DNS система за именуване

Процес на резолвинг на
имената по IPv4 и IPv6.

DNS , защо ни трябва?

През ноември 1983 са публикувани RFC-та, които дефинират Domain Name System (DNS): RFC 882 и RFC 883.

Защо: Чрез IP адресите се адресират компютри и интерфейси в Мрежата.

Но те са числа (10-ни и 16-ни) - трудно се запомнят.

Затова се въвежда система за именуване – **DNS**.

Domain Name System

Domain Name System (DNS) е йерархична разпределена база от данни.

Тя съхранява информация за съотвтствието между Internet хост имена и IP адреси и обратно, информация за маршрутизиране на ел. поща и др. данни, използвани от Internet приложения.

Клиентите търсят информация в DNS, извиквайки ***resolver library***, която изпраща заявки до един от сървърите за имена (***name servers***) и интерпретира отговорите.

BIND софтуерът съдържа сървър за имена **named**, и две библиотеки - *resolver libraries*: **liblwres** и **libbind**.

ISC BIND

BIND (Berkeley Internet Name Domain) е реализация на DNS протоколите и осигурява отворена система за редистрибуция на основните компоненти на Domain Name System:

- Domain Name System server (процесът **named**);
- Domain Name System resolver library;
- средства за верифициране на операциите на DNS server.

https://www.isc.org/downloads/bind/

Домейни и имена на домейни

Данните, съхранени в DNS са *domain names,* организирани в дървовидна структура.

Всеки възел в дървото се нарича *domain и му се дава етикет*.

Името на домейна във възела е поредица от етикети, показващи пътя от възела до корена (*root)*.

В писмена форма се представя като низ от етикети, от дясно наляво, разделени с точки.

Написано на C.

Домейни и имена на домейни

Домейните представляват области от имена. Домейните са от първо, второ и трето ниво.

(Ако не се брои root.)

Няма пречки да има домейни от четвърто ниво, но те почти не се използват.

Основният домейн е така наречения **root** домейн. Той няма име и е един единствен. Представя се с точка (.).

Домейни и поддомейни

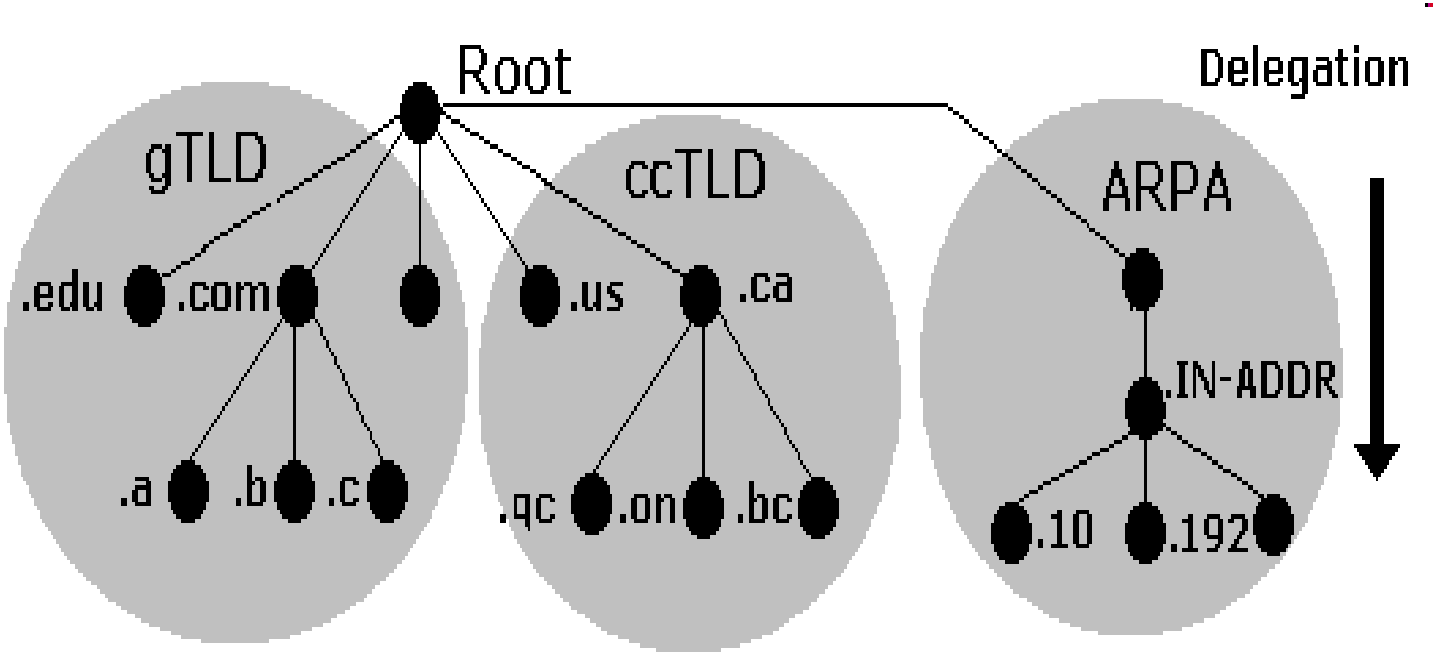
Под него се нареждат домейните от първо ниво, **top-level domain** (**TLD**).

Управлението на TLDs е делегирано на различни организации от страна на ICANN, която менижира IANA, и е отговорна за DNS root зоната.

Най-често използвани TLDs са:

generic top-level domains (gTLD) – отворени за регистрация за всеки по света, например: com, net, org, biz и др.

DNS йерархия



Домейни и поддомейни

В началото всички те са в САЩ, но после в тях влизат още много имена на обекти извън САЩ, те нарастват твърде много.

Затова се въвежда друга голяма група от домейни на първо ниво, свързани с географското разположение по държави – uk, de, bg и др. Това са country-code top-level domains (ccTLD), показващи принадлежност към държава. Състоят се само от две букви. В повечето случаи съвпадат с кода на страната по ISO 3166.

infrastructure top-level domain: Има само една TLD - Address and Routing Parameter Area (ARPA). Управлява се от IANA и има отношение към обратния резолвинг - от IP към име.

Цялостното име, което включва домейните и обекта се нарича **URL** (uniform resource locator). Пример за URL е

http://www.fmi.uni-sofia.bg

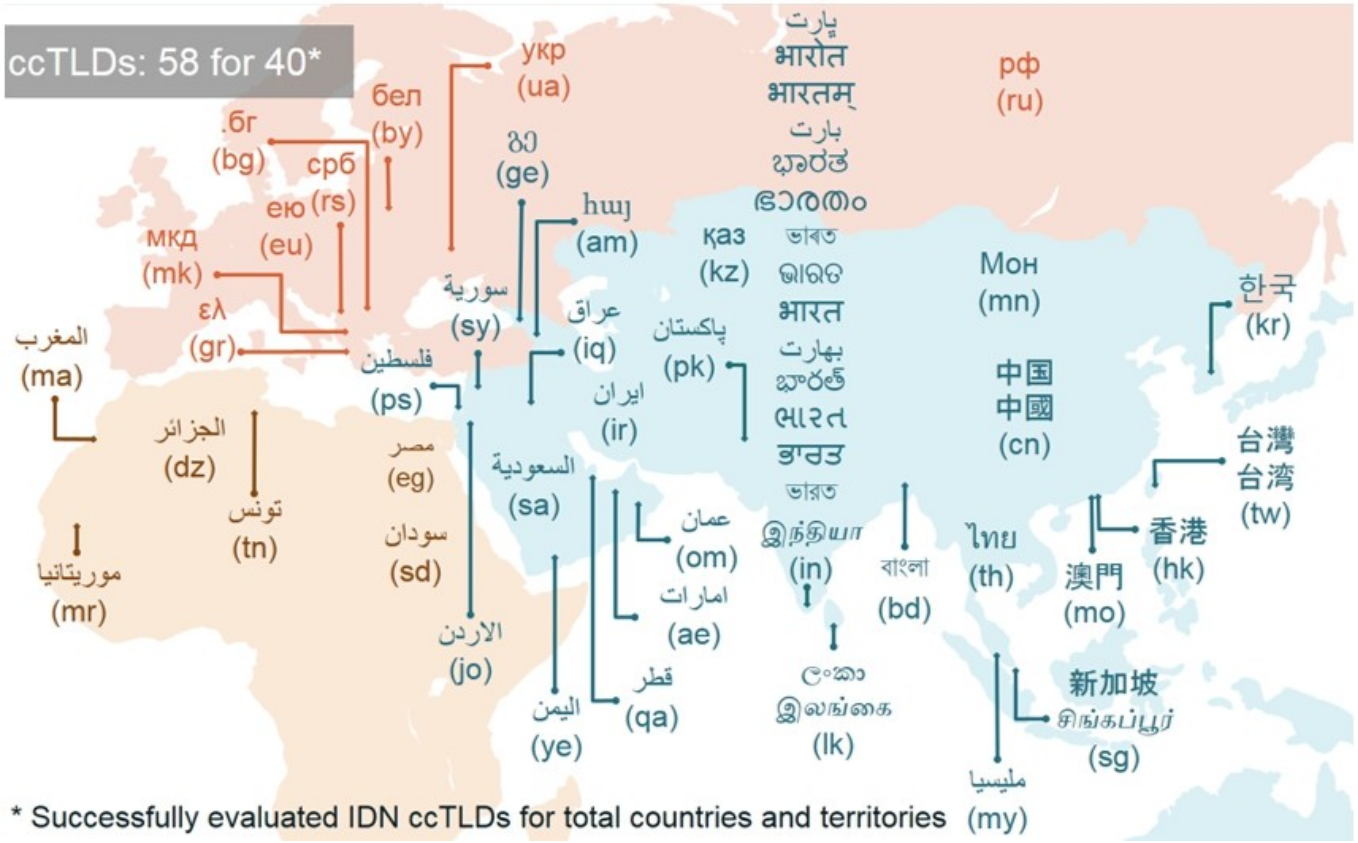
Internationalized Domain Names

До скоро имаше ограничение домейн имената да са с US-ASCII (латински) букви – LDH (letter, digit, hyphen).

Това се промени с въвеждането на Internationalized Domain Names (IDNs): скриптове, които позволяват достъп до ccTLDs и gTLDs на над 100 различни езици (включително и Български).

Те се кодират по многобайтовия Unicode стандарт и се прилагат по правилата на IDN протоколи.

IDN ccTLD стрингове, одобрени от ICANN



IDNs чрез Punycode

IDNs се записват в DNS като ASCII низове (strings), прилагайки Punycode транскрипция (RFC 3492, обновен с 5891).

Трансформира Unicode низ в ASCII такъв. ASCII буквите в Unicode низа се представят непроменени, а non-ASCII – чрез ASCII букви, които са позволени за хост имена (LDH).

Punycode

RFC 5891 дефинира общ алгоритъм Bootstring:

низ от базови кодови точки да представя всеки уникален низ от кодови точки, изведен от по-голямо множество.

Punycode е частен случай на Bootstring. Използва конкретни стойности на параметрите, съответстващи на IDNA (Internationalizing Domain Names in Applications - RFC 5890).

Punycode

IDNA2003 пренастройва определен брой кодови точки към друг тип кодови точки.

Резултатът е ASCII-кодирана последователност, която се въвежда в DNS.

След това се припендва низа "xn--" - ASCII Compatible Encoding (ACE) префикс.

Домейни на СУ на български (21 на брой):

български	Punycode
су-свклимент-охридски.бг	xn-----elcjdrcbmffnop4afbmnk8c.xn--90ae
софуни.бг	xn--h1ajcllf.xn--90ae
су-св-климент-охридски.бг	xn------fddmeucbofgopq6afbnok1d.xn--90ae
софийскиуниверситет.бг	xn--b1agajaacdj5ag6addhrbin.xn--90ae
софиа-уни.бг	xn----7sbxcycuuf.xn--90ae
сусвклиментохридски.бг	xn--b1aeclbbjeelmm0afalmj2c.xn--90ae
су-климент-охридски.бг	xn-----llccocbkfemno2afmlk2c.xn--90ae
софияуни.бг	xn--h1aaocoof3g.xn--90ae
алмаматер.бг	xn--80aaau1agb8aq.xn--90ae
унисофия.бг	xn--h1aanfojl3g.xn--90ae
софия-уни.бг	xn----otbbtcrrf4h.xn--90ae
суклиментохридски.бг	xn--d1abibbhedkllyfkkj6b.xn--90ae
суклимент-охридски.бг	xn----htbblcbifdlmn0aflkk9b.xn--90ae
алма-матер.бг	xn----7sbabz6ahc2bs.xn--90ae
су-свклимент-охридски.бг	xn----8sbwbwiumo.xn--90ae
софиауни.бг	xn--80apbtcrrf.xn--90ae
су-свклиментохридски.бг	xn----dtbgdobblefmnn2afblnj5c.xn--90ae
су-св-климентохридски.бг	xn-----elcjerbbnegnoo4afbmoj8c.xn--90ae
уни-софия.бг	xn----otbbrhrkn4h.xn--90ae
су-климентохридски.бг	xn----htbclbbjeelmm0afllj9b.xn--90ae
софийски-университет.бг	xn----dtbjalabcdk9ag9addisbjn.xn--90ae

Resolving

DNS е йерархична именна система с три компонента – пространство на имената (как се изграждат имената), resolver-и и сървъри на имената (name servers).

Resolver-те са абонатите в Internet, които знаят URL и искат да получат съответния IP адрес.

Процесът на преобразуване се нарича resolving. Той се извършва от DNS протокола.

DNS, UDP и TCP

DNS основно използва User Datagram Protocol (UDP) на порт 53 за обслужване на заявки.

DNS заявките се състоят от една единствена UDP заявка от клиента, последвана от един единствен UDP отговор от сървъра.

Transmission Control Protocol (TCP) се използва, когато в отговора се съдържат повече от 512 bytes или при трансфер на зони.

Зони

За по-лесно администриране пространството с имената е разделено на области, наречени зони (*zones)*

Всяка зона започва от даден възел и се простира надолу до “листата” (leaf nodes) или до възли, където стартират други зони.

Данните за всяка зона се съхраняват в сървър за имена (*name server)*, наречен **Authoritative** — отговорен (или овластен) за дадената зона. Той отговаря на запитвания (queries) в рамките на зоната, използвайки *DNS протокол*.

Данните, които са обвързани с всяко име на домейн, се съхраняват под формата на ресурсни записи, *resource records* (RRs).

Зони

От особена важност е да се разбере разликата между *зона* и *домейн,* за да се вникне в същността на сървъра за имена.

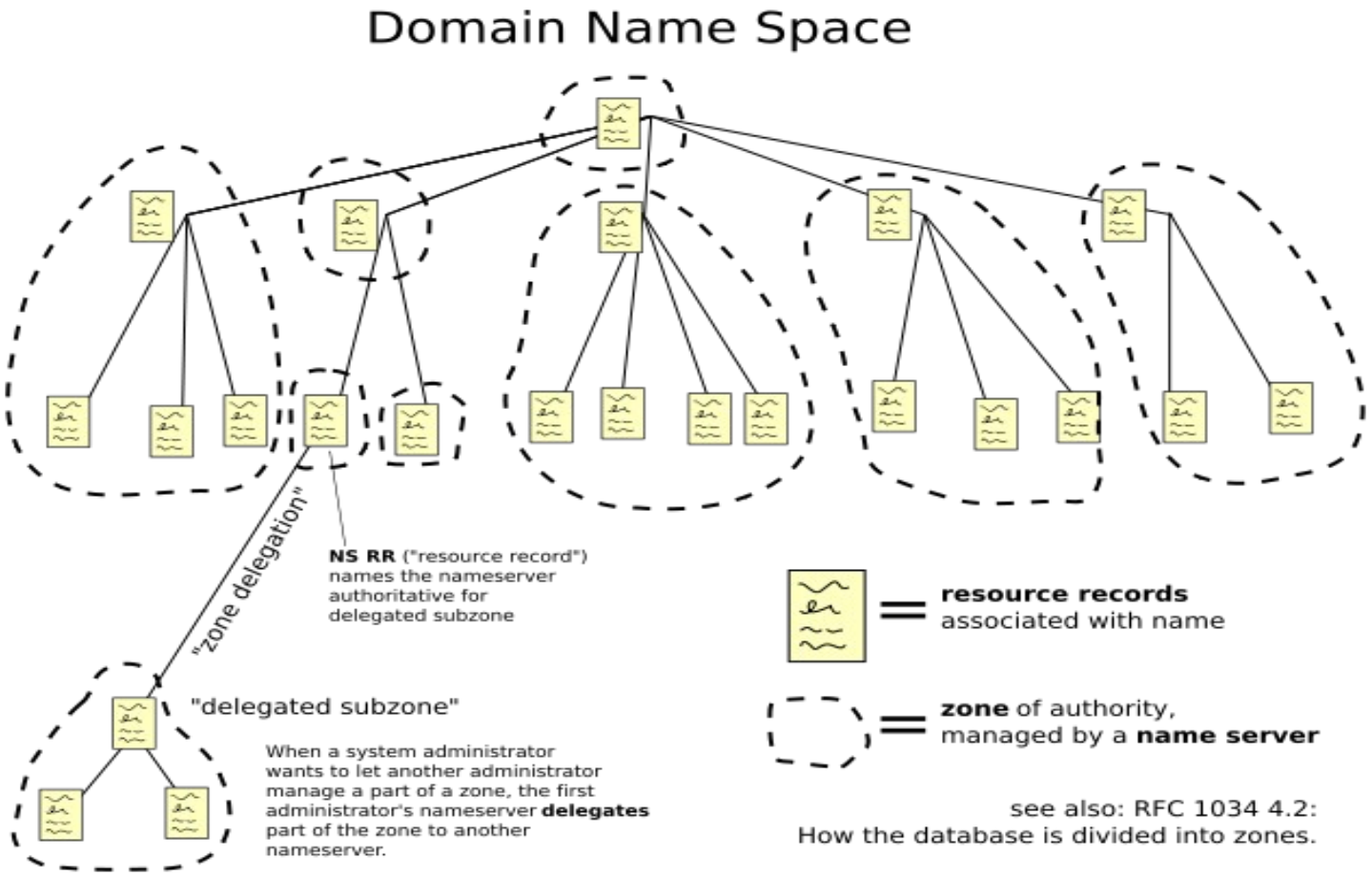
Зона е точката на делегиране на права върху DNS дървото на Интернет доставчик или клиент.

Зоната се състои от тези последователни части от дървото на домейните, за които отговорният сървър за имена има пълна информация и върху която има власт.

Състои се от всички имена на домейни, от дадена точка надолу по дървото с изключение на тези, които са делегирани на други зони.

Точката на делегиране се маркира с един или повече записа от тип **N**ame**S**erver (NS).

Зони


Domain Name Space
"zone delegation"
NS RR ("resource record") names the nameserver authoritative for delegated subzone
"delegated subzone"
When a system administrator wants to let another administrator manage a part of a zone, the first administrator's nameserver delegates part of the zone to another nameserver.
resource records associated with name
zone of authority, managed by a name server
see also: RFC 1034 4.2: How the database is divided into zones.

Йерархия на зоните: master и slave

Напр., да вземем домейна uni-sofia.bg, който включва имена като www.uni-sofia.bg, email.uni-sofia.bg и др.

Зоната uni-sofia.bg съответства точно на домейна uni-sofia.bg.

Докато поддомейнът fmi.uni-sofia.bg е делегиран на други сървъри за имена (съответно администраторите на ФМИ) и е друга зона, подчинена (slave) на зоната uni-sofia.bg (която се явява master).

Hint зона. В тази зона се дефинират root-servers.

Authoritative Name Servers. Master и Slave.

Всяка зона се обслужва най-малко от един овластен сървър за имена (*authoritative name server)*, който държи всички данни за зоната.

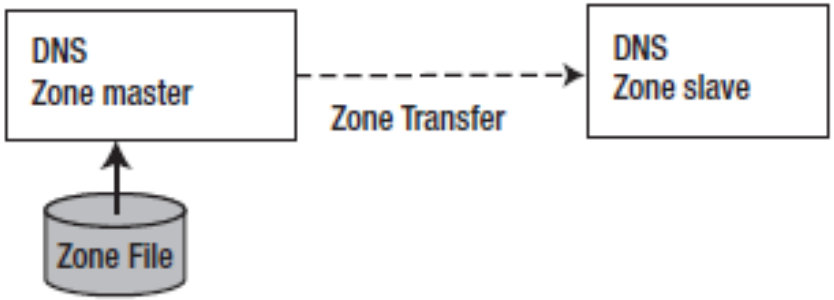
За по-висока надеждност се препоръчва зоната да има два или повече такива сървъри.

Той зарежда съдържанието на зоната от локален файл, редактиран ръчно или генериран от някакъв друг локален файл.

Този файл се нарича *зонов* - *zone file* или *master file.*

slave сървърите зареждат съдържанието на зоната от друг сървър (обикновено от master) чрез процес на репликация - zone transfer.

Зонов трансфер



Refresh таймер – периодът, през който slave DNS сървърът "запитва" master на зоната.

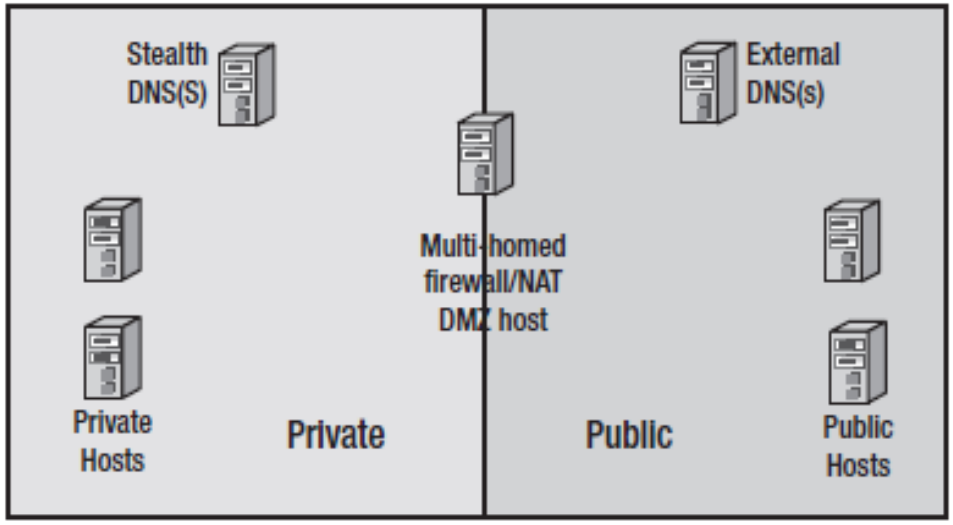
Ако serial No. (master) > serial No. (slave)

→ зонов трансфер

При овластените сървъри йерархията е отдолу нагоре: master сървърът на master зоната изпълнява slave функции по отношение на master сървъра на slave зоната.

Stealth Server

Stealth server (скрит) DNS сървър, който не се появява в никакви видими NS RRs записи за дадена зона или домейн.



Конфигурационен файл named.conf (на СУ)

```
options
{
        // Put files that named is allowed to write in the data/ directory:
        directory               "/var/named";           // "Working" directory
...
        listen-on port 53       { any; };
        listen-on-v6 port 53    { any; };

        allow-query             { any; };
        allow-query-cache       { none; };

        allow-notify { key "stealth-ns1" ; };

        dnssec-enable yes;
};
include "/etc/zones.conf";
```

/etc/zones.conf

```
// zone "." IN {
//         type hint;
//         file "named.ca";
//};
...
zone "fmi.uni-sofia.bg" {
        type slave;
        file "slaves/fmi.uni-sofia.bg";
        masters { 62.44.101.1; 62.44.96.7; };
        };
```

/etc/zones.conf

```
zone "theo.uni-sofia.bg" {
        type slave;
        file "slaves/theo.uni-sofia.bg";
        masters { 62.44.106.1; 62.44.96.7; };
        };
...
zone "slav.uni-sofia.bg" {
        type master;
        file "slav.uni-sofia.bg";
        };
```

named.conf във факултет (slave зона)

```
zone "." IN {
        type hint;
        file "named.ca";
};
zone "theo.uni-sofia.bg" IN {
        type master;
        file "theo.uni-sofia.bg";
};
zone "0.0.127.in-addr.arpa" IN {
        type master;
        file "named.local";
};
zone "106.44.62.in-addr.arpa" IN {
        type master;
        file "106.44.62.in-addr.arpa";
};
```

Пример на зонов файл (на факултет)

```
$ORIGIN .
$TTL 3600       ; 1 hour
theo.uni-sofia.bg       IN SOA  ns.theo.uni-sofia.bg. root.ns.theo.uni-sofia.bg.theo.uni-sofia.bg. (
                                2009030901 ; serial
                                3600       ; refresh (1 hour)
                                900        ; refresh retry (15 minutes)
                                3600000    ; expire (5 weeks 6 days 16 hours)
                                3600       ; nx domain TTL (1 hour)
                                )
                        NS      ns.theo.uni-sofia.bg.
                        NS      ns1.uni-sofia.bg.
                        NS      ns2.uni-sofia.bg.
                        MX      5 ns.theo.uni-sofia.bg.
                        MX      10 ns1.uni-sofia.bg.
                        MX      20 ns2.uni-sofia.bg.
$ORIGIN theo.uni-sofia.bg.
2017012600._domainkey      TXT   "v=DKIM1\; k=rsa\; t=s\; s=email\; "...
assitenti               A       62.44.106.76
autoconfig              CNAME      mailbox.uni-sofia.bg.
```

Таймери

TTL – времето (в секунди), за което даден запис се кешира от друг (slave) сървър.

nx domain TTL – периодът от време, за което се кешира отрицателен отговор от резолвер.

Ресурсни записи

SOA определя кой е първичният сървър и как се обработват данните към него.

NS съдържа информация кои DNS сървъри са отговорни за този домейн.

MX указва име на хост, готов да приема електронна поща в рамките на домейн.

Адресните записи съдържат съответствие между име и IP-адрес. Имат следния формат:

<hostname> A <IP address>

Ресурсни записи

В DNS е възможно създаването на прякори, т.е. няколко имена да отговарят на един и същ IP адрес. Това става с помощта на CNAME-записите, които имат следния формат:

```
mail        CNAME   tiger
proxy       CNAME   tiger
tiger       A       62.44.118.1
```

Ресурсни записи в IPv6 (glue records)

...

```
$ORIGIN uni-sofia.bg.
ns1             A     62.44.96.140
                AAAA    2001:67c:20d0:ff::140
ns2             A     62.44.96.141
                AAAA    2001:67c:20d0:ff::141
ns              A       62.44.96.1
                AAAA    2001:67c:20d0:ff::142
ady             A     62.44.96.7
                AAAA    2001:67c:20d0:ff::143
```

Резолвинг на имена

За да се използва системата на URL-имената в клиента (resolver) трябва да има агент, който да може да работи с URL - началото на resolving процеса.

Освен това в клиента (браузъра и/или ОС) трябва да има и малък кеш, в който да се съхранява информация за вече заявени и resolve-нати адреси за този клиент.

Също така, клиентът трябва да разполага с адрес на DNS сървър, който отговаря за съответната област.

Резолвинг на имена

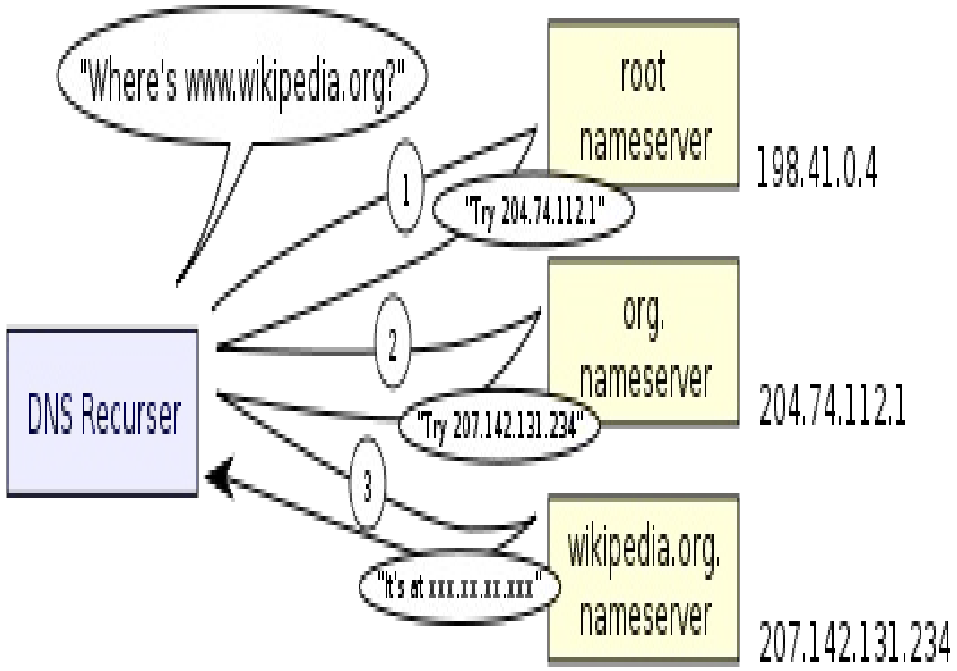
Когато към агента се подаде URL за resolve-ане той първо проверява дали отговора не стои в кеша.

Ако не, той изпраща заявка до рекурсивен DNS сървър.

DNS сървърът може да формира три типа заявки – рекурсивна, итеративна или инверсна.

Изпълнение на итеративна заявка

Една тежка процедура, която товари много ***root*** сървърите



Рекурсивна заявка

При рекурсивна заявка DNS сървърът има прилежащ към него друг сървър за имена.

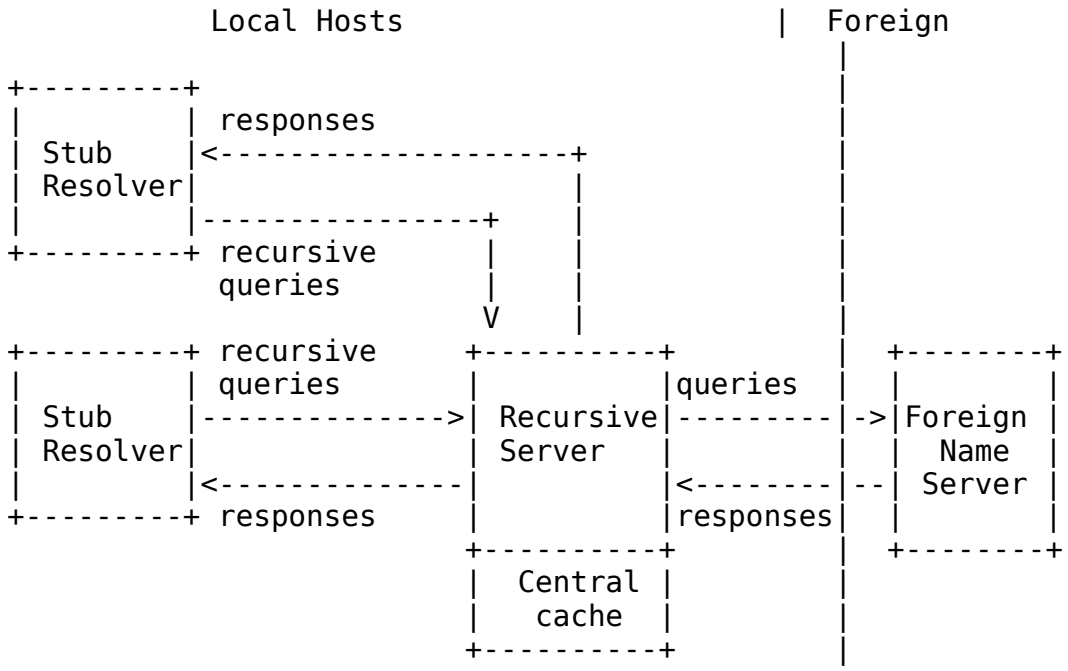
Този сървър също може да има кеш, който евентуално да съдържа отговора.

Сървърът може да съдържа отговора в своите зонални файлове.

Ако и двата случая не са налице, но има конфигуриран друг сървър за имена, той ще изпрати заявката към него и т.н.

В един момент някой сървър по описаната верига може да направи рекурсивната заявка в итеративна.

Рекурсивно търсене



Recursive (Caching) Name Servers

resolver библиотеките, които присъстват в повечето операционни системи, са *stub resolvers*, т.е те не са способни да изпълняват пълния процес на DNS резолюция, "говорейки" директно с authoritative servers.

Те разчитат на локален сървър за имена, който да изпълнява резолюцията вместо тях.

Такъв сървър се нарича "***recursive***" (рекурсивен) сървър за имена, защото изпълнява *рекурсивни търсения* за сметка на локалните клиенти.

Caching (*recursive*) Servers

За да се подобри производителността, рекурсивните сървъри кешират резултатите от търсенията, които са изпълнили.

Процесите на рекурсия и кеширане са взаимно свързани, на термините *recursive server* и *caching server* често се гледа като на синоними.

Продължителността от време, за който един запис се държи в кеша, се контролира от Time To Live (TTL) полето в него.

Caching Servers. Forwarding.

Кеширащият сървър за имена не е необходимо да изпълнява сам пълното рекурсивно търсене.

Вместо това той препраща (*forward*) някои или всички заявки, които не може да удовлетвори, от своя кеш към кеша на друг сървър за имена, който се определя като ***forwarder***.

Cache1.uni-sofia.bg

```
options
{
        directory               "/var/named";           // "Working" directory
        dump-file               "data/cache_dump.db";
       ...
        listen-on port 53       { any; };
        listen-on port 5353     { any ; };

        listen-on-v6 port 53    { any; };
        listen-on-v6 port 5353  { any ; };

        allow-query-cache       { recursive-clients ;};
        allow-query             { recursive-clients ;};
        allow-recursion         { recursive-clients ;};

        recursive-clients 10000;

};
```

Многофункционални сървъри

Сървърът за имена BIND може едновременно да бъде и **master** за някои зони, и **slave** за други зони, и кеширащ (рекурсивен) сървър за определен брой локални клиенти.

Все пак, функциите на овластени (authoritative) услуги за имена и такива на caching/recursive са логически разделени.

Затова е по-изгодно да работят на различни машини. (Може и виртуални.) Така ще се повиши надеждността и сигурността.

В СУ: authoritative: ns1.uni-sofia.bg, ns2.uni-sofia.bg; и кеширащи:

cache1.uni-sofia.bg, cache2.uni-sofia.bg

Public DNS резолвери

На Cloudflare (претендира, че не търгува с потребителски данни с цел реклама и затова е най-бърз.):

1.1.1.1 и 1.0.0.1;

2606:4700:4700::1111 и 2606:4700:4700::1001

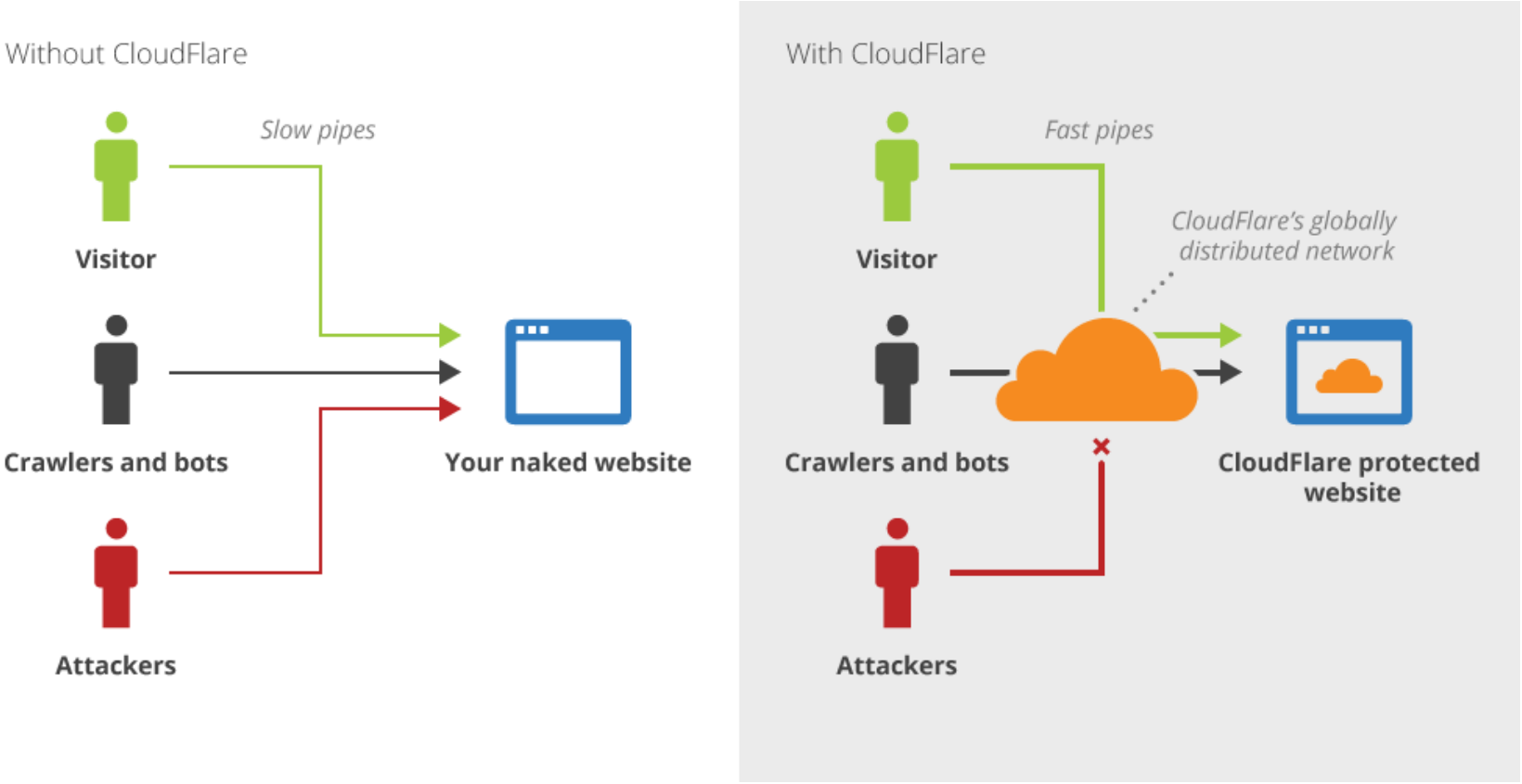
Google Public DNS (https://dns.google.com/):

8.8.8.8 и 8.8.4.4;

2001:4860:4860::8888 и 2001:4860:4860::8844

Иначе поддържат криптиране на заявките, DNSSEC, Anycast.

DNS Proxy (услуга на Cloudflare)



DNS Proxy. Как се защитава ЦИК.

www.cik.bg отговаря на IPv4: 104.20.29.59

www.cik.bg отговаря на IPv4: 104.20.28.59

www.cik.bg отговаря на IPv6: 2400:cb00:2048:1::6814:1c3b

www.cik.bg отговаря на IPv6: 2400:cb00:2048:1::6814:1d3b

Тези адреси принадлежат на AS13335 - Cloudflare, Inc., San Francisco, US

Определяте два Cloudflare сървъри за имена като authoritative nameservers за домейна (напр., cik.bg).

Прилага се Anycast за маршрутизация на DNS търсения за домейна към център за данни (ЦД) на Cloudflare (1000+ по света).

Този ЦД връща като отговор IP адрес на Cloudflare (напр. 104.20.29.59) вместо на web сървъра на ЦИК.

Root сървъри за имена

Кореновият сървър за имена (**root nameserver**) е DNS сървър, който отговаря на запитвания относно имената в коренния домейн и отправя заявките към конкретни top-level domain (TLD), т.е към техните сървъри за имена.

Всички имена в Internet завършват с точка **.** - напр., "**www.uni-sofia.bg.**" Но съвременният DNS софтуер не се нуждае от нея, когато се опитва да транслира домейн име в IP адрес.

Празният низ след крайната точка се нарича коренов домейн (root domain), а всички останали (т.е. .com, .org, .net, и т.н.) се съдържат вътре в коренния (root).

Root сървъри за имена

Информацията не се променя често, затова се кешира, така че DNS *търсенията* към **root nameservers** са относително редки.

Но в Internet има доста некоректно конфигурирани системи, които генерират трафик към root servers.

Напр., заявки с източник адрес 0.0.0.0 (т.е където и да е, навсякъде) отиват натам.

В момента има **13 root name servers**, като имената им са с формат *буква*.root-servers.net (*буква* е от A до M

Root сървъри за имена

A.ROOT-SERVERS.NET.
B.ROOT-SERVERS.NET.
...
M.ROOT-SERVERS.NET.

По-подробна информация:

http://www.internic.net/zones/named.root

http://en.wikipedia.org/wiki/Root_nameserver

http://www.root-servers.org/

Root сървърите са разпръснати на много места по Земята, т.е са **anycast**-ти, което осигурява разпределена услуга и предпазва от DoS и DDoS атаки.

Hint зоната named.ca

```
;This file holds the information on root name servers needed to initialize cache
    of Internet domain name servers
...
.                    3600000      NS    A.ROOT-SERVERS.NET.
A.ROOT-SERVERS.NET.      3600000      A     198.41.0.4
A.ROOT-SERVERS.NET.      3600000      AAAA  2001:503:ba3e::2:30
;
; FORMERLY NS1.ISI.EDU
;
.                    3600000      NS    B.ROOT-SERVERS.NET.
B.ROOT-SERVERS.NET.      3600000      A     192.228.79.201
B.ROOT-SERVERS.NET.      3600000      AAAA  2001:500:84::b
;
; FORMERLY C.PSI.NET
;
.                    3600000      NS    C.ROOT-SERVERS.NET.
C.ROOT-SERVERS.NET.      3600000      A     192.33.4.12
C.ROOT-SERVERS.NET.      3600000      AAAA  2001:500:2::c
```

Регистриране на име

Регистрирането на име не е автоматично, а става чрез специална заявка към регистратор за съответния домейн или фирма, на която са делегирани съответни права за регистрация.

За домейна .bg регистратор е register.bg.

За домейна .бг регистратор е Имена.бг ("ИМЕНА.БГ" АД).

Инверсни заявки

Инверсните заявки служат за обратен resolving:

IP --> URL

За тях се „грижи“ TLD-а **.ARPA**.

(Address and Routing Parameter Area)

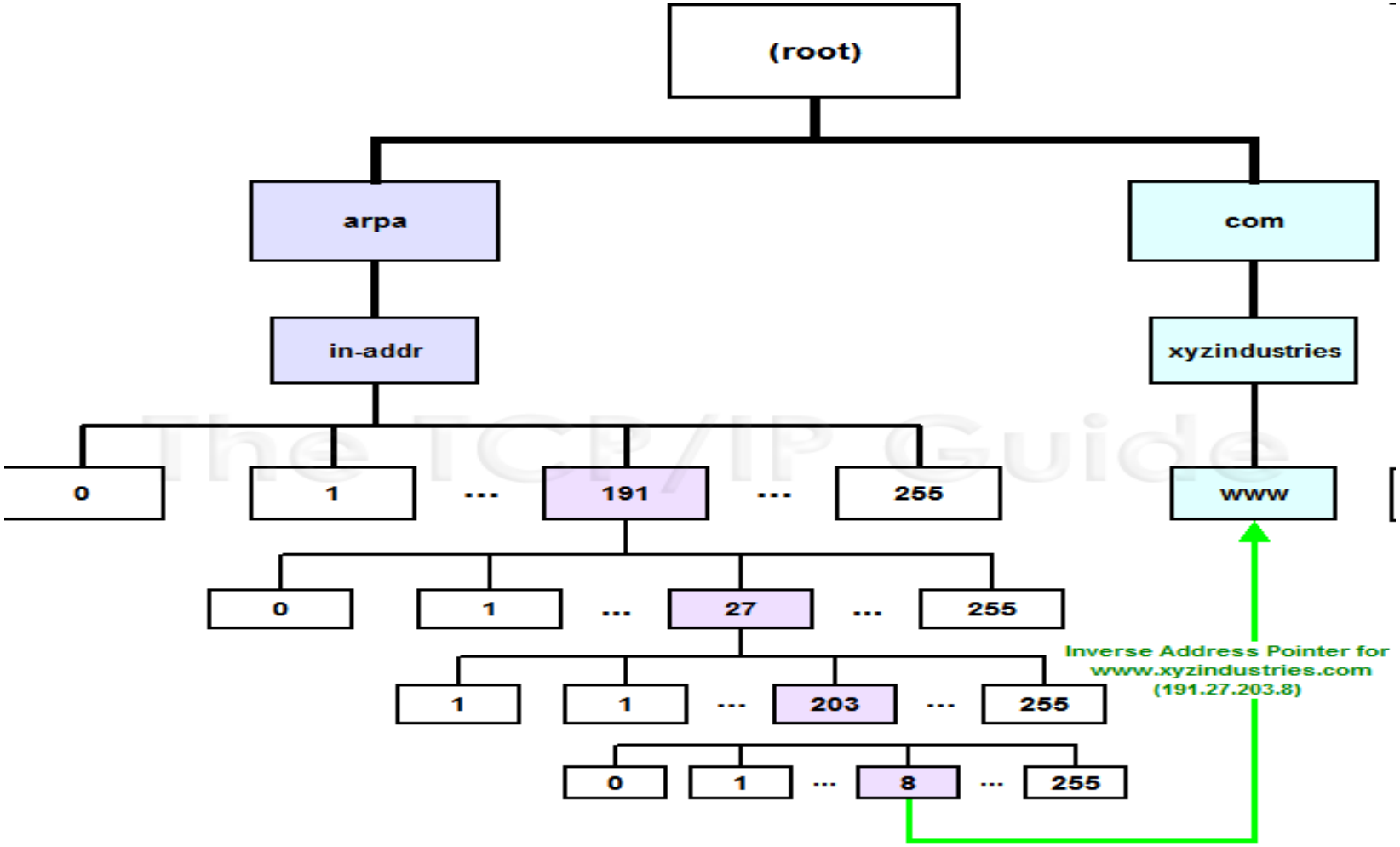
В сървърите за имена има специални записи, предназначени за инверсни заявки - **PTR** (Pointer) записи.

Йерархията на имената в този домейн следва йерархията на IP адресите:

- IPv4 адресите се обслужват от поддомейна in-addr.arpa;
- IPv6 - от поддомейна ip6.arpa.

Главни регистратори на имена в този домейн са 5-те RIRs.

IN-ADDR.ARPA Reverse Name Resolution Hierarchy



in-addr.arpa имена

Предвид спазване на йерархията in-addr.arpa имената се записват в ред, обратен на записа на IPv4 адресите – от младши към старши или отляво надясно.

Например, машина с IP адрес 10.1.2.3 ще има in-addr.arpa име 3.2.1.10.in-addr.arpa.

Това име ще има PTR ресурсен запис:

```
$ORIGIN          2.1.10.in-addr.arpa
3        PTR     foo.example.com.
```

$ORIGIN се поставят само за пояснение, но не са задължителни.

Reverse в named.conf

```
zone "0.0.127.in-addr.arpa" {
        type master;
        file "local/0.0.127.in-addr.arpa";
        };
zone "96.44.62.in-addr.arpa" {
        type master;
        file "96.44.62.in-addr.arpa.signed";
        };
...
zone "118.44.62.in-addr.arpa" {
        type slave;
        file "slaves/118.44.62.in-addr.arpa";
        masters { 62.44.118.1; 62.44.96.7; };
        };
```

named.conf в сървър 62.44.118.1 (ns.flaw.uni-sofia.bg)

```
zone "localhost" IN {
        type master;
        file "localhost.zone";
        allow-update { none; };
...
zone "0.0.127.in-addr.arpa" IN {
        type master;
        file "named.local";
        allow-update { none; };
};
...
zone "118.44.62.in-addr.arpa" IN {
        type master;
        file "118.44.62.in-addr.arpa";
```

ns.theo.uni-sofia.bg зонов reverse file

```
$ORIGIN .
$TTL 86400 ; 1 day
106.44.62.in-addr.arpa      IN SOA ns1.uni-sofia.bg. 123.ucc.uni-sofia.bg. (
2011122101 ; serial
3600       ; refresh (1 hour)
3600       ; retry (1 hour)
1209600    ; expire (2 weeks)
3600       ; minimum (1 hour)
)
NSns1.uni-sofia.bg.
NSns2.uni-sofia.bg.
$ORIGIN 106.44.62.in-addr.arpa.
1          PTR      ns.theo.uni-sofia.bg.
10         PTR      kab-1-1.theo.uni-sofia.bg.
```

Classless reverse DNS

```
zone "64.96.44.62.in-addr.arpa" {
        type slave;
        masters { 62.44.96.80 ; };
        file "slaves/64.96.44.62.in-addr.arpa";
};
```

В миналото Internet регистраторите и ISPs алокираха октет-базирани IP адресни блокове от по 256 (Class C - /24) или по-големи - класове B и A.

С въвеждането на CIDR се алокират по-малки адресни блокове. RFC 2317 решава този проблем чрез делегиране на права за администриране:

Classless reverse DNS

```
64.96.44.62.in-addr.arpa IN SOA ns2-it.fmi.uni-sofia.bg.
    misho.fmi.uni-sofia.bg. (
                        2010040802 ; serial
                        28800      ; refresh (8 hours)
                        7200       ; retry (2 hours)
                        604800     ; expire (1 week)
                        86400      ; minimum (1 day)
                        )
                NS      ns.uni-sofia.bg.
                NS      ady.uni-sofia.bg.
```

IPv6 reverse

Обратният DNS резолвинг за IPv6 адреси използва домейна ip6.arpa.

IPv6 адресите се представят като последователност от nibbles (полуоктети – 16-ни цифри) в обратен ред (както при IPv4).

Например, домейн за IPv6 address 2001:db8::567:89ab:

b.a.9.8.7.6.5.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa.

f.f.f.0.d.0.2.c.7.6.0.1.0.0.2.ip6.arpa

```
$ORIGIN .
$TTL 86400      ; 1 day
f.f.f.0.d.0.2.c.7.6.0.1.0.0.2.ip6.arpa IN SOA ns1.uni-sofia.bg. root.uni-sofia.bg. (
                        2018022200 ; serial
                        3600       ; refresh (1 hour)
                        3600       ; retry (1 hour)
                        1209600    ; expire (2 weeks)
                        36000      ; minimum (10 hours)
                        )
                NS      ns1.uni-sofia.bg.
                NS      ns2.uni-sofia.bg.

$ORIGIN 0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.f.f.f.f.0.d.0.2.c.7.6.0.1.0.0.2.ip6.arpa.
1               PTR     border-lozenets.uni-sofia.bg.
2               PTR     border-rectorate.uni-sofia.bg.
3               PTR     ivkm-gw.uni-sofia.bg.
```

Диагностични и администраторски инструменти

Dig - domain information groper:

dig @server domain query-type query-class

(вж. #man dig)

e.g. dig @localhost uni-sofia.bg или

dig @ns1.uni-sofia.bg a1.bg

root@ns ~]# **host** portal.uni-sofia.bg

portal.uni-sofia.bg has address 62.44.96.22

[root@ns ~]# **host** 62.44.96.22

22.96.44.62.in-addr.arpa domain name pointer portal.uni-sofia.bg.

rndc

С помощта на програмата remote name daemon control (**rndc**) администраторът контролира работата на name сървъра.

След всяка промяна в zone и/или reverse файл се изпълнява rndc reload (e.g.):

[root@ns named]# vi uni-sofia.bg

[root@ns named]#rndc reload uni-sofia.bg

или

[root@ns named]# vi 96.44.62.in-addr.arpa

[root@ns named]#rndc reload 96.44.62.in-addr.arpa

DNSSEC

IETF отчитат слабостите на DNS още през 1990-те, липсата на силна аутентикация. Затова разработиха DNSSEC Security Extensions (DNSSEC).

DNSSEC добавя две важни характеристики към DNS протокола:

- Аутентикация на източника на данните. Резолверът криптографски да верифицира, че получените от него данни наистина идват от зоната, която е техен източник;

- Защита на интегритета на данните. Резолверът да се увери, че данните не са изменяни по време на трансфера, защото са подписани от собственика на зоната с помощта на частния ключ на зоната.

DNSSEC (прод.)

DNSSEC прилага цифрово подписване, базирано на криптографията с публичен ключ.

DNSSEC не подписва DNS запитванията и отговорите, а самите DNS данни.

Всяка DNS зона притежава двойка ключове – публичен/частен.

Собственикът на зоната подписва DNS данните с помощта на частния ключ и генерира цифрови подписи за тези данни.

"Частният ключ" се пази и използва единствено и само от собственика на

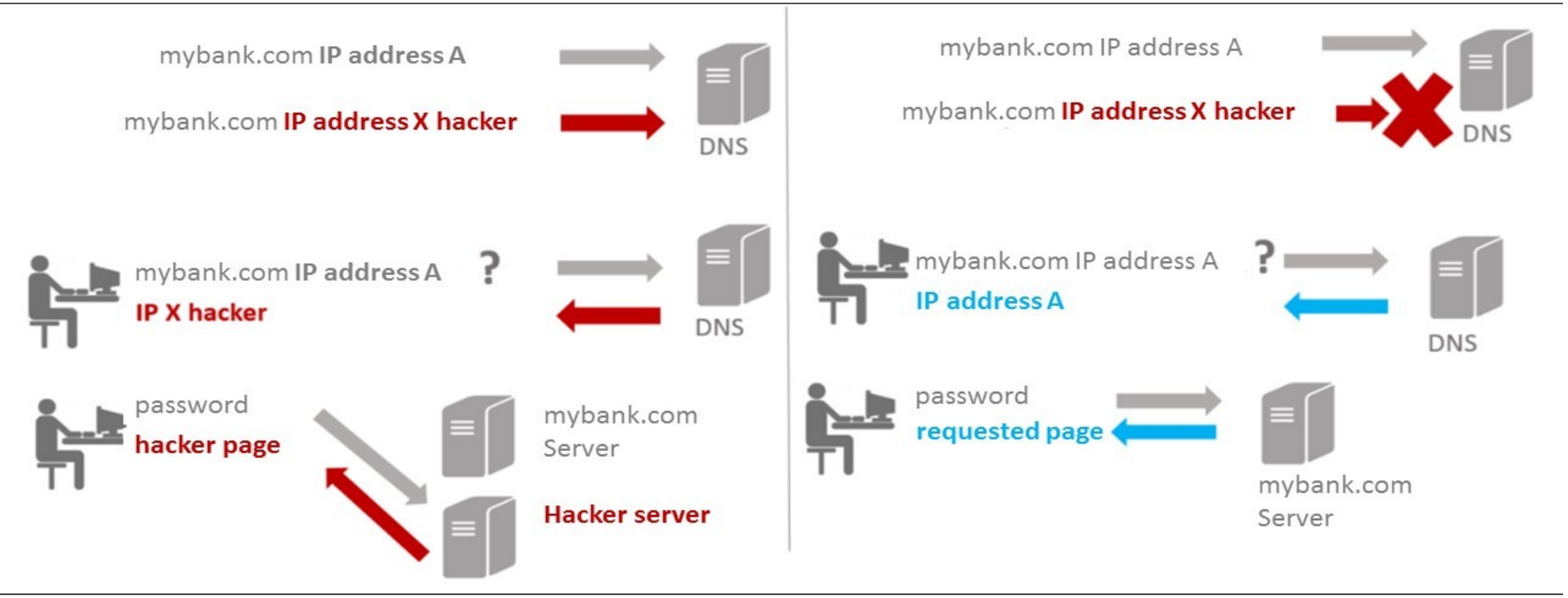
DNSSEC (прод.)

Публичният ключ се публикува в зоната и всеки може да го изтегли, напр. всеки рекурсивен резолвер, който търси данни в зоната.

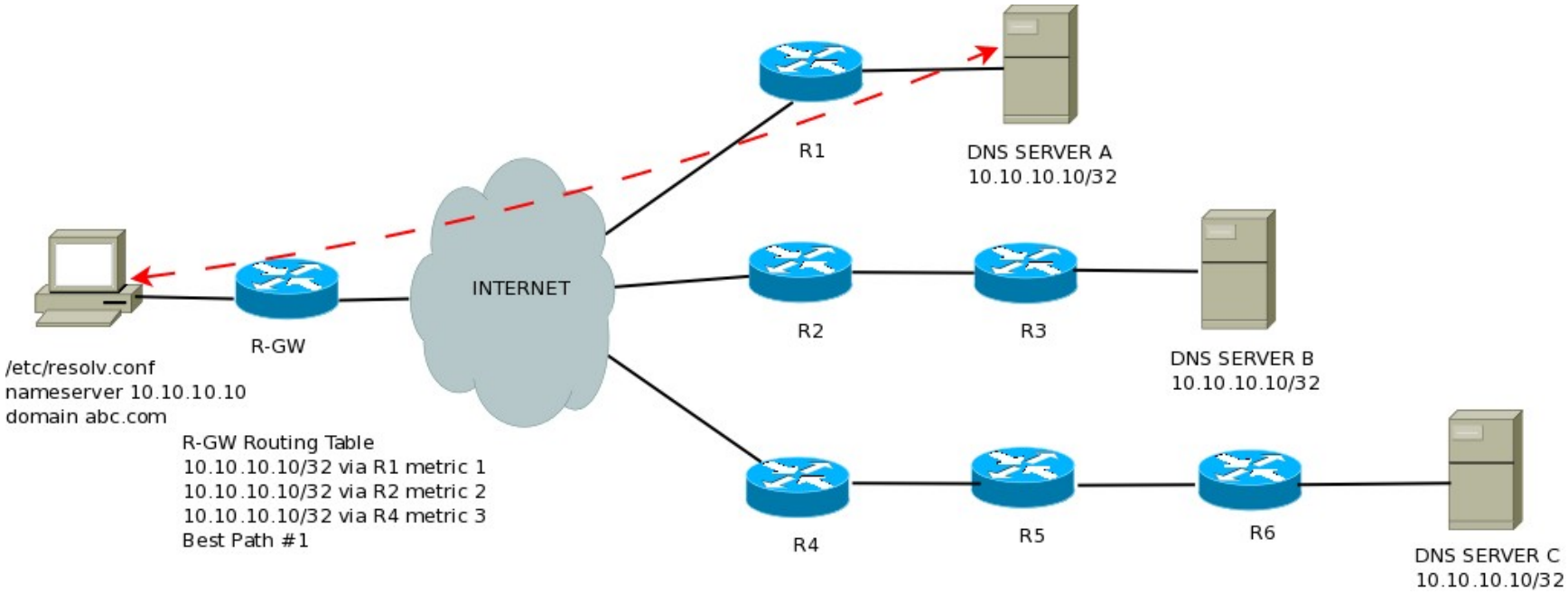
Резолверът потвърждава, че извлеченият цифров подпис върху DNS данните е валиден и че DNS данните са легитимни. Те се връщат като отговор към потребителя.

В противен случай резолверът приема, че има атака и отхвърля данните, като изпраща към потребителя съобщение за грешка.

DNSSEC (прод.)


mybank.com IP address A
mybank.com IP address X hacker
DNS
mybank.com IP address A
IP X hacker
?
DNS
password
hacker page
mybank.com
Server
Hacker server
mybank.com IP address A
mybank.com IP address X hacker
DNS
mybank.com IP address A
IP address A
?
DNS
password
requested page
mybank.com
Server

DNS Anycast схема



DNS Anycast йерархия с BGP LocPref