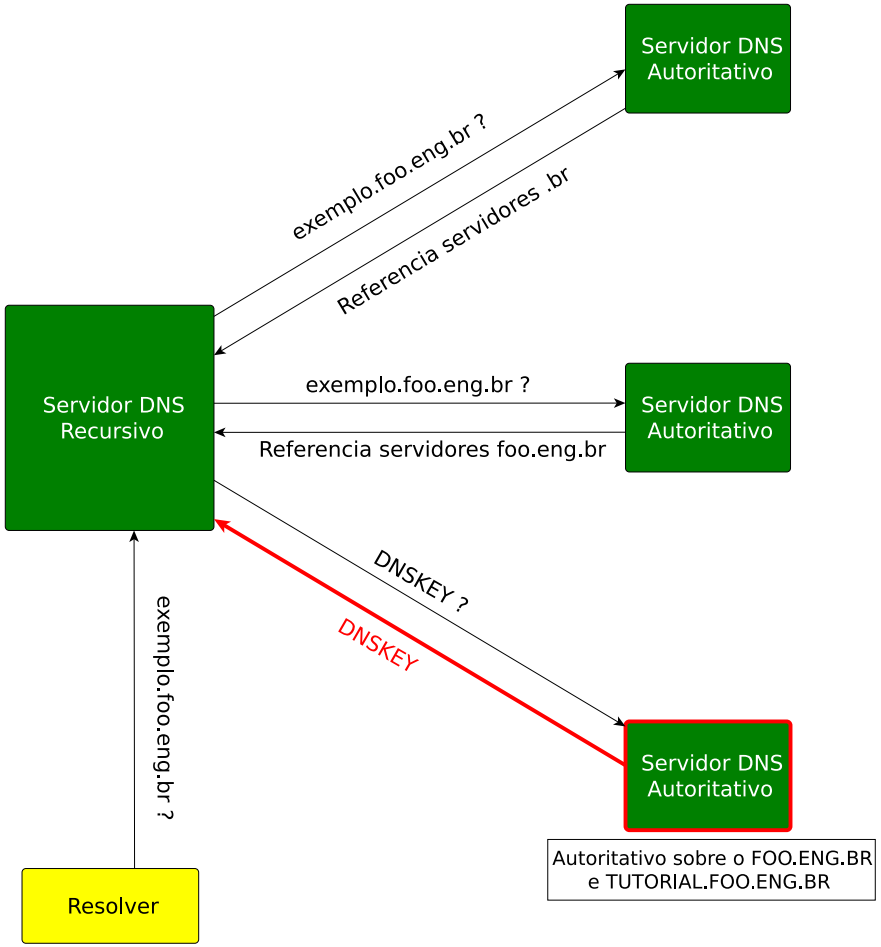


- The recursive server uses the DNSKEY to verify the signature on the DS record (RRSIG)

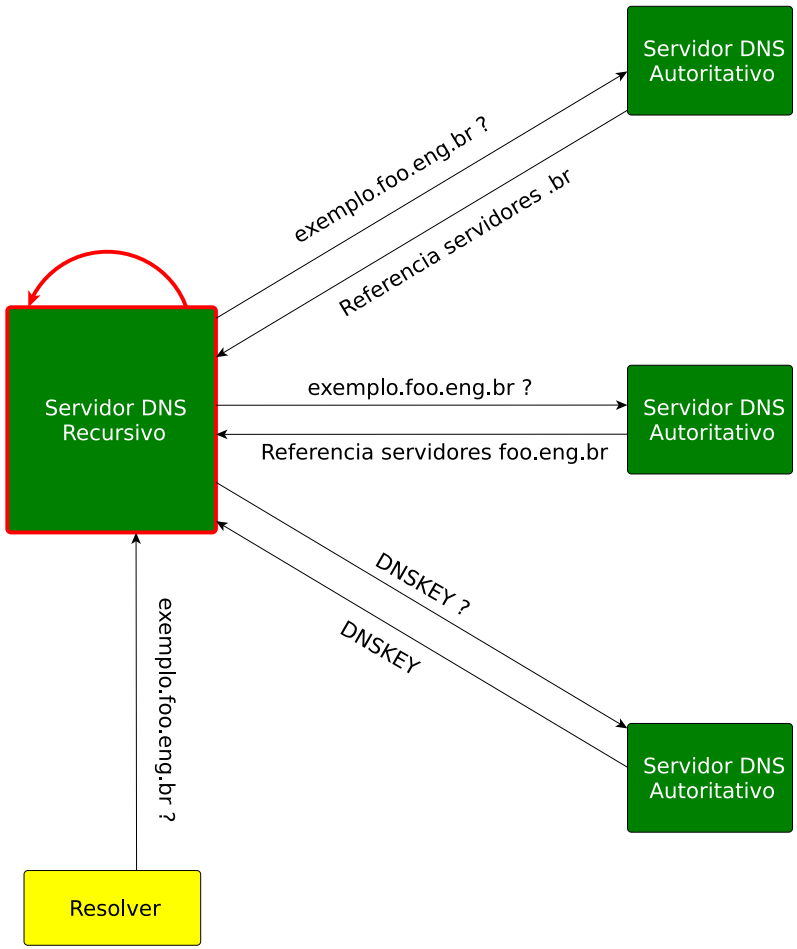
Example


Servidor DNS Autoritativo
exemplo.foo.eng.br ?
Referencia servidores .br
Servidor DNS Recursivo
exemplo.foo.eng.br ?
Servidor DNS Autoritativo
Referencia servidores foo.eng.br
DNSKEY ?
exemplo.foo.eng.br ?
Servidor DNS Autoritativo
Autoritativo sobre o FOO.ENG.BR e TUTORIAL.FOO.ENG.BR
Resolver

Example

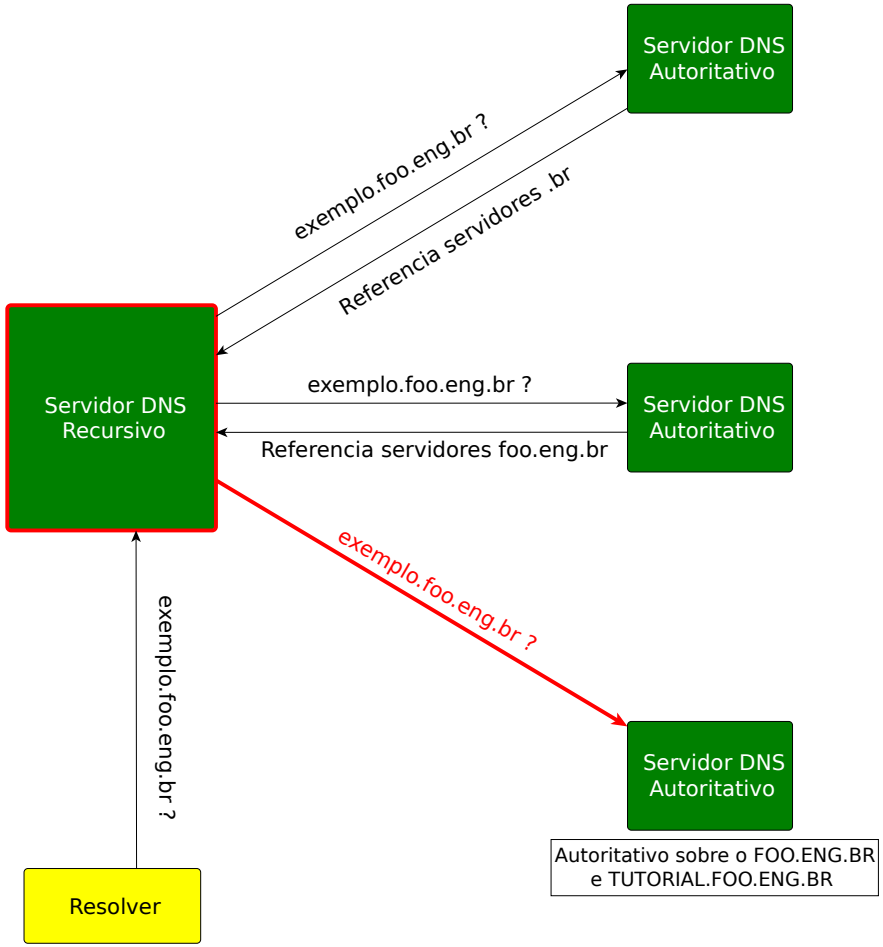

Servidor DNS Autoritativo
exemplo.foo.eng.br ?
Referencia servidores .br
Servidor DNS Recursivo
exemplo.foo.eng.br ?
Servidor DNS Autoritativo
Referencia servidores foo.eng.br
DNSKEY ?
DNSKEY
exemplo.foo.eng.br ?
Servidor DNS Autoritativo
Autoritativo sobre o FOO.ENG.BR e TUTORIAL.FOO.ENG.BR
Resolver

Example

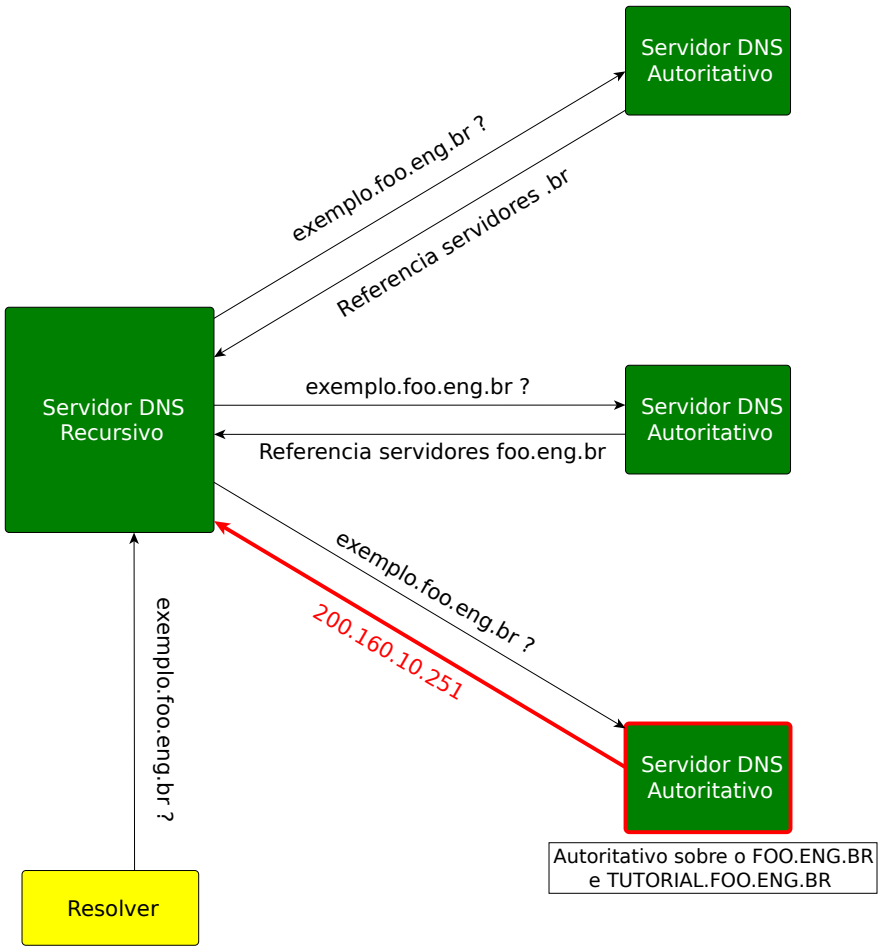


- The recursive DNS server checks if this DNS server is valid using the DS and DNSKEY.

Example

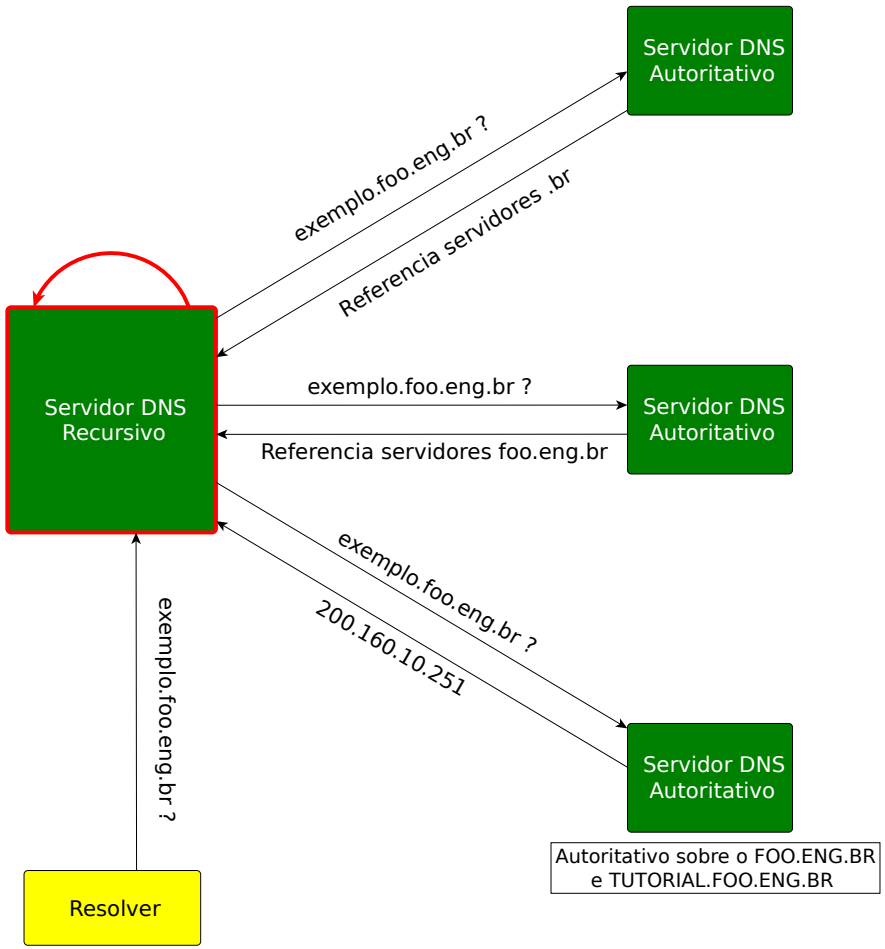

Servidor DNS Autoritativo
exemplo.foo.eng.br ?
Referencia servidores .br
Servidor DNS Recursivo
exemplo.foo.eng.br ?
Servidor DNS Autoritativo
Referencia servidores foo.eng.br
exemplo.foo.eng.br ?
exemplo.foo.eng.br ?
Servidor DNS Autoritativo
Autoritativo sobre o FOO.ENG.BR e TUTORIAL.FOO.ENG.BR
Resolver

Example



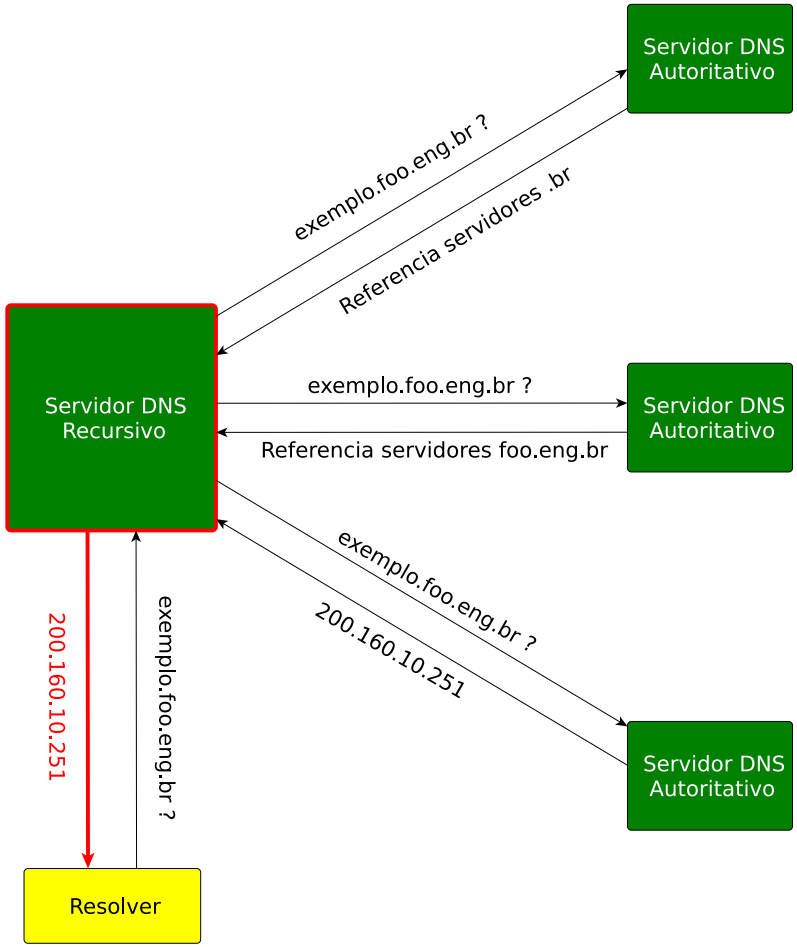
- Responds with the A record and its RRSIG.

Example



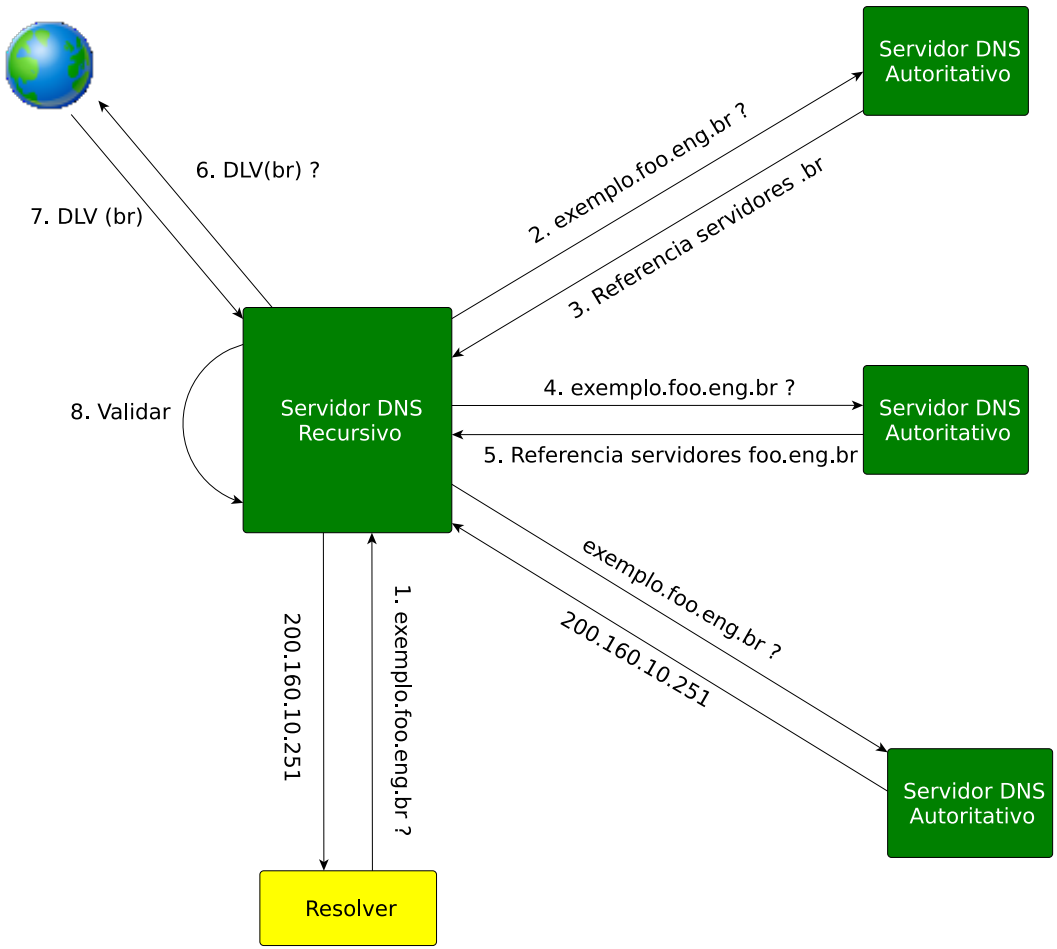
- The recursive server used the DNSKEY to verify the signature on the A record

Example


Servidor DNS Autoritativo
exemplo.foo.eng.br ?
Referencia servidores .br
Servidor DNS Recursivo
exemplo.foo.eng.br ?
Servidor DNS Autoritativo
Referencia servidores foo.eng.br
exemplo.foo.eng.br ?
200.160.10.251
200.160.10.251
exemplo.foo.eng.br ?
Servidor DNS Autoritativo
Resolver

DLV – DNSSEC Lookaside Validation

Example of an key that is not anchored



- Allows for the domain to use DNSSEC without the need for its parent to do so

RFC 4431

The DNSSEC Lookaside Validation (DLV) DNS Resource Record

EDNS0 (RFC 2671)

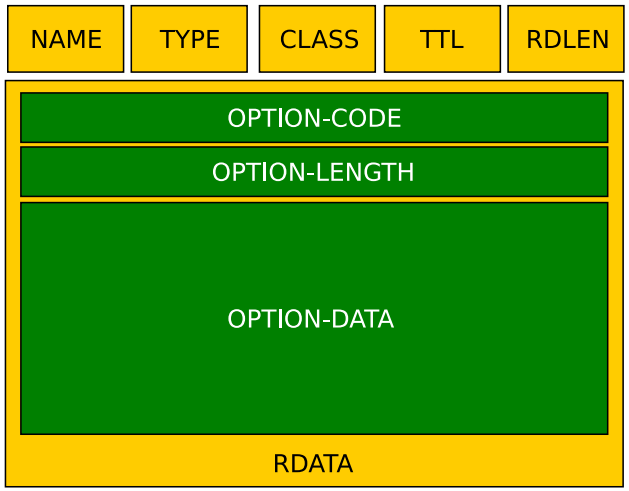
Created in order to make the limitations on the fields on the DNS protocol more flexible

- Lets the querier inform the maximum capacity of their UDP frames. Eliminates the limitation of 512 bytes UDP frames.
- Increases the number of flags and expands some existing fields to allow for a larger diversity of values
- Structured in a way that makes expansion of the protocol possible
- Indicates the support of DNSSEC on servers by the new flag DO (DNSSEC OK)

Reminder

It is necessary that TCP is also enabled on the server.

EDNS0




```
NAME        ‘‘.’’
TYPE        ‘‘OPT’’
CLASS       UDP frame size
TTL         Extended RCODE and Flags
RDATA       Pairs {Attribute, Value}
```

OPT Pseudo-RR

- Stores information regarding the EDNS0
- Not stored in the file system, since it's used only in the moment of communication between servers

registro.br
Registro de Domínios para a Internet no Brasil

Firewall Configuration

The firewall must be configured to handle the reassembly of UDP fragments correctly before checking the other rules.

In case that is not possible, an alternative is to configure the recursive server to query for smaller UDP responses. If the server in use is Bind, that can be done in versions newer than 9.3.0 with the following options:

```
options {
   edns-udp-size 1252; # Recursive servers
   max-udp-size 1252; # Recursive and authoritative servers
};
```

1252 is just a suggestion. This value should be defined according to the configurations of firewall.

nic.br

Concept of Keys

registro.br
Registro de Domínios para a Internet no Brasil

Why two types of keys?

- Allow the substitution of a more frequently used key (ZSK) without the need to change the DS on the parent zone (hash on the KSK)
- Possibility of having smaller ZSKs so that signatures are smaller

Key Signing Key (KSK)

Key used only to sign the keys of the zone, that is, they sign only the DNSKEY RRSETs – **SEP bit** flag on

Zone Signing Key (ZSK)

Key used to sign all the records that are authoritative of the zone

nic.br

Concept of Keys

registro.br
Registro de Domínios para a Internet no Brasil

Reminder

- The DNSKEY record stores the public part of both the KSK and ZSK
- The RRSIG record stores the signatures for a RRSET created by a KSK or a ZSK

Concept of Keys

registro.br
Registro de Domínios para a Internet no Brasil

Reminder

- The DNSKEY record stores the public part of both the KSK and ZSK
- The RRSIG record stores the signatures for a RRSET created by a KSK or a ZSK

Working with only one key!

It is recommended that a zone has only one key.
More information can be found on page 105.

Rollover (Changing of Keys - RFC 4641)

- Introduces the procedure for periodical changes of keys
- Keeps the structure of the zone and caches consistent during the whole period in which the keys are replaced (await the TTLs to expire)

Cache

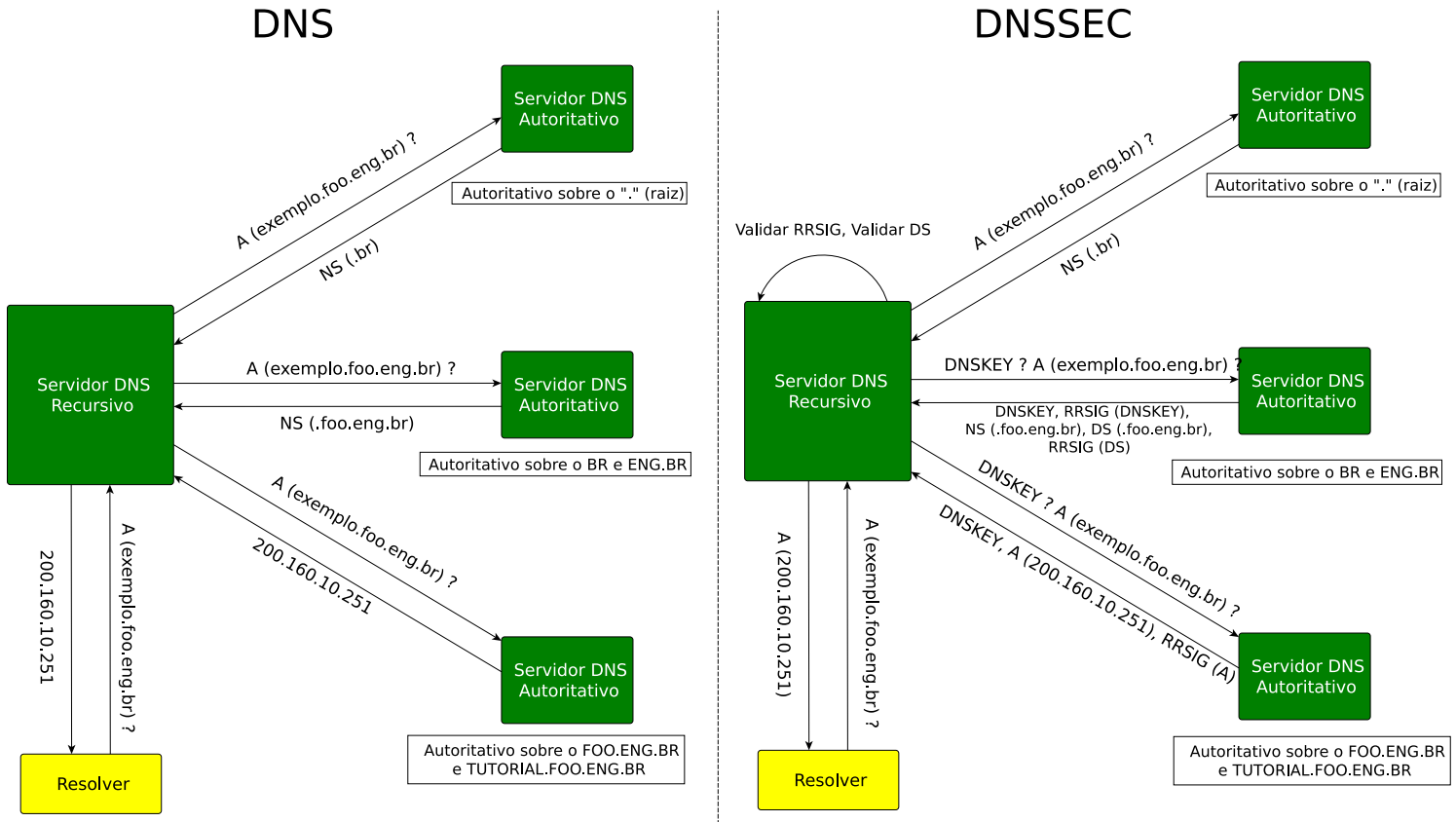
- A DNS cache keeps locally the results of queried names for future use, avoiding the need of querying for the same information, and increasing the response time
- Information on the cache is stored for a determined period of time defined in the TTL (Time To Live), which is sent on the response of every request

TTL

- The larger the TTL, the longer a piece of information will be kept in the cache. This is interesting for servers with zones that are updated rarely
- The smaller the TTL, the more traffic there will be, because recursive servers will have query for data more frequently

DNS vs DNSSEC

Differences between a DNS request and one with DNSSEC



8 Packets — X Bytes

12 Packets ± 6X Bytes[a]

[a] proportional to the size of the key

Part II

Using DNSSEC

Softwares

DNSSEC Compatibility

	Authoritative	Recursive	Caching	DNSSEC	DNSSEC bis[a]	NSEC3[b]	TSIG	IPv6
ANS	√			√	√		√	√
BIND	√	√	√	√	√	√[c]	√	√
djbdns	√	√	√					√
DNSSHIM	√				√		√	√
IPControl	√	√	√	√	√		√	√
IPM DNS	√	√	√	√	√		√	√
MaraDNS	√	√	√					?
Microsoft DNS	√	√	√	√	√[d]		√	√
NSD	√			√	√	√	√	√
PowerDNS	√	√	√					√
Unbound		√	√		√	√		√
Vantio		√	√	√	√		√	√
VitalQIP	√	√	√	√	√		?	√

[a] Current protocol version

[b] Recursive servers must support NSEC3

[c] From version 9.6.0

[d] Add support in Windows Server 2008 R2 or Windows 7 versions

Softwares

Compatibility between DNS servers and Operational Systems

	BSD[a]	Solaris	Linux	Windows	MAC OS X
ANS	√	√	√	?	?
BIND	√	√	√	√	√
djbdns	√	√	√		√
DNSSHIM	√	√	√	√	√
IPControl		√	√	√	
IPM DNS	√	√	√		√
MaraDNS	√	√	√	√[b]	√
Microsoft DNS				√	
NSD	√	√	√		√
PowerDNS	√	√	√	√	√[c]
Unbound	√	√	√		√
Vantio	√	√	√	?	?
VitalQIP		√	√	√	

[a] System compatible with POSIX as other Unix like systems.

[b] Only recent versions of the operational system

[c] Beta version

	Owner	Open Source	Free
ANS	Nominum		
BIND	Internet System Consortium	√	√
djbdns	Daniel J. Bernstein	√	√
DNSSHIM	Registro.br	√	√
IPControl	INS		
IPM DNS	EfficientIP		
MaraDNS	Sam Trenholme	√	√
Microsoft DNS	Microsoft		
NSD	NLnet Labs	√	√
PowerDNS	PowerDNS.com / Bert Hubert	√	√
Unbound	NLnet Labs	√	√
Vantio	Nominum		
VitalQIP	Lucent Technologies		

DIG (Domain Information Groper)

DNS query tool

– for validating the chain of trust compile with the **sigchase** option enabled

DRILL

Similar DIG tool with native support to DNSSEC

– until version 1.2.0 of ldns, DRILL was **not** capable of validating CNAME records

BIND in Windows

- Download the last version of BIND on http://www.isc.org
- Extract the ZIP file and run the program BINDInstall.exe
- After the installation, go to Services (administrative tools) and start the service "ISC BIND"

Error while starting ISC BIND service

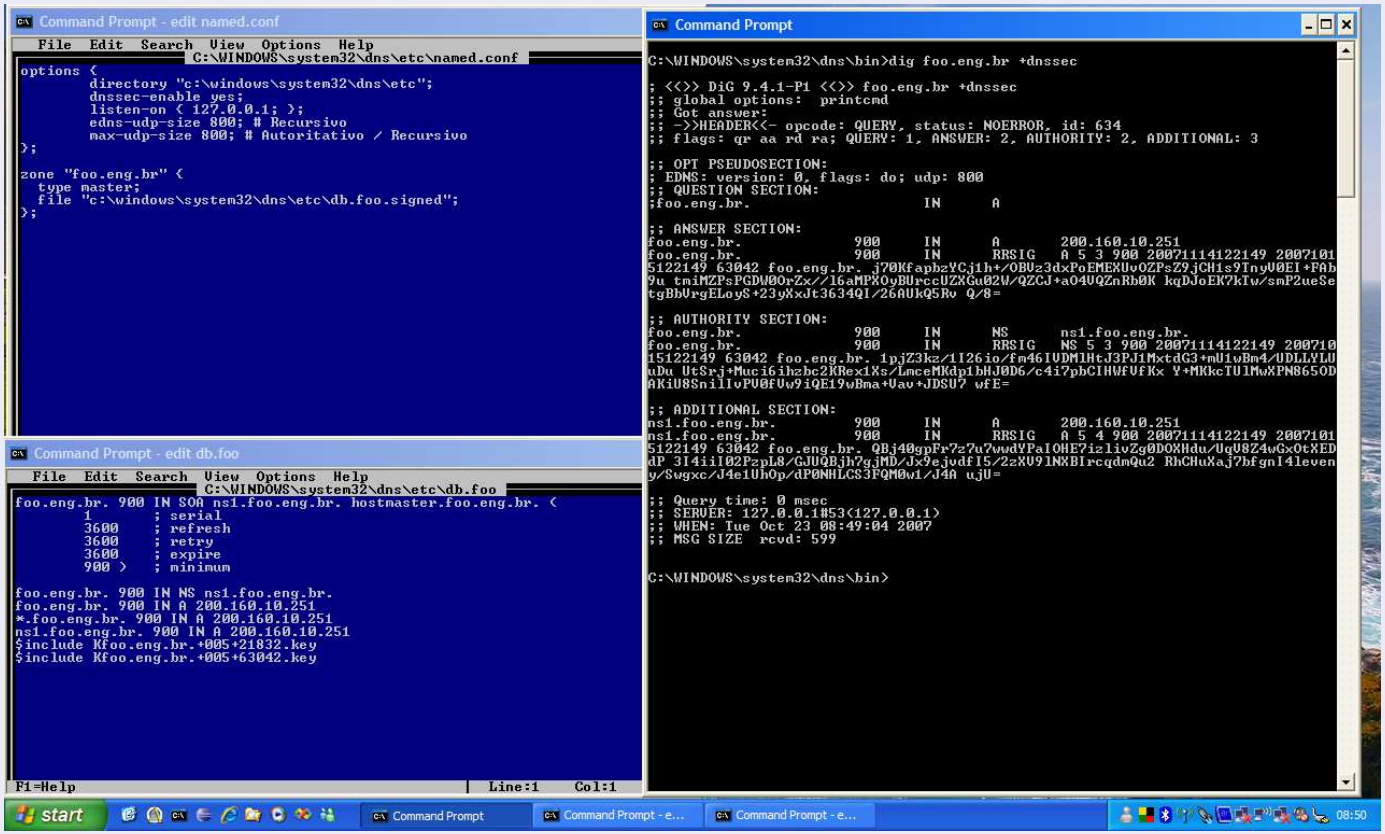
Access the service properties, on the "Log On" tab select the "Local System account" option

registro.br
Registro de Domínios para a Internet no Brasil

BIND in Windows

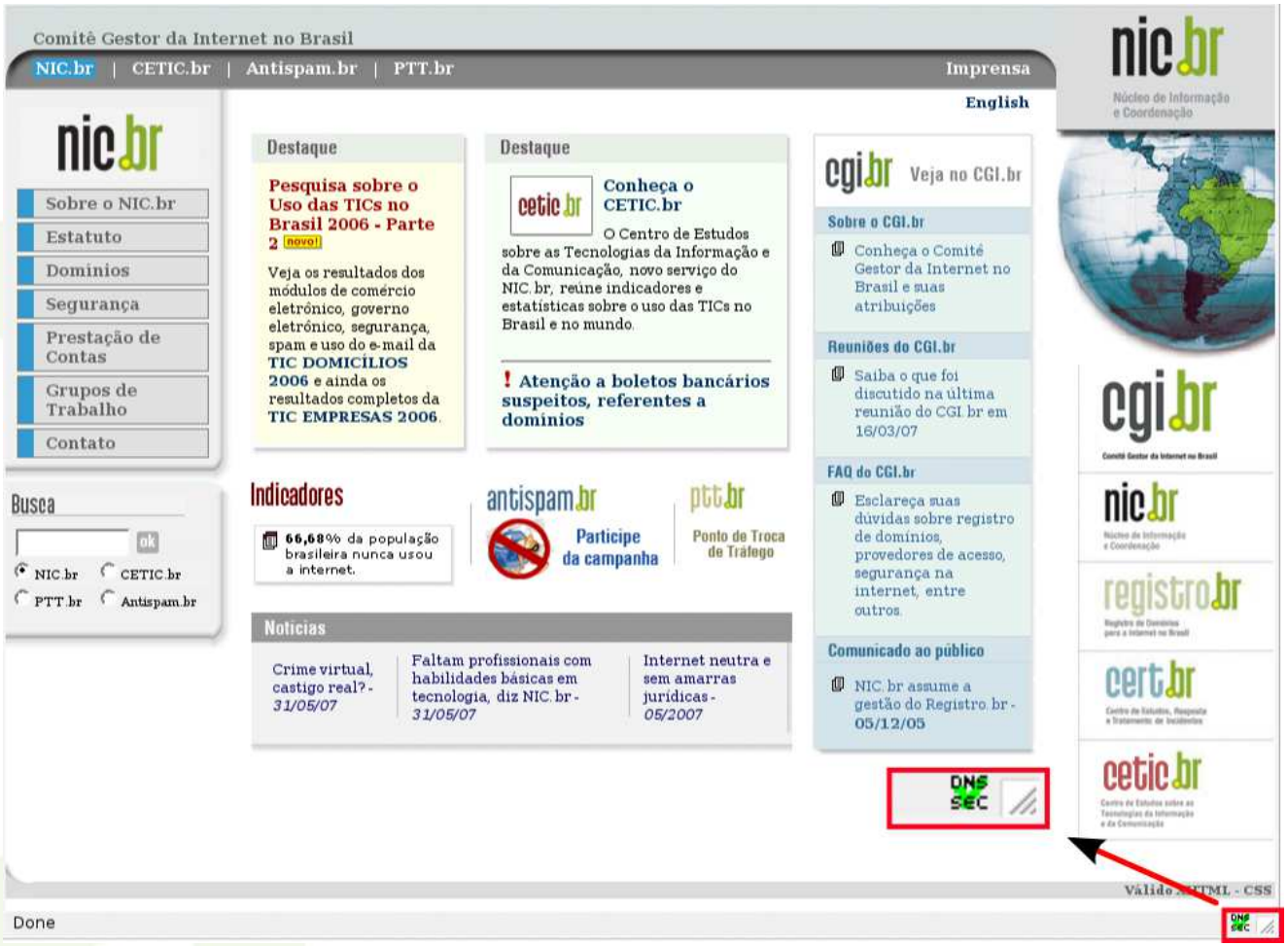
The BIND in Windows works as is in Linux, but the files are stored in different places.

- The configuration files are located in **c:\windows\system32\dns\etc**
- The executable files (named, dig) are located in **c:\windows\system32\dns\bin**

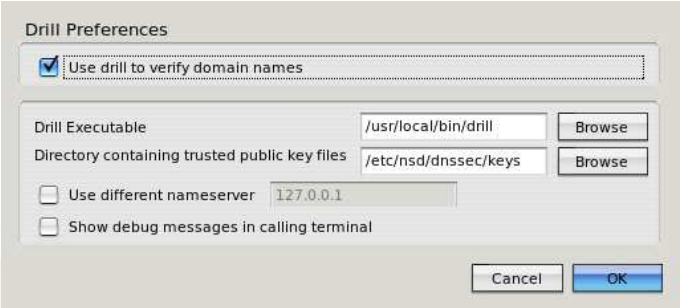


Firefox plugin that allows validating the chain of trust in case of a DNSSEC enabled domain

– http://nlnetlabs.nl/projects/drill/drill_extension.html



After installing the plugin, change the configuration setting the location of the “trusted-keys” directory.



In the “trusted-keys” directory, each file must have a key in a specific format:

```
br.  IN DNSKEY 257 3 5
AwEAAcmqkFULGgm1VlBbUYQEuCzSbEByjAcNInY9gxftbTK+CSYwlGAfxl5hwx7kxOQAZ2ZMLrxD+sTQVC4StoAPPcUhFqEGOV+9G
I6SsD/fikQOIhtXaS6INKkOPOkfBqotk6C5QbbVXcsML54/dtZYWi/Z7CaG2Hz93ouyUMQzIPohER+gkbFYq3ewDajqJKNsVm8caT
9/mkNw+CQHR+QNmWM=
```

http://registro.br/dnsshim/

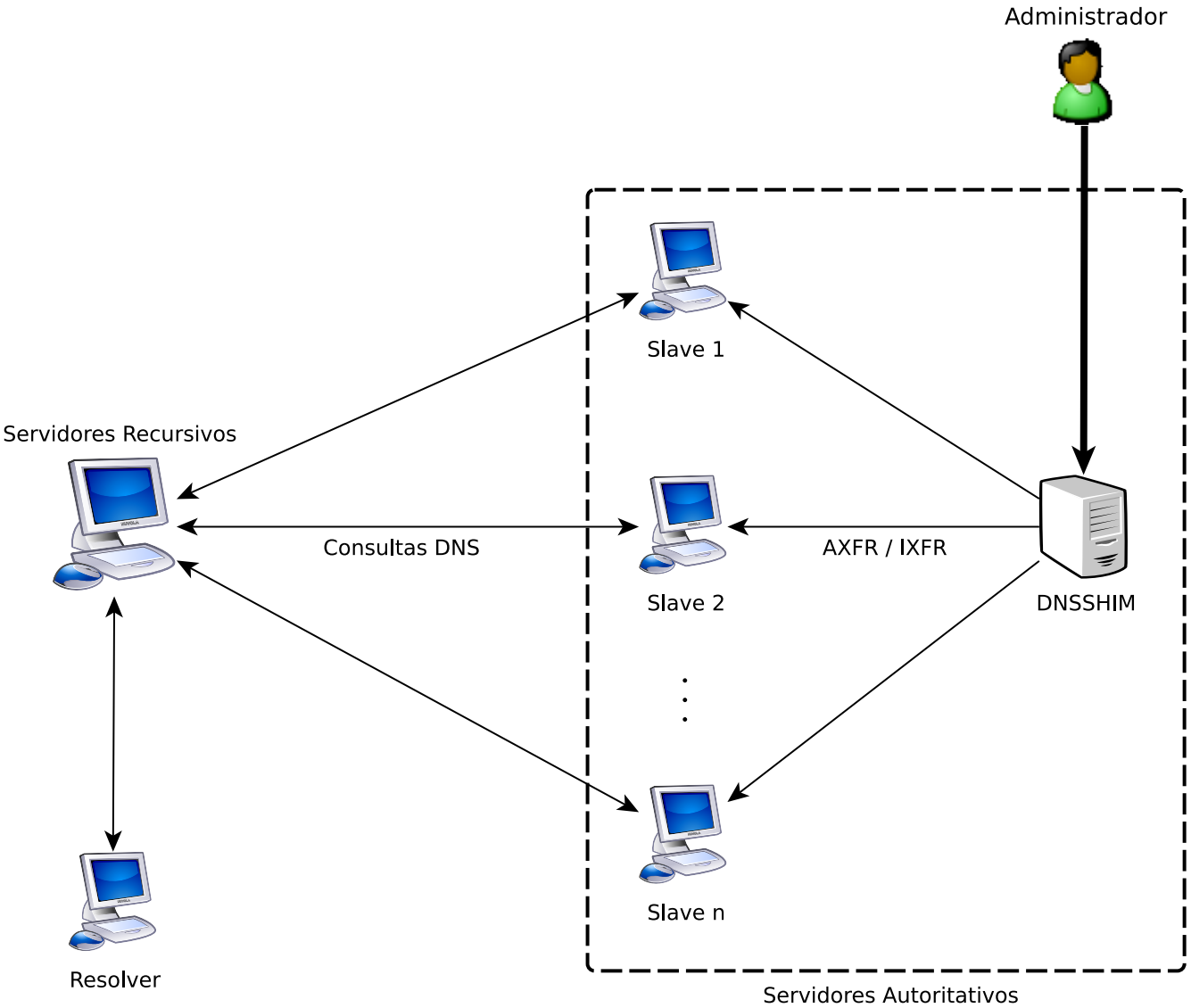
- Open-Source
- Automates the provisioning process of zones
- DNSSEC support
- Automated interface
- Maintenance keys / signatures

Target Public

Host provider or any other institution responsable for administrating authoritative DNS servers for many zones

Softwares

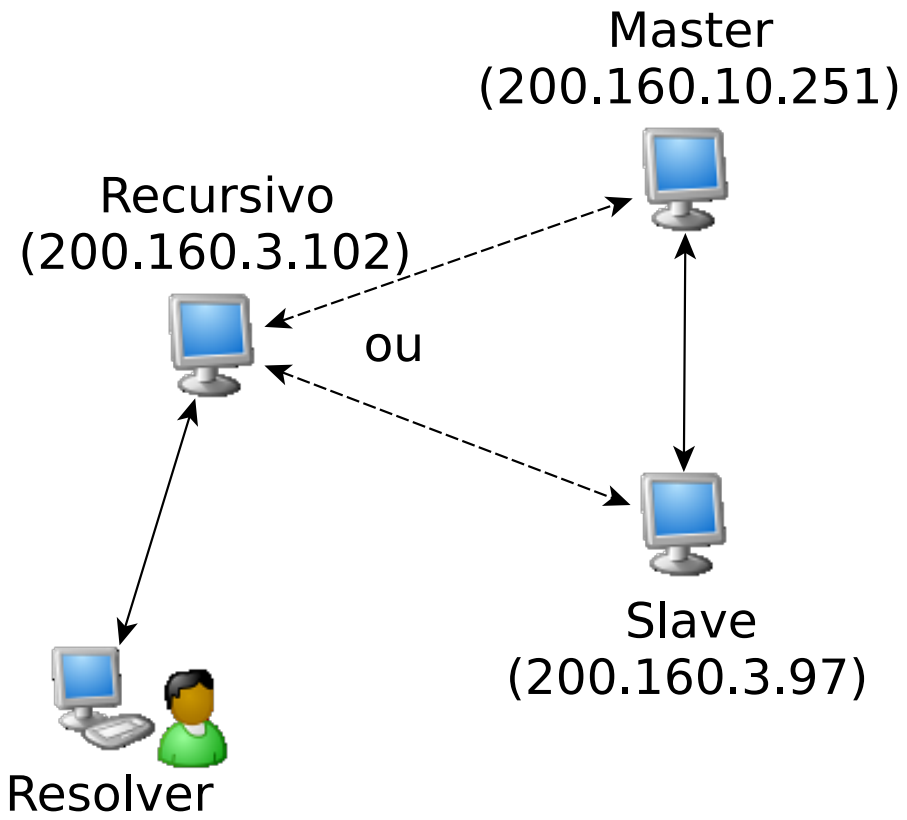
DNSSHIM - DNS Secure Hidden Master



Practical demonstration

1. BIND configuration
   a. Zone file
   b. named.conf
2. Test – DNS record query
3. DNSSEC configuration in BIND
   a. Generate KSK key
   b. Update zone file
   c. Sign zone
   d. Update named.conf file
4. Test – DNS record query with DNSSEC
5. Test – Validating chain of trust

Test enviroment



registro.br
Registro de Domínios para a Internet no Brasil

foo.eng.br file

```
foo.eng.br. IN SOA ns1.foo.eng.br. hostmaster.foo.eng.br. (
            1       ; serial
            3600    ; refresh
            3600    ; retry
            3600    ; expire
            900 )   ; minimum
foo.eng.br.               IN NS ns1.foo.eng.br.
foo.eng.br.               IN NS ns2.foo.eng.br.
tutorial.foo.eng.br.      IN NS ns2.tutorial.foo.eng.br.
tutorial.foo.eng.br.      IN NS ns1.tutorial.foo.eng.br.
ns1.foo.eng.br.           IN A 200.160.3.97
ns2.foo.eng.br.           IN A 200.160.10.251
ns1.tutorial.foo.eng.br. IN A 200.160.3.97
ns2.tutorial.foo.eng.br. IN A 200.160.10.251
exemplo.foo.eng.br.       IN A 200.160.10.251
```

nic.br

named.conf (master)

```
options {
    directory "/etc/namedb";
    pid-file "/var/run/named/pid";
    dump-file "/var/dump/named_dump.db";
    statistics-file "/var/stats/named.stats";
    listen-on { 200.160.10.251; };
};
zone "foo.eng.br" {
    type master;
    file "/etc/namedb/foo.eng.br";
    allow-transfer {
        200.160.3.97;
    };
};
zone "tutorial.foo.eng.br" {
    type master;
    file "/etc/namedb/tutorial.foo.eng.br";
    allow-transfer {
        200.160.3.97;
    };
};
```

named.conf (slave)

```
options {
    directory "/etc/namedb";
    pid-file "/var/run/named/pid";
    dump-file "/var/dump/named_dump.db";
    statistics-file "/var/stats/named.stats";
    listen-on { 200.160.3.97; };
};
zone "foo.eng.br" {
    type slave;
    file "/etc/namedb/foo.eng.br";
    masters {
        200.160.10.251;
    };
};
zone "tutorial.foo.eng.br" {
    type slave;
    file "/etc/namedb/tutorial.foo.eng.br";
    masters {
        200.160.10.251;
    };
};
```

Test - DNS record query

registro.br
Registro de Domínios para a Internet no Brasil

Zona foo.eng.br

dig @200.160.10.251 foo.eng.br soa +noadditional +multiline

```
; <<>> DiG 9.3.3 <<>> @200.160.10.251 foo.eng.br soa +noadditional +multiline
; (1 server found)
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 40573
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 2
;; QUESTION SECTION:
;foo.eng.br.             IN SOA
;; ANSWER SECTION:
foo.eng.br.             900 IN SOA ns1.foo.eng.br. hostmaster.foo.eng.br. (
                                1          ; serial
                                3600       ; refresh (1 hour)
                                3600       ; retry (1 hour)
                                3600       ; expire (1 hour)
                                900        ; minimum (15 minutes)
                                )
;; AUTHORITY SECTION:
foo.eng.br.             900 IN NS ns1.foo.eng.br.
foo.eng.br.             900 IN NS ns2.foo.eng.br.
;; Query time: 1 msec
;; SERVER: 200.160.10.251#53(200.160.10.251)
;; WHEN: Wed May 23 16:05:56 2007
;; MSG SIZE  rcvd: 143
```

nic.br

KSK key generation

registro.br
Registro de Domínios para a Internet no Brasil

BIND: dnssec-keygen

Zona foo.eng.br:

```
dnssec-keygen -f KSK -a RSASHA1 -b 2048 -n ZONE foo.eng.br
```

Where,

- -f : Type of the key
- -a : Algorithm
- -b : Key size (bits)
- -n : Key owner type
- -r : Randomic device

In some enviroments, when the key generation process takes a long time, it may be necessary to use a randomic device, like: "`-r /dev/urandom`"

PS1: Keep the name of the generated keys, remembering wich one is the KSK and wich one is the ZSK for futher use.

PS2: Keys generated using dnssec-keygen don't have passphrase.

Key size examples

- KSK BR: 1280 bits

Keys file example

Public key (.key)

```
foo.eng.br. IN DNSKEY 257 3 5 AwEAAdDaICi4nCQX+dC+kkGlGmi7+Pjww4O5WYZtt+oe1RG329H2+k0Y XhYiZx7tLULD8Fn3DtBC
hGTeFND+gCBj0vFS9MEjxHIkD2gtt3fFIbqN /sQIHDjNGr1M6aFngKxWTENWqkl71hT9jOEvzsLOD+deFDge4sDF5qOQ 4D8njiqIIqDsU
kt3I1cJoFtP9k9RPIijxWdILWuKgh7nEvKpX7eOEuXO YK1W88Av9ctpm3y6lzbsWC0K4OIl7nGTB+qMCbt/ZdYMwcaVuTBHQpEUKNVuq3m
FGj1MxwtadBimmqq+YhleGzn21x0CYmsStwNUAWcb/H9Sqg0G F3CVcH0t86k=
```

Private key (.private)

```
Private-key-format:  v1.2
Algorithm:  5 (RSASHA1)
Modulus: 0NogKLicJBf50L6SQaUaaLv4+PDDg7lZhm236h7VEbfb0fb6TRheFiJnHu0tQsPwWfcO0EKEZN4U0P6AIGPS8VL0wSPEciQPaC
23d8Uhuo3+xAgcOM0avUzpoWeArFZMQ1aqSXvWFP2M4S/Ows4P514UOB7iwMXmo5DgPyeOKogioOxSS3cjVwmgW0/2T1E8iKPFZ0gta4qCH
ucS8qlft44S5c5grVbzwC/1y2mbfLqXNuxYLQrg4iXucZMH6owJu39l1gzBxpW5MEdCkRQo1W6reYUaPUzHC1p0GKaaqr5iGV4bOfbXHQJi
axK3A1QBZxv8f1KqDQYXcJVwfS3zqQ==

...
```

Sign zones

When signing the zone the RRSIG and NSEC records generated are sorted inside the zone file using the canonical form

BIND: dnssec-signzone

Zona foo.eng.br:

```
$ dnssec-signzone -S -z foo.eng.br
```

Where,

- -S: smart signing: automatically finds key files for the zone and determines how they are to be used
- -z : Ignore the KSK key bit SEP and sign all the zone
- -e : Expiration date of the signatures (format AAAAMMDDHHMMSS) - Default is 30 days
- the last parameter refers to *zone file*

DS records generation

A file containing the DS records, used in the delegations, is created when the zone is signed.
– the generated file in this example: `dsset-foo.eng.br.`

Adding DS records in delegations

If there's a delegated zone using DNSSEC inside your domain, the DS record of this zone must be added in foo.eng.br file to continue the chain of trust beteween the domain (foo.eng.br) and your delegations.

SHA1 `tutorial.foo.eng.br. IN DS 3112 5 1 386B4390C5B30DB65D74EA8B660978077171948C`

SHA256 `tutorial.foo.eng.br. IN DS 3112 5 2 19602F6089F8877E037AA077B8376F30869E261EB55460F2A74E32AD1424F53A`

```
foo.eng.br IN SOA ns1.foo.eng.br. hostmaster.foo.eng.br. (
                3          ; serial
                3600       ; refresh (1 hour)
                3600       ; retry (1 hour)
                3600       ; expire (1 hour)
                900        ; minimum (15 minutes)
                )
foo.eng.br.           IN NS ns1.foo.eng.br.
foo.eng.br.           IN NS ns2.foo.eng.br.
...
tutorial.foo.eng.br.    IN DS 3112 5 1 386B4390C5B30DB65D74EA8B660978077171948C
```

PS

The zone foo.eng.br must be re-signed after adding the DS record into the zone

Update named.conf file

Must be updated in Master

Change the path to zone file

```
zone "foo.eng.br" {
    type master;
    file "/etc/namedb/foo.eng.br.signed";
    allow-transfer {
        200.160.3.97;
    };
};
```

Adding DLV key as a trusted-key (Recursive)

1. Get the DLV public key on http://ftp.isc.org/www/dlv/dlv.isc.org.named.conf
2. Add the public key at the trusted-key block in named.conf file
3. Add "dnssec-lookaside . trust-anchor dlv.isc.org.;" to the options block in named.conf file

Adding a DLV key example

```
options {
    ...
    dnssec-enable yes; # only on BIND 9.3
    dnssec-validation yes; # BIND 9.4 or greater
    dnssec-lookaside .  trust-anchor dlv.isc.org.;
};
trusted-keys {
...
dlv.isc.org.  257 3 5
    "BEAAAAPp1USu3BecNerrrd78zxJIslqFaJ9csRkxd9LCMzvk9Z0wFzoF
    kWAHMmMhWFpSLjPLX8UL6zDg85XE55hzqJKoKJndRqtncUwHkjh6zERN
    uymtKZSCZvkg5mG6Q9YORkcfkQD2GIRxGwx9BW7y3ZhyEf7ht/jEh01N
    ibG/uAhj4qkzBM6mgAhSGuaKdDdo40vMrwdv0CHJ74JYnYqU+vsTxEIw
    c/u+5VdA0+ZOA1+X3yk1qscxHC24ewPoiASE7XlzFqIyuKDl0cFySchT
    Ho/UhNyDra2uAYUH1onUa7ybtdtQclmYVavMplcay4aofVtjU9NqhCtv
    f/dbAtaWguDB";

}; ...
```

Test - DNSSEC records query

registro.br
Registro de Domínios para a Internet no Brasil

foo.eng.br zone

dig @200.160.10.251 foo.eng.br soa +dnssec +noadditional +multiline

```
;; ANSWER SECTION:
foo.eng.br.             900 IN SOA ns1.foo.eng.br. hostmaster.foo.eng.br. (
                                3          ; serial
                                3600       ; refresh (1 hour)
                                3600       ; retry (1 hour)
                                3600       ; expire (1 hour)
                                900        ; minimum (15 minutes)
                                )
foo.eng.br.             900 IN RRSIG SOA 5 3 900 20070617200428 (
                                20070518200428 62745 foo.eng.br.
                                glEeCYyd/CCBfzH64y0RAQf9OxYDsI4xuBNaam+8DZQZ
                                xeoSLQEEtwmp6wBtQ7G10wSM9nEjRRhbZdNPNKJMp2PE
                                lLLgLI+BLwdlz0t8MypcpLOaTm9rc7pP7UR5XLzU1k8D
                                m6ePW1bNkId7i0IPSghyoHM7tPVdL2GW51hCujA= )
;; AUTHORITY SECTION:
foo.eng.br.             900 IN NS ns2.foo.eng.br.
foo.eng.br.             900 IN NS ns1.foo.eng.br.
foo.eng.br.             900 IN RRSIG NS 5 3 900 20070617200428 (
                                20070518200428 62745 foo.eng.br.
                                3iLm1ROC+UeqYk0xgQGQQXkBzcKiKQRPwe+1JZlpjEzj
                                U1Uj0HUOHefajXzMv7FlFMWYeU51Ybg49HFe67XQVlK5
                                4GeAFXWB7YS59yODLoNEBxQl9QEy6g/OOnLpuKTrST8q
                                qd5Fc/eYqN/Ag3GnfcAviZgiQhhveGH9mJHWZyc= )
```

nic.br

Test - Chain of trust validation

DIG - Sigchase

- The sigchase is an option that enables checks in the DNSSEC chain of trust from the signed point up to the trusted-key
- Good for cheking zone's signatures and keys
- To use DIG +sigchase, it's necessary to install OPENSSL development library[a], compile BIND 9.4 or newer with the enviroment variable "`STD_CDEFINES`" defined with the value "`-DDIG_SIGCHASE=1`" and run the command "configure" with the parameter "–with-openssl"

[a] In Ubuntu, the package libssl-dev

Adding trusted-key for DIG +sigchase

1. Get the public part of the trusted-key of the key
2. Copy this part to the file `/etc/trusted-key.key`
   don't let any line break in the key
   don't let any empty line after the end of the key

DIG file example

trusted-key.key file

```
br.                     21600 IN DNSKEY 257 3 5 AwEAAcmqkFULGgm1VlBbUYQEuCzSbEByjAcNInY9gxftbTK+CSYwlGA
fxl5hwx7kxOQAZ2ZMLrxD+sTQVC4StoAPPcUhFqEGOV+9GI6SsD/fikQ0IhtXaS6INKk0POkfBqotk6C5QbbVXcsML54/dtZYWi/Z7CaG2
Hz93ouyUMQzIPohER+gkbFYq3ewDajqJKNsVm8caT9/mkNw+CQHR+QNmWM=
```

Test - Chain of trust validation

registro.br
Registro de Domínios para a Internet no Brasil

Example

dig @200.160.3.69 foo.eng.br soa +sigchase +multiline

```
;; RRset to chase:
foo.eng.br.             83392 IN SOA dixit.foo.eng.br. fneves.registro.br. (
                              2008070301 ; serial
                              3600       ; refresh (1 hour)
                              900        ; retry (15 minutes)
                              3600000    ; expire (5 weeks 6 days 16 hours)
                              3600       ; minimum (1 hour)
                              )
;; RRSIG of the RRset to chase:
foo.eng.br.             83392 IN RRSIG SOA 5 3 86400 20080911163449 (
                              20080703163449 6928 foo.eng.br.
                              i8FmOSkMR6vp9cg4nCfVIPkulspiHlWsW5u8kxCgtL5N
                              TF4ImBks7YHMoEtL8uBux2Y5PiqPXgFnrJebbJJK7m/N
                              XV387mqWOsuUwoTlq/uHDeNNngFsPn1w5AhkxPOZ89wG
                              y5ANukInWzaTYeRk5XzUXvcDqJMeG9BrmQkZPvo= )
Launch a query to find a RRset of type DNSKEY for zone: foo.eng.br.
;; DNSKEYset that signs the RRset to chase:
foo.eng.br.             83392 IN DNSKEY 257 3 5 (
                              AwEAAfhPH7Cg3s1icCNWzAkRU6fQeIHJBibJAoluteAB
                              ZAHnIinvgO6cbFFIP8j+4URgeowBOMUYhAxcCYp1PO1H
                              UhqiL6fGsF649OfWOdR/GF5OBHnlkZdVBVVHQTglXM5a
                              7RsoGHEFKXjSZ/DdwTKA3poJtF61wvOX78UjNxcwhoYF
                              ) ; key id = 6928
```

nic.br

registro.br
Registro de Domínios para a Internet no Brasil

Example (Continue)

```
;; RRSIG of the DNSKEYset that signs the RRset to chase:
foo.eng.br.             83392 IN RRSIG DNSKEY 5 3 86400 20080911163449 (
                                20080703163449 6928 foo.eng.br.
                                GRc+rmJcsx6OGBsCHcIdAlUXMqTGdIChIW5g0/Ek5V4v
                                8MBIg3mxVSn6dNyByqxjfXt30cew5oyYJU6TaIwXvAL/
                                I8pQpcYprHQwp9uJBoDb/8wLHvkldklmHz/VZCWvHxAz
                                /KDbMnt8RxUwXBMwQMrTBgcjDg2PW7tAgmE21to= )
Launch a query to find a RRset of type DS for zone: foo.eng.br.
;; DSset of the DNSKEYset
foo.eng.br.             18592 IN DS 6928 5 1 (
                                CA7D9EE79CC37D8DC8011F33D330436DF76220D1 )
;; RRSIG of the DSset of the DNSKEYset
foo.eng.br.             18592 IN RRSIG DS 5 3 86400 20080721050001 (
                                20080714050001 33094 eng.br.
                                zQ4S0GiF25+1NTMZh0BT8Aveg4L6SJnyris9jKvBuz7O
                                +rtVkSuGnxHuVHjdoOHj/Ohpsg4xyUFb9tujlonJdtX8
                                qS3RK0C6pD59/q91UBiTI659Px3BK6cZGb7Lvc11lnna
                                /1Xj4t4/RG3L+Ft8PyNl5lFad8/ViPHFnmnsv30= )
```

nic.br

Test - Chain of trust validation

registro.br
Registro de Domínios

Example (Continue)

```
;; WE HAVE MATERIAL, WE NOW DO VALIDATION
;; VERIFYING SOA RRset for foo.eng.br. with DNSKEY:6928: success
;; OK We found DNSKEY (or more) to validate the RRset
;; Now, we are going to validate this DNSKEY by the DS
;; OK a DS valids a DNSKEY in the RRset
;; Now verify that this DNSKEY validates the DNSKEY RRset
;; VERIFYING DNSKEY RRset for foo.eng.br. with DNSKEY:6928: success
;; OK this DNSKEY (validated by the DS) validates the RRset of the DNSKEYs, thus the DNSKEY validates the
RRset
;; Now, we want to validate the DS :  recursive call
Launch a query to find a RRset of type DNSKEY for zone: eng.br.
;; DNSKEYset that signs the RRset to chase:
eng.br.   18592 IN DNSKEY 257 3 5 (
    AwEAAc37RO7W/9o86Cx2qwp1uTSXbkag9k0fW977zHxE
    9l6LIKgxq3d8Sj/n4esUpVkRev8HloRqy3l7Yh+K0OmG
    0UiEzfAoYiR6dA6trfy+ogBzXLOC2QqgYZVDmW0xtzX0
    WIvuZQpRvC6xU/lCb4AJxaekGT9iDql7DGZo0ry/T2jl
    ) ; key id = 33094
;; RRSIG of the DNSKEYset that signs the RRset to chase:
eng.br.   18592 IN RRSIG DNSKEY 5 2 21600 20080721050001 (
    20080714050001 33094 eng.br.
    LGSiquTFsPOaU/2o8tRyQEzbmOHuD6jCC7r3Hk5DB61p
    HeDds/JKkaURDf/hWDEjmj/C2JIO7JHojkS171hYCHY1
    P45D6S75Qg9T5Bwx1R3F3KvBfgT7jMlejeuD2nslMB77
    8vp+e2RAjCGJInJ0ugoszZoK7L4a/5VoP+ZLxio= )
Launch a query to find a RRset of type DS for zone: eng.br.
;; DSset of the DNSKEYset
eng.br.   4970 IN DS 33094 5 1 (
    C9B752615F450C0231762D263A8A704BFBEEBABD )
```

Test - Chain of trust validation

registro.br
Registro de Domínios

Example (Continue)

```
;; WE HAVE MATERIAL, WE NOW DO VALIDATION
;; RRSIG of the DSset of the DNSKEYset
eng.br.   4970 IN RRSIG DS 5 2 86400 20080721050001 (
    20080714050001 63183 br.
    Ob362cVMBK00WBq2sb6tS9aD6/j7GuJTWSEOg5XDk/SI
    ndP/szpxdZVCcRgPocssmTxt7BIXWLIs0JPqA4YQQSz8
    ZK84SJLQnzVs7J/ruZG4vUsS9/Xy8Ha810d6Y5P5/4BL
    USviI5zPoV41j0HTrVGDblUGnttETsg9+fb+WSsg7Y+/
    9p3UcKPV9n6XxPbW )
;; WE HAVE MATERIAL, WE NOW DO VALIDATION
;; VERIFYING DS RRset for foo.eng.br. with DNSKEY:33094: success
;; OK We found DNSKEY (or more) to validate the RRset
;; Now, we are going to validate this DNSKEY by the DS
;; OK a DS valids a DNSKEY in the RRset
;; Now verify that this DNSKEY validates the DNSKEY RRset
;; VERIFYING DNSKEY RRset for eng.br. with DNSKEY:33094: success
;; OK this DNSKEY (validated by the DS) validates the RRset of the DNSKEYs, thus the DNSKEY validates the
RRset
;; Now, we want to validate the DS :  recursive call
Launch a query to find a RRset of type DNSKEY for zone: br.
;; Truncated, retrying in TCP mode.
;; DNSKEYset that signs the RRset to chase:
br.   18592 IN DNSKEY 257 3 5 (
    AwEAAdDoVnG9CyHbPUL2rTnE22uN66gQCrUW5W0NTXJB
    NmpZXP27w7PMNpyw3XCFQWP/XsT0pdzeEGJ400kdbbPq
    Xr2lnmEtWMjj3Z/ejR8mZbJ/6OWJQ0k/2YOyo6Tiab1N
    GbGfs513y6dy1hOFpz+peZzGsCmcaCsTAv+DP/wmm+hN
    x94QqhVx0bmFUiCVUFKU3TS1GP415eykXvYDjNpy6AM=
    ) ; key id = 18457
```

Test - Chain of trust validation

registro.br
Registro de Domínios

Example (Continue)

```
   18592 IN DNSKEY 256 3 5 (
    AwEAAae2jeyarOoqyrNopuBMIJ9zk4JSxQm4fOMc/InK
    bqqAF268sVb+pZoZScMAyIZADj31w1Xc7F/8qQQOWWIe
    YpQ8YpsORVqraKiiLiAO+S7GUBbNACebsyKO4ESAp3qU
    R63i7jHDIjwBNwvJo/GRMjoNoQlmUgdbKis+wxnGeWgY
    HLOOkPhPXqnfUn8ohvOK9Q==
    ) ; key id = 63183
   18592 IN DNSKEY 257 3 5 (
    AwEAAcmqkFULGgm1VlBbUYQEuCzSbEByjAcNInY9gxft
    bTK+CSYwlGAfxl5hwx7kxOQAZ2ZMLrxD+sTQVC4StoAP
    PcUhFqEGOV+9GI6SsD/fikQ0IhtXaS6INKkOPOkfBqot
    k6C5QbbVXcsML54/dtZYWi/Z7CaG2Hz93ouyUMQzIPoh
    ER+gkbFYq3ewDajqJKNsVm8caT9/mkNw+CQHR+QNmWM=
    ) ; key id = 61207
;; RRSIG of the DNSKEYset that signs the RRset to chase:
br.   18592 IN RRSIG DNSKEY 5 1 21600 20080721050001 (
    20080714050001 63183 br.
    Oyg8s2L+Omqt6hvJqG5ICkaPeJPH/S/+Uub6gOYfcyzo
    rO0bIY4pPL+QzzyQtiVnopR7cb9zF6+3V6h/Z17556LI
    kGfMOvYtFfneD1tLRnJCZpb5CNNOPM1RS7pweaWt0Zmb
    moQy/fmcu615farDUEBXiH690sd6LobVzFg6iarH6DV2
    1tlAahPgHuofW71w )
   18592 IN RRSIG DNSKEY 5 1 21600 20080908000000 (
    20080624173000 18457 br.
    c3pko0teEvAtb2RCkgeOpysWUT4xmCphO+x5rWcYZUwI
    uBOOeDs6BdeIvEeyCAlDvbqQf7cvzsgxzk13QoTJr6Eo
    WdHgBwH+A4HbDm+LyrbBli5zjtCyVGfoYuyUDhTfaNBu
    t9FqBkMvYMifDQmkkBoJCoVQF8u4zfJLpL8Kholn9y4D
    ArjxeVzuoDrPe2WQipP1HACzrd4DU9mG/0qRLA== )
```

registro.br
Registro de Domínios para a Internet no Brasil

Example (Continue)

```
   18592 IN RRSIG DNSKEY 5 1 21600 20080908000000 (
    20080624173000 61207 br.
    Vc6Y6INYvWEb5tUoOiMfVaWrMc18G5CYw2O+HhW32t6X
    7LZ7GZlUlZm4TGcIvYNrwWjUfPpvs1aAP3SaE16UjRZj
    1L3oI9U7yD9ekvnq2HxOudSrIODmXWulRaiBbKIPvc0T
    mEoRN+Z9SUCNdQaLKQ9BoUDUkka8jnEQ5dePyH8tNZ3e
    31SkA/AdRwVuMJi/Hzc+JRB4FKz6mEGhoHVJ6A== )
Launch a query to find a RRset of type DS for zone: br.
;; NO ANSWERS: no more
;; WARNING There is no DS for the zone: br.
;; WE HAVE MATERIAL, WE NOW DO VALIDATION
;; VERIFYING DS RRset for eng.br. with DNSKEY:63183: success
;; OK We found DNSKEY (or more) to validate the RRset
;; Ok, find a Trusted Key in the DNSKEY RRset: 61207
;; VERIFYING DNSKEY RRset for br. with DNSKEY:61207: success
;; Ok this DNSKEY is a Trusted Key, DNSSEC validation is ok: SUCCESS
```

nic.br

Guide - Authoritative server configuration

registro.br
Registro de Domínios para a Internet no Brasil

1. Install BIND in servers
2. Configure the zone files and the named.conf file in the Master server
3. Configure named.conf file in the Slave server
4. Run BIND (named) in Master and Slave servers
5. Run tests in servers (DIG)

6. Create KSK key (dnssec-keygen) (slide 103)
7. Sign the zone (dnssec-signzone) (slide 105)
8. If there're signed delegations, add the DS of this delegations in the zone file and re-sign it (slide 106)
9. Restart BIND (named) in Master and Slave servers

nic.br

Guide - Recursive server configuration

1. Install development openssl library
2. Install BIND
3. Get the public part of the trusted-key that you want
4. Add trusted-key in named.conf file
5. Run BIND (named)

Guide - Chain of Trust test

1. Install BIND with sigchase
2. Get the public part of the trusted-key that you want
3. Add the public part of the trusted-key in file /etc/trusted-key.key
4. Run tests in server (DIG +sigchase)

registro.br
Registro de Domínios para a Internet no Brasil

Authoritative Server

Re-sign the zone before the signatures expire

1. Increase the serial (SOA record) in the original zone file
2. Re-sign the zone using the command dnssec-signzone

Recursive Server

Change the trusted keys when there's a rollover in domain that you trust

nic.br

Questions?

End of the Presentation

References

References

RFC 2671
Extension Mechanisms for DNS (EDNS0)

RFC 2845
Secret Key Transaction Authentication for DNS (TSIG)

RFC 4033
DNS Security Introduction and Requirements (DNSSEC-bis)

RFC 4034
Resource Records for the DNS Security Extensions (DNSSEC-bis)

RFC 4035
Protocol Modifications for the DNS Security Extensions (DNSSEC-bis)

RFC 4431
The DNSSEC Lookaside Validation (DLV) DNS Resource Record

RFC 4470
Minimally Covering NSEC Records and DNSSEC On-line Signing

RFC 4641
DNSSEC Operational Practices

RFC 5155
DNSSEC Hashed Authenticated Denial of Existence

References

- DNSSEC.NET
  http://www.dnssec.net
- DNSSHIM
  http://www.registro.br/dnsshim
- Wikipedia - DNSSEC
  http://pt.wikipedia.org/wiki/DNSSEC
- Wikipedia - Comparison of DNS server softwares
  http://en.wikipedia.org/wiki/Comparison_of_DNS_server_software
- Firewalls e DNS, como e porque configurar corretamente
  ftp://ftp.registro.br/pub/doc/dns-fw.pdf
- Recomendações para Evitar o Abuso de Servidores DNS Recursivos Abertos
  http://www.cert.br/docs/whitepapers/dns-recursivo-aberto
- FAQ - Registro.br
  http://registro.br/faq
- Last version of this tutorial can be found at
  ftp://ftp.registro.br/pub/doc/tutorial-dnssec.pdf
- DNSSEC – Olaf Kolkman (RIPE NCC/NLnet Labs)
  http://www.nlnetlabs.nl/dnssec_howto

Thank you!