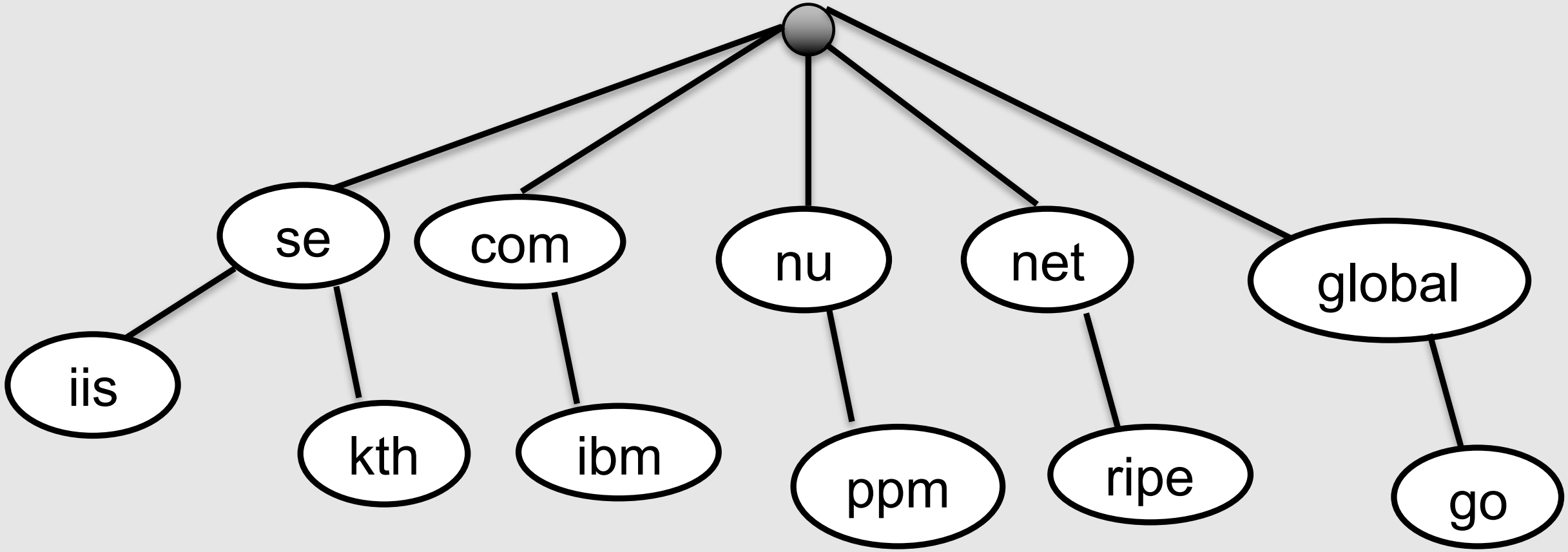

se
com
nu
net
global
iis
kth
ibm
ppm
ripe
go

INTERNET STIFTELSEN

Internets domännamnssystem*

Föreläsning FL10, HT 2023

Mats Dufberg

* Se “Internets domännamnssystem”

Innehåll

▶Namn från IP-adress

[Till Innehåll]

Hitta namn från IP-adress

DNS ger oss ett sätt att slå upp slå upp IP-adressen för namn via en A- eller AAAA-post:

```
namn.se. A 192.0.2.5
```

När vi har fått adressen så är det den vi använder i vidare kommunikation. Många gånger är det tillräckligt, men när vi administrerar system så kan vi vilja veta vilket namn som ligger bakom IP-adressen 192.0.2.5.

Hitta namn från IP-adress

Det finns inget sätt att automatiskt hitta det eller de namn som pekar ut en viss IP-adress med A- eller AAAA-post.

Det går inte att leta igenom hela databasen för vi kommer inte åt den.

Det finns inget sätt att fråga DNS – "Vilken är A-posten som har IP-adressen 192.0.2.5?"

▶Baklängesuppslagning

[Till Innehåll]

Baklängesuppslagning

Istället så kan vi konfigurera DNS med den information som behövs för att göra uppslagningen från IP-adress till namn.

Det enda sätt vi kan slå upp data i DNS är att starta med ett namn – ***owner name*** – och sedan ge data för det.

Det generella sättet att lägga informationen om vilket namn som ligger bakom en viss IP-adress är att mappa in IP-adresserna som domännamn i DNS-trädet.

Baklängesuppslagning

Det finns ett standardiserat sätt att skapa dessa namn

1. Ett namn skapas från, in princip, varje IP-adress.
2. Dessa namn läggs i ett träd under infrastrukturtoppdomänen .arpa.

Detta kallas ”baklängesuppslagning”, ”revers” eller ***reverse lookup***.

IP-adresser i arpa

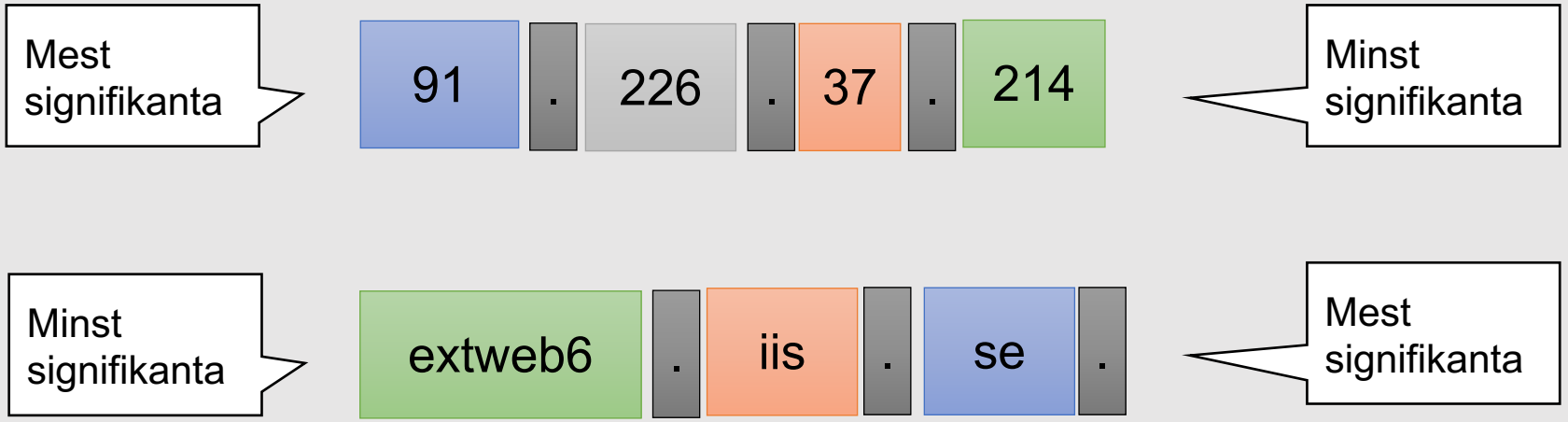
Vi har två subträd under arpa för baklängesuppslagning:

- in-addr.arpa (IPv4)
- ip6.arpa (IPv6)

I början användes ip6.int för baklängesuppslagning av IPv6, men det används inte längre.

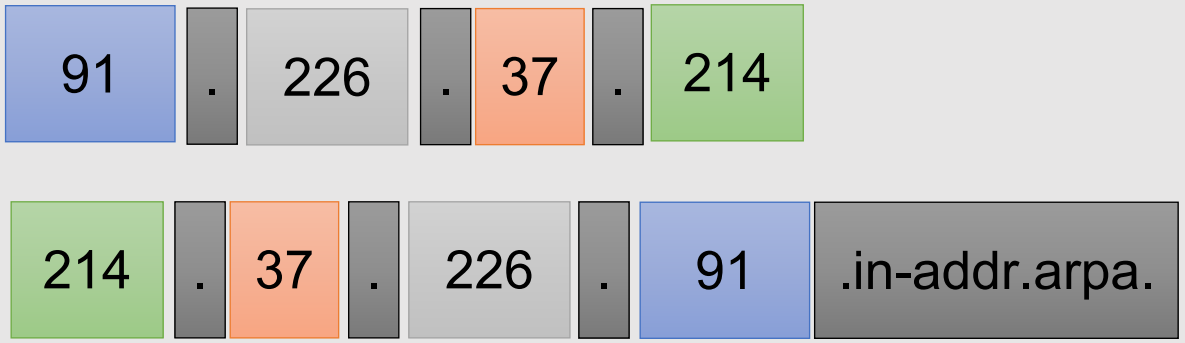
IP-adress mot domännamn

```
extweb6.iis.se.   A   91.226.37.214
```



Gör domännamn av IPv4-adress

Skapa ett unikt domännamn per IP-adress. Respektera domännamnets signifikansordning.



Gör domännamn av IPv4-adress

214.37.226.91.in-addr.arpa – ett namn eller en IP-adress?

Det är ett domännamn som vilket annat domännamn som helst. Det lyder samma regler som alla andra domännamn. ”214” i namnet ovan är inget tal, utan en namnsträng, en ***label***.

Kärnan av DNS-protokollet gör ingen skillnad på dessa namn och ”vanliga” namn.

Men dessa har en annan användning än ”vanliga” domännamn.

Normaliserat format på IP-adressen

IP-adresser skrivs ibland på andra sätt, t.ex. 091.226.037.214. Innan in-addr.arpa-domänen skapas så måste formatet ha följande form:

- Fyra decimala oktetter
- Ingen inledande nolla om inte oktetten är noll.

T.ex.

- 91.226.37.214
- 10.0.0.1

▶Posttyp PTR

[Till Innehåll]

PTR

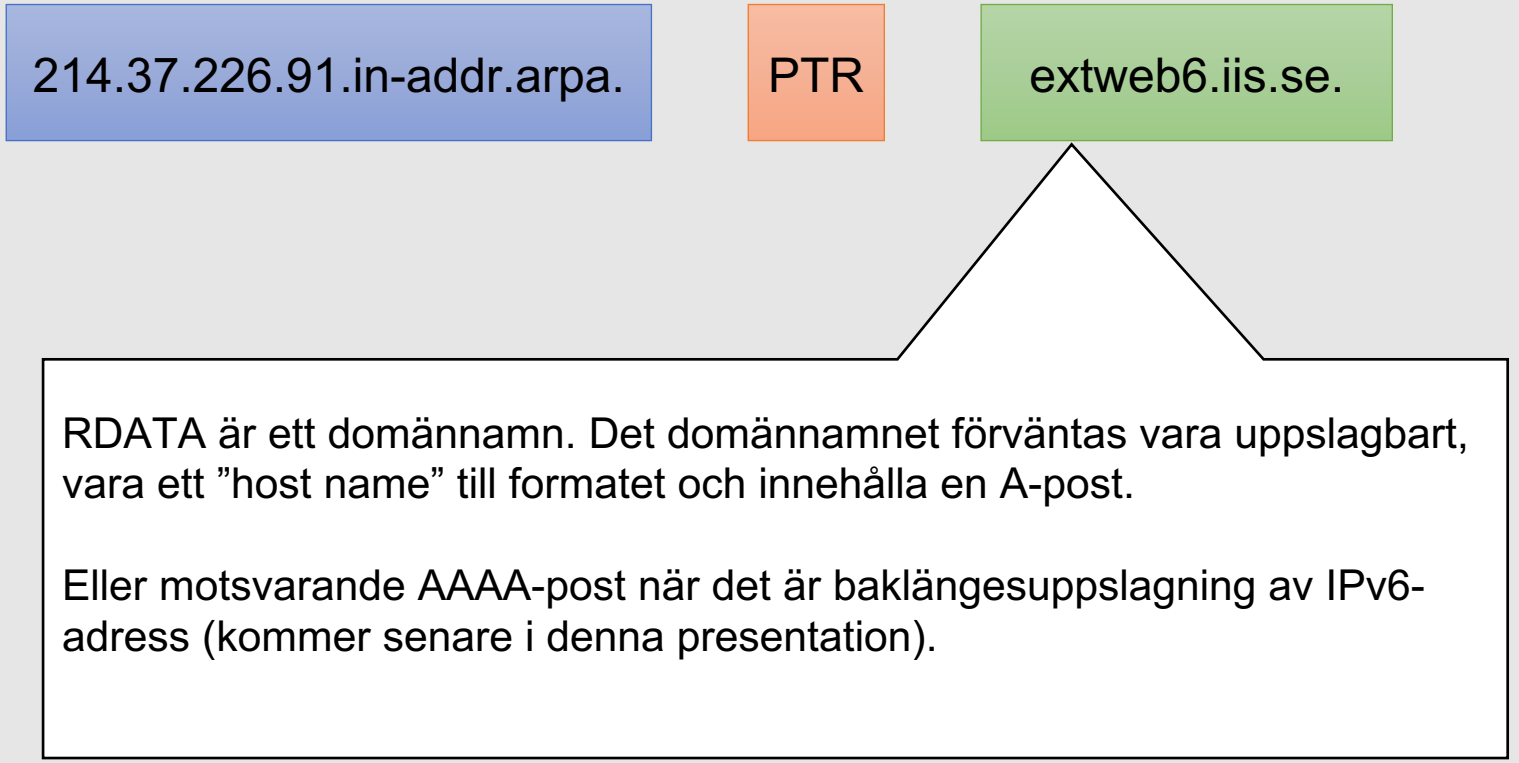
```
$ dig -x 91.226.37.214 +noedns

; <<>> DiG 9.10.6 <<>> -x 91.226.37.214 +noedns
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 2994
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;214.37.226.91.in-addr.arpa.  IN  PTR

;; ANSWER SECTION:
214.37.226.91.in-addr.arpa. 60    IN  PTR extweb6.iis.se.

;; Query time: 51 msec
;; SERVER: 172.17.41.10#53(172.17.41.10)
;; WHEN: Wed Feb 13 11:15:11 CET 2019
;; MSG SIZE  rcvd: 72
```

Om man ger växel ”-x” till ”dig” följt av en IP-adress så kommer den att skapa korrekt namn, domännamn, enligt standarden för baklängesuppslagning.

Om man inte anger något annat så blir frågetypen PTR när man har valt "-x" till "dig".

Posttyp PTR

PTR = ”Pointer”



PTR, CNAME och NS

Följande DNS-poster som har samma format på RDATA och kan innehålla samma namn:

```
namn.se.                    NS     ns1.namn.se.
host5.namn.se.              CNAME  ns1.namn.se.
25.2.0.192.in-addr.arpa.    PTR    ns1.namn.se.
```

En resolver kommer att hantera dessa DNS-poster på helt olika sätt (nästa bild).

PTR, CNAME och NS

En resolver kommer att hantera dessa DNS-poster på helt olika sätt:

- NS-posten kan användas för att slå upp zoninnehållet mot en namnserver.
- Ett CNAME kommer att följas och namnet i RDATA kommer att slås upp.
- Resolvern kommer inte att göra något alls med namnet som PTR pekar på. Det är helt upp till applikationen som frågade efter PTR.

PTR, CNAME och NS

Applikationer som får PTR, CNAME och NS kommer också att hantera deras RDATA på olika sätt eftersom de har olika funktioner och syften.

▶Användning av baklängesdata

[Till Innehåll]

Baklängesuppslagning

Om följande finns,

```
5.2.0.192.in-addr.arpa.  PTR  www.namn.se.
```

så förväntas även följande finnas.

```
www.namn.se.  A  192.0.2.5
```

men det är helt OK om vi har

```
www.namn.se.    A  192.0.2.5
host5.namn.se.  A  192.0.2.5
```

Om A-posterna inte finns så har vi ingen möjlighet att hitta tillbaka till IP-adressen från det namn vi fick.

När används baklängesuppslagning?

- Loggfiler
- Traceroute
- Accesslistor
- Mailservrar
- Information för övervakning

Istället för att bara ha IP-adress så ger baklängesuppslagningen ett meningsfullt namn.

Trace (spårning)

www.sunet.se är ett alias (CNAME) för webc.sunet.se. Det är därför det står ”traceroute to webc.sunet.se”.

```
$ traceroute www.sunet.se
traceroute to webc.sunet.se (192.36.171.231), 64 hops max, 52 byte packets
 1  172.23.124.1 (172.23.124.1)  4.646 ms  4.454 ms  3.204 ms
 2  213.108.25.18 (213.108.25.18)  7.481 ms  6.125 ms  7.586 ms
 3  static-212-247-125-21.cust.tele2.se (212.247.125.21)  8.553 ms  6.246 ms  5.156 ms
 4  avk-core-1.bundle-ether19.tele2.net (212.151.188.26)  8.790 ms  9.160 ms  8.444 ms
 5  avk6-peer-1.ae0-unit0.tele2.net (130.244.64.71)  7.480 ms  6.976 ms  5.753 ms
 6  se-fre.nordu.net (194.68.128.24)  6.115 ms  26.237 ms  8.164 ms
 7  fre-r1.sunet.se (109.105.102.10)  7.370 ms  9.712 ms  7.354 ms
 8  stockholm-fre-r2.sunet.se (130.242.4.95)  7.463 ms  10.434 ms  6.498 ms
 9  * * *
(...)
```

Accesslistor

Vi har en webbserver där bara de som kommer från domänen namn.se får ansluta. Vi förlitar oss på att det finns baklängesuppslagning och tittar på namnet vi får.

Bra idé?

Accesslistor

Den som kontrollerar baklängeszonen kan stoppa in vilket namn som helst så vi kan inte lita bara på det.

Om det kombineras med att vi sedan kontrollerar att det finns motsvarande framlängesuppslagning (A, AAAA till adress) så kan det vara OK.

Baklängesuppslagning och säkerhet

Baklängesuppslagning är utmärkt för information, för att underlätta läsandet av loggfiler m.m.

Man ska vara försiktig med att basera säkerhetslösningar på resultatet av baklängesuppslagningar. Minst ska det kombineras med kontroll av att motsvarande framlängesuppslagning stämmer.

Baklängesuppslagning

Om följande finns,

```
6.2.0.192.in-addr.arpa.  PTR  www.namn.se.
```

Är följande OK?

```
www.namn.se. A  192.0.2.5
```

Nej, det är inte enligt förväntan. Resulterande IP-adress ska stämma med ursprunglig.

Baklängesuppslagning

Om följande finns,

```
www.namn.se.     A   192.0.2.5
host5.namn.se.   A   192.0.2.5
```

Är följande OK?

```
5.2.0.192.in-addr.arpa.   PTR   www.namn.se.
5.2.0.192.in-addr.arpa.   PTR   host5.namn.se.
```

Ja, men det är kanske inte vad man vill.

Flera PTR på samma namn/IP-adress?

Följande finns

```
5.2.0.192.in-addr.arpa.   PTR  www.namn.se.
5.2.0.192.in-addr.arpa.   PTR  host5.namn.se.
```

I loggfiler och spårutskrifter så kan vi oftast bara ha ett namn istället för IP-adress. PTR-posterna är oordnade vilket betyder att det är godtyckligt vilken som väljs.

En PTR-post per IP-adress

Om man i loggar får olika namn för samma IP-adress i olika loggmeddelanden eller olika spårutskrifter så är det oftast mindre hjälpsamt.

I normalfallet så ska man ha en PTR-post per IP-adress.

▶Uppslagning av in-addr.arpa

[Till Innehåll]

in-addr.arpa är som en vanlig domän

När resolvern får en fråga (”query”) om en PTR-post under in-addr.arpa så kommer resolvern att hantera den som en vanlig domän.

```
; <<>> DiG 9.10.6 <<>> arpa ns @i.root-servers.net +nottl +nocl
(...)
;; ANSWER SECTION:
arpa.                   NS  m.root-servers.net.
arpa.                   NS  c.root-servers.net.
arpa.                   NS  l.root-servers.net.
arpa.                   NS  k.root-servers.net.
arpa.                   NS  b.root-servers.net.
arpa.                   NS  f.root-servers.net.
arpa.                   NS  i.root-servers.net.
arpa.                   NS  a.root-servers.net.
arpa.                   NS  d.root-servers.net.
arpa.                   NS  g.root-servers.net.
arpa.                   NS  e.root-servers.net.
arpa.                   NS  h.root-servers.net.
```

Zonen för toppdomänen .arpa ligger rotnamnservrarna.

in-addr.arpa är som en vanlig domän

```
; <<>> DiG 9.10.6 <<>> in-addr.arpa ns @i.root-servers.net +nocl +nottl
(...)
;; AUTHORITY SECTION:
in-addr.arpa.      NS  e.in-addr-servers.arpa.
in-addr.arpa.      NS  a.in-addr-servers.arpa.
in-addr.arpa.      NS  d.in-addr-servers.arpa.
in-addr.arpa.      NS  b.in-addr-servers.arpa.
in-addr.arpa.      NS  c.in-addr-servers.arpa.
in-addr.arpa.      NS  f.in-addr-servers.arpa.
```

Zonen för in-addr.arpa ligger på andra servrar (delegering).

in-addr.arpa är som en vanlig domän

```
; <<>> DiG 9.10.6 <<>> -x 159 ns @e.in-addr-servers.arpa.
(...)
;; AUTHORITY SECTION:
159.in-addr.arpa.  86400   IN  NS  r.arin.net.
159.in-addr.arpa.  86400   IN  NS  u.arin.net.
159.in-addr.arpa.  86400   IN  NS  z.arin.net.
159.in-addr.arpa.  86400   IN  NS  y.arin.net.
159.in-addr.arpa.  86400   IN  NS  x.arin.net.
159.in-addr.arpa.  86400   IN  NS  arin.authdns.ripe.net.
```

Zonen för 159.in-addr.arpa ligger hos ARIN (hänvisning). ARIN är en RIR för Nordamerika.

in-addr.arpa är som en vanlig domän

```
; <<>> DiG 9.10.6 <<>> -x 159.253.30 ns @r.arin.net +nottl +nocl
(...)
;; AUTHORITY SECTION:
30.253.159.in-addr.arpa. NS   ns2.namesystem.se.
30.253.159.in-addr.arpa. NS   ns3.namesystem.se.
30.253.159.in-addr.arpa. NS   ns1.namesystem.se.
```

Nu är vi nera på den zon som motsvarar 159.253.30.0/24 och vi får hänvisning till andra namnservrar som verkar vara hos en svensk operatör.

in-addr.arpa är som en vanlig domän

```
; <<>> DiG 9.10.6 <<>> -x 159.253.30.216 @ns1.namesystem.se. +nottl +nocl +norec
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 45146
;; flags: qr aa; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1680
;; QUESTION SECTION:
;216.30.253.159.in-addr.arpa. INPTR

;; ANSWER SECTION:
216.30.253.159.in-addr.arpa. PTR internetstiftelsen.se.

;; Query time: 6 msec
;; SERVER: 195.238.76.18#53(195.238.76.18)
;; WHEN: Mon Feb 10 22:42:35 CET 2020
;; MSG SIZE  rcvd: 91
```

Reversfrågan är speciell

När en applikation, direkt eller via ett API, frågar efter baklängesdata för en IPv4-adress så är det alltid följande format i DNS-frågan:

```
d.c.b.a.in-addr.arpa. PTR
```

Vilket motsvarar IP-adressen ”a.b.c.d”. När vi lägger in PTR-datat i zonfilen så måste det gå att slå upp namnet ovan. Direkt eller indirekt via CNAME. Och där ska det finnas en PTR-post.

Direkt data

Om vi frågar efter reversen på 192.0.2.5, d.v.s. frågan

```
5.2.0.192.in-addr.arpa.  PTR
```

så kommer följande DNS-post att kunna ge ett direkt svar:

```
5.2.0.192.in-addr.arpa.  PTR  www.namn.se.
```

Indirekt data

DNS kan alltid följa CNAME för PTR så om vi via CNAME hittar svaret så går det lika bra. D.v.s. om frågan är

```
5.2.0.192.in-addr.arpa.   PTR
```

så kommer följande DNS-poster att kunna ge ett indirekt svar, och det fungerar lika bra.

```
5.2.0.192.in-addr.arpa.     CNAME    5.0-7.2.0.192.in-addr.arpa.
5.0-7.2.0.192.in-addr.arpa. PTR      www.namn.se.
```

Indirekt data

Båda posterna nedan måste vara tillgängliga för att baklängesuppslagningen ska fungera.

```
5.2.0.192.in-addr.arpa.     CNAME 5.0-7.2.0.192.in-addr.arpa.
5.0-7.2.0.192.in-addr.arpa. PTR       www.namn.se.
```

Den första posten (CNAME) har ett ”owner name” som uppfyller kravet på att det ska vara ”d.c.b.a.in-addr.arpa” och den andra posten är en PTR. Det *motsvarar*

```
5.2.0.192.in-addr.arpa.  PTR   www.namn.se. ; Funktionellt samma!
```

Indirekt data – jämför med www.dn.se

När vi frågar efter A-posten för www.dn.se får vi också indirekt data:

```
www.dn.se.                  CNAME www.dn.se.edgekey.net.
www.dn.se.edgekey.net.      CNAME e12723.a.akamaiedge.net.
e12723.a.akamaiedge.net. A      23.199.249.127
```

Den första posten (CNAME) har ett ”owner name” som uppfyller kravet på att det ska vara ”www.dn.se” och den tredje posten är en A-post.

Det *motsvarar*

```
www.dn.se.                  A      23.199.249.127 ; Funktionellt samma!
```

Baklängesdata via CNAME

Så länge det är en obruten CNAME-kedja så **kan** det vara CNAME i flera steg, men antalet CNAME bör hållas nere till ett fåtal (1-2).

CNAME pekar ju på ett nytt namn, som blir ”owner name” för PTR-posten. Det normala är att det namnet också ligger under in-addr.arpa även om det inte måste vara så.

Indirekt data

Samma namn gör att de länkas ihop.

Normal lösning som fungerar och förstås av de flesta:

```
5.2.0.192.in-addr.arpa.      CNAME    5.0-7.2.0.192.in-addr.arpa.
5.0-7.2.0.192.in-addr.arpa.  PTR      www.namn.se.
```

En möjlig lösning som kan förvirra för både vänner och fiender:

```
5.2.0.192.in-addr.arpa.      CNAME    5.ipv4-reverse.namn.se.
5.ipv4-reverse.namn.se.      PTR      www.namn.se.
```

Baklängesnamnet är bara ett domännamn i domännamnsträdet och följer samma regler som andra domännamn.

Samma namn gör att de länkas ihop.

▶Revers följer IP-adressen

[Till Innehåll]

IP-adresstilldelning

IP-adresser måste, liksom domännamnen, vara unika på Internet annars kommer routingen att misslyckas.

- ICANN koordinerar IP-adressutdelningen genom att dela ut adresser till RIR (Regional Internet Registries).
- RIPE är RIR för Europa och västra Asien. RIPE delar ut IP-adresser i första hand till Internetoperatörer inom sitt område, t.ex. Telia.
- Internetoperatörerna, t.ex. Telia, delar sedan ut adresser till kunder (företag m.m.) i samarbete med RIPE.

IP-adresstilldelning

Reversen följer IP-adresstilldelningen. Den som får IP-adresser från RIPE, direkt eller via en Internetoperatör, kan också få reversen (motsvarande in-addr.arpa-domän) delegerad till sina namnservrar.

Om man inte har kontroll över IP-adresserna så ska man heller inte ha kontroll över reversen. De två hör ihop.

IP-adress 213.108.25.21

```
$ whois -h whois.ripe.net 213.108.25.21
inetnum:        213.108.24.0 - 213.108.25.255
netname:        INTERNETSTIFTELSEN-BLOCK3
country:        SE
org:            ORG-SfII2-RIPE
admin-c:        BYTE2-RIPE
tech-c:         BYTE2-RIPE
status:         ASSIGNED PI
mnt-by:         RIPE-NCC-END-MNT
mnt-by:         dotse-mnt
mnt-routes:     dotse-mnt
mnt-domains:    dotse-mnt
created:        2015-11-03T13:12:03Z
last-modified:  2020-03-31T11:38:09Z
source:         RIPE

(...)
```

213.108.24.0/23 är tilldelat Internetstiftelsen av RIPE.

IP-adress 213.108.25.21

Internetstiftelsen har följande IP-block:

- 213.108.24.0/24
- 213.108.25.0/24

De två IP-blocken är tillsammans samma sak som 213.108.24.0/23.

Här uppdelat i två block för att följa hur det delegeras i DNS. Det måste vara hela oktetter, d.v.s. 8 bitar, per steg.

213.in-addr.arpa

```
; <<>> DiG 9.10.6 <<>> 213.in-addr.arpa. ns
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 26790
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 5, AUTHORITY: 0, ADDITIONAL: 5


;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 3072
;; QUESTION SECTION:
;213.in-addr.arpa. IN NS


;; ANSWER SECTION:
213.in-addr.arpa. 85593 IN NS pri.authdns.ripe.net.
213.in-addr.arpa. 85593 IN NS ns3.lacnic.net.
213.in-addr.arpa. 85593 IN NS ns3.afrinic.net.
213.in-addr.arpa. 85593 IN NS ns4.apnic.net.
213.in-addr.arpa. 85593 IN NS rirns.arin.net.

(...)
```

RIPE ansvarar för zonen
213.in-addr.arpa

25.108.213.in-addr.arpa

```
; <<>> DiG 9.10.6 <<>> 25.108.213.in-addr.arpa. @pri.authdns.ripe.net.
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 8312
;; flags: qr rd; QUERY: 1, ANSWER: 0, AUTHORITY: 2, ADDITIONAL: 1
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
;; QUESTION SECTION:
;25.108.213.in-addr.arpa. IN A

;; AUTHORITY SECTION:
25.108.213.in-addr.arpa. 172800 IN NS ns.nic.se.
25.108.213.in-addr.arpa. 172800 IN NS ns3.nic.se.

(...)
```

25.108.213.in-addr.arpa är delegerat till Internetstiftelsen (som är innehavare av nic.se).

▶Delegering av in-addr.arpa

[Till Innehåll]

Tilldelning av IP-adresser till *Second*

Man kan utan problem delegera baklänges-zonerna vid punkterna i IPv4-adressen eller IPv4-adressblocken.

Vi tänker oss att företaget *First* har hand om 10.0.0.0/8 och delar ut till operatörer och företag.

Företaget *Second* är ett större företag och får många IP-adresser, 10.1.0.0/16 (drygt 65.000 adresser).

DNS-delegering från *First* till *Second*

First delegerar domänerna för baklängesuppslagning motsvarande 10.1.0.0/16 till *Second*.

```
$ORIGIN 10.in-addr.arpa. ; Motsvarar IP-blocket 10.0.0.0/8
@       SOA (...)
        NS ns1.first.xa.
        NS ns2.first.xa.
;
1       NS ns1.second.xa.     ; Delegering av 1.10.in-addr.arpa, motsvarande
1       NS ns2.second.xa.     ; IP-blocket 10.1.0.0/16 till företaget Second
```

Normalt är namnet på namnservern ett namn under en vanlig domän.

Second får 10.1.0.0/16 och baklängesuppslagningen av den delegerad till sig.

DNS-delegering från *First* till *Second*

Ett /8-block kan delas upp i 256 /16-block (0-255).

Second får ett av dessa, 10.**1**.0.0/16 (drygt 65.000 adresser).

Baklängesuppslagningen följer IP-adresserna.

Intern delegering av in-addr.arpa hos *Second*

Second tycker att zonen blir för stor och delar internt upp den i mindre delar

```
$ORIGIN 1.10.in-addr.arpa.   ; Motsvarande 10.1.0.0/16
@       SOA (...)
        NS ns1.second.xa.
        NS ns2.second.xa.
;
0       NS ns1.second.xa.    ; 0.1.10.in-addr.arpa motsvarande 10.1.0.0/24
        NS ns2.second.xa.
1       NS ns1.second.xa.    ; 1.1.10.in-addr.arpa motsvarande 10.1.1.0/24
        NS ns2.second.xa.
(...)
255     NS ns1.second.xa.    ; 255.1.10.in-addr.arpa motsvarande 10.1.255.0/24
        NS ns2.second.xa.
```

Intern delegering av in-addr.arpa hos *Second*

Ett /16-block kan delas upp i 256 /24-block (0-255).

- 10.1.**0**.0/24 (256 adresser, 10.1.0.0—10.1.0.255)
- 10.1.**1**.0/24 (256 adresser)
- 10.1.**2**.0/24 (256 adresser)
- (...)
- 10.1.**255**.0/24 (256 adresser)

Intern hantering av in-addr.arpa hos *Second*

Varje zonfil per /24-block hanterar PTR.

```
$ORIGIN 1.1.10.in-addr.arpa. ; Motsvarande 10.1.1.0/24
@       SOA (...)
        NS ns1.second.xa.
        NS ns2.second.xa.
;
1       PTR ns1.second.xa.        ; Motsvarande 10.1.1.1
2       PTR www.second.xa.        ; Motsvarande 10.1.1.2
3       PTR mail.second.xa.       ; Motsvarande 10.1.1.3
(...)
```

Intern hantering av in-addr.arpa hos *Second*

Second kunde istället ha lagt PTR-posterna direkt i 1.10.in-addr.arpa eller blandat.

```
$ORIGIN 1.10.in-addr.arpa.          ; Motsvarande 10.1.0.0/16
@       SOA (...)
        NS ns1.second.xa.
        NS ns2.second.xa.
;
1.1     PTR ns1.second.xa.          ; 10.1.1.1
2.1     PTR www.second.xa.          ; 10.1.1.2
(...)
1.20    PTR ns2.second.xa.          ; 10.1.20.1
2.20    PTR webb.second.example.    ; 10.1.20.2 – PTR mot namn i annan domän
(...)
```

Tilldelning av IP-adresser till *Third*

First (som kontrollerar hela 10.0.0.0/8) delar också ut IP-adresser till företaget *Third* som är ett medelstort företag, men som inte får lika många IP-adress som företaget *Second* fick.

Delegering av in-addr.arpa till *Third*

Third får 256 IP-adresser i ett block, 10.3.1.0/24.

```
$ORIGIN 10.in-addr.arpa. ; Motsvarar IP-blocket 10.0.0.0/8
@       SOA (...)
        NS ns1.first.xa.
        NS ns2.first.xa.
;
(...)
1.3     NS ns1.third.xa.  ; Delegering av 1.3.10.in-addr.arpa, motsvarande
1.3     NS ns2.third.xa.  ; IP-blocket 10.3.1.0/24 till företaget Third
(...)
```

Third får 10.3.1.0/24 och baklängesuppslagningen av den delegerad till sig.

Intern hantering av in-addr.arpa hos *Third*

Third hanterar zonfilen för 1.3.10.in-addr.arpa som en sammanhållen fil med PTR-posterna.

```
$ORIGIN 1.3.10.in-addr.arpa. ; Motsvarande 10.3.1.0/24
@       SOA (...)
        NS ns1.third.xa.
        NS ns2.third.xa.
;
1       PTR ns1.third.xa.         ; Motsvarande 10.3.1.1
2       PTR www.third.xa.         ; Motsvarande 10.3.1.2
3       PTR www.third.example.    ; Motsvarande 10.3.1.3 – PTR mot annan domän.
(...)
```

Normalt är namnet på namnservern ett namn under en vanlig domän.

Delegering av in-addr.arpa till *Forth* och *Fifth*

Företaget *Forth* är ett mindre företag och får färre IP-adresser, 10.100.1.0/26 (64 adresser). Nu går det inte att delegera på en oktett för Forth har bara adresserna 10.100.1.0—10.100.1.63.

Företaget *Fifth* får 64 adresser i samma /24-block, 10.100.1.64—10.100.1.127.

Det går alltså inte att delegera hela 1.100.10.in-addr.arpa till varken *Forth* eller *Fifth*.

Delegering utanför oktettgräns

Vi vill ha en lösning så att uppslagning av 1.1.100.10.in-addr.arpa går till *Forths* namnservrar. Och uppslagning av 65.1.100.10.in-addr.arpa går till *Fifths* namnservrar.

Båda baklängesnamnen måste starta i samma zon.

Delegering utanför oktettgräns

First lägger upp en zonfil för IP-blocket 10.100.1.0/24 i vilket *Forth* och *Fifth* fått adresser. För att uppläggningen av PTR ska göras av *Forth* resp. *Fifth* så lägger *First* in CNAME som pekar på namn som *Forth* resp. *Fifth* kontrollerar.

Det skulle kunna vara vilka namn som helst, men vi väljer en vanlig lösning där vi skapar "specialzoner" som *Forth* resp. *Fifth* får delegerat till sig:

- 0-63.1.100.10.in-addr.arpa (dotterzon till 1.100.10.in-addr.arpa)
  - För *Forths* 10.100.1.0—10.100.1.63
- 64-127.1.100.10.in-addr.arpa (också dotterzon till 1.100.10.in-addr.arpa)
  - För *Fifths* 10.100.1.64—10.100.1.127

”0-63” och ”64-127” är två namn, två ”lablar”, som hanteras av DNS på vanligt sätt.

Delegering och CNAME för in-addr.arpa

```
$ORIGIN 1.100.10.in-addr.arpa.    ; Motsvarar IP-blocket 10.100.1.0/24
@       SOA (...)
        NS ns1.first.xa.
        NS ns2.first.xa.
;
0-63    NS  ns1.forth.xa.       ; Delegering till specialzonen
        NS  ns2.forth.xa.       ; 0-63.1.100.10.in-addr.arpa
0       CNAME   0.0-63          ; Motsvarar 10.100.1.0
1       CNAME   1.0-63          ; Motsvarar 10.100.1.1
2       CNAME   2.0-63
3       CNAME   3.0-63
(...)
64-127  NS ns1.fifth.xa.        ; Delegering till specialzonen
        NS ns2.fifth.xa.        ; 64-127.1.100.10.in-addr.arpa
64      CNAME   64.64-127       ; Motsvarar 10.100.1.64
65      CNAME   65.64-127       ; Motsvarar 10.100.1.65
(...)
```

CNAME och NS för in-addr.arpa som FQDN

DNS-posterna som skapas för *Forth* på förra bilden, men här som FQDN.

```
0-63.1.100.10.in-addr.arpa.    NS  ns1.forth.xa.
0-63.1.100.10.in-addr.arpa.    NS  ns2.forth.xa.
0.1.100.10.in-addr.arpa.       CNAME   0.0-63.1.100.10.in-addr.arpa.
1.1.100.10.in-addr.arpa.       CNAME   1.0-63.1.100.10.in-addr.arpa.
2.1.100.10.in-addr.arpa.       CNAME   2.0-63.1.100.10.in-addr.arpa.
3.1.100.10.in-addr.arpa.       CNAME   3.0-63.1.100.10.in-addr.arpa.
```

”Owner name” i CNAME är det namn som baklängesuppslagningen kommer att fråga efter och som måste finnas för att det ska fungera.

RDATA i CNAME är det namn som det hänvisas till och som vi måste slå upp för att hitta PTR-posten. Jfr. med www.dn.se där vi sökte efter A-posten.

Zon med PTR via CNAME

I zonen nedan finns de PTR-poster som CNAME på förra bilden pekar på.

```
$ORIGIN 0-63.1.100.10.in-addr.arpa.
@       SOA (...)
        NS  ns1.forth.xa.
        NS  ns2.forth.xa.
;0      ; Används troligen inte
1       PTR     ns1.forth.xa.
2       PTR     www.forth.xa.
3       PTR     mail.forth.xa.
(...)
```

Tänkt uppslagning

```
$ dig -x 10.100.1.3 +noedns
(...)
;; QUESTION SECTION:
;3.1.100.10.in-addr.arpa.  IN  PTR

;; ANSWER SECTION:
3.1.100.10.in-addr.arpa.      60  IN  CNAME  3.0-63.1.100.10.in-addr.arpa.
3.0-63.1.100.10.in-addr.arpa. 60  IN  PTR    mail.fourth.xa.

(...)
```

Alternativ till CNAME

Om nätet är mindre än /24 (färre adresser än 256) så går det inte att delegera på ”vanligt” sätt. Finns det alternativ till CNAME?

- Istället för att operatören gör en delegering så kan den lägga upp PTR-posten på kundens beställning, t.ex. via en portal.
  - Operatören har i så fall en zonfil för hela /24 med data från två eller flera kunder.
- Istället för CNAME så går det att skapa en delegeringspunkt för det som motsvarar en enskild IP-adress. Det blir då en zonfil för varje IP-adress.
  - Tekniskt möjligt, men jag har aldrig sett någon som gör så. Enklare med CNAME om det är fler än en adress.

▶Baklängesuppslagning av IPv6-adresser

[Till Innehåll]

Baklängesuppslagning för IPv6

Baklängesuppslagning för IPv6-adresser fungerar på i princip samma sätt fast domänerna ligger under ip6.arpa istället.

- PTR-posten ska motsvara en AAAA-post istället.

Baklängesuppslagning för IPv6

Mest signifikant till vänster.

- Starta med adressen

```
2001:67c:124c:4006::214
```

Minst signifikant till höger.

- Kanonifiera adressen genom att göra en explicit. Fyll ut ev. "::" och fyll på med nollor så att alla "ord" får fyra hexadecimala siffror.

```
2001:067c:124c:4006:0000:0000:0000:0214
```

Alltid 8 grupper om 4 hexadecimala siffror.

- Tag bort alla ”:”

```
2001067c124c4006000000000000214
```

Baklängesuppslagning för IPv6

- Vänd så att minst signifikanta kommer först för att anpassa till domännamn

```
41200000000000006004c421c7601002
```

- Sätt punkter mellan de hexadecimala sifforna

```
4.1.2.0.0.0.0.0.0.0.0.0.0.0.0.0.6.0.0.4.c.4.2.1.c.7.6.0.1.0.0.2
```

- Lägg på ”.ip6.arpa.” och gör det till ett domännamn

```
4.1.2.0.0.0.0.0.0.0.0.0.0.0.0.0.6.0.0.4.c.4.2.1.c.7.6.0.1.0.0.2.ip6.arpa.
```

Delegering av ip6.arpa

Man kan utan problem delegera vid varje punkt.

```
ip6.arpa. NS a.ip6-servers.arpa.
ip6.arpa. NS b.ip6-servers.arpa.
(...)

6.0.1.0.0.2.ip6.arpa. NS pri.authdns.ripe.net.
6.0.1.0.0.2.ip6.arpa. NS ns4.apnic.net.
(...)

c.4.2.1.c.7.6.0.1.0.0.2.ip6.arpa. NS ns3.nic.se.
c.4.2.1.c.7.6.0.1.0.0.2.ip6.arpa. NS ns.nic.se.
(...)

4.1.2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.6.0.0.4.c.4.2.1.c.7.6.0.1.0.0.2.ip6.arpa. PTR
(...)
```

Delegering av ip6.arpa

Inget behov av CNAME eller ”konstiga” mellan-domäner som med IPv4 och in-addr.arpa. Varje siffra motsvara 4 bitar.

Om nätmasken inte är delbar med 4 så får man skapa två delegeringar istället.

Använd aldrig CNAME för IPv6-revers.

▶DNS-poster i reverszoner

[Till Innehåll]

Hur speciell är in-addr.arpa och ip6.arpa?

Ur DNS:s synpunkt är dessa inte speciella. Allt fungerar precis som vanligt även om namnen är ovanliga.

DNS bryr sig inte om namnen.

Hur speciell är in-addr.arpa och ip6.arpa?

Men användningen är speciell. Förväntade DNS-poster under in-addr.arpa och ip6.arpa är följande:

- SOA, NS
- PTR/CNAME
- (TXT)
- IPSECKEY (för IPsec)
- DNSKEY, RRSIG, NSEC/NSEC3, DS

Använd reverszonerna för det som de är avsedda för.

▶Låt "dig" konvertera

[Till Innehåll]

Använd -x med dig

Enkelt med dig och ”-x”. Med **fullständig** IPv4-adress i **rätt format** så får man reversfrågan i rätt format i ”question section”.

```
; <<>> DiG 9.10.6 <<>> -x 91.226.37.214 +noedns
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 34428
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;214.37.226.91.in-addr.arpa.  IN  PTR

;; ANSWER SECTION:
214.37.226.91.in-addr.arpa. 22    IN  PTR extweb6.iis.se.
```

Om du bara ska få rätt format så spelar det ingen roll vilken status svaret har (NOERROR, NXDOMAIN, SERVFAIL, REFUSED). ”Question section” ger alltid rätt format.

Använd -x med dig

Med **fullständig IPv6-adress i rätt format** så får man reversfrågan i rätt format i ”question section”.

```
; <<>> DiG 9.10.6 <<>> -x 2001:67c:124c:4006::214 +noedns
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 12658
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;4.1.2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.6.0.0.4.c.4.2.1.c.7.6.0.1.0.0.2.ip6.arpa. IN PTR


;; ANSWER SECTION:
4.1.2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.6.0.0.4.c.4.2.1.c.7.6.0.1.0.0.2.ip6.arpa. 800 IN PTR extweb6.iis.se.
```

IPv6-adressen kan vara i de normala formaten med ”:” mellan 16-bitarstalen, med eller utan förkortningar.

"Question section" ger alltid rätt format.

Använd -x med dig

För att få rätt format så behöver svaret inte finnas. Status kan vara vad som helst (NOERROR, SERVFAIL, REFUSED, NXDOMAIN).

Man får rätt format i ”question section” ändå.

- Om det är en IPv4-adress så slutar domänen på in-addr.arpa
- Om det är en IPv6-adress så slutar domänen på ip6.arpa

▶Privata IP-adresser baklänges

[Till Innehåll]

Privata IP-adresser

- 10.0.0.0/8 = 10.0.0.0 - 10.255.255.255
- 172.16.0.0/12 = 172.16.0.0 - 172.31.255.255
- 192.168.0.0/16 = 192.168.0.0 - 192.168.255.255

Viktiga att känna igen när man håller på med DNS.

https://tools.ietf.org/html/rfc1918

Privata IP-adresser

- Routas inte på Internet.
- Används på många olika interna nät, inte unika.
- Är inte tilldelade någon.
- Kan användas fritt utan ansökan eller registrering.
- Används väldigt mycket på lokala nätverk p.g.a. bristen på IPv4-adresser.

Privata IP-adresser och in-addr.arpa

Delegeringen av baklängesuppslagning följer tilldelningen av motsvarande IP-adresser.

Eftersom de privata IP-adresserna inte är tilldelade någon så kan baklängesuppslagningen inte delegeras till någon.

Det enda som finns är en delegering till ett ”svart hål”.

Publik baklängesuppslagning av 192.168.0.0

```
; <<>> DiG 9.10.6 <<>> -x 192.168.0.0 +mult +noedns
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 29327
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 0

;; QUESTION SECTION:
;0.0.168.192.in-addr.arpa. IN PTR

;; AUTHORITY SECTION:
168.192.in-addr.arpa.  875 IN SOA prisoner.iana.org. hostmaster.root-servers.org. (
                1          ; serial
                604800     ; refresh (1 week)
                60         ; retry (1 minute)
                604800     ; expire (1 week)
                604800     ; minimum (1 week)
                )

(...)
```

Om man får ett annat svar så har resolvern modifierat svaret.

Så här ser det ut i det publika DNS-trädet.

Motsvarande för adresser i 172.16.0.0/12 och 10.0.0.0/8

▶Lokal baklängesuppslagning

[Till Innehåll]

Lokal 168.192.in-addr.arpa

Privata IP-adresser som 192.168.0.0/16 finns bara lokalt och då kan man sätta upp baklängesuppslagning lokalt genom resolvern.

För att det ska fungera så måste användarna på det lokala nätverket använda den lokala resolvern, inte 8.8.8.8 eller 9.9.9.9.

Lokal 168.192.in-addr.arpa

Den lokala baklängeszonen ska sättas upp på vanligt sätt för en baklängeszon.

Det finns flera sätt att få tillgång till baklängesuppslagning på det lokala nätet via den lokala resolvern. Här presenteras ett sätt. Vi antar att framlängeszonen är namn.se och att den körs lokalt.

Lokal 168.192.in-addr.arpa

1. Lägg upp zonfil för 168.192.in-addr.arpa (eller en del av den eller motsvarande för andra privata IP-adresser) på samma server som är masterserver för zonen namn.se.
2. Konfigurera resolvern att skicka alla frågor om eller under 168.192.in-addr.arpa (eller den eller de zoner man sätter upp) till masterservern.

named.conf på resolver

Lägg in i named.conf (named.conf.local):

```
zone ”168.192.in-addr.arpa." {
   type static-stub;
   server-addresses { a.b.c.d; };
};
```

IP-adressen till masterservern där zonen 168.192.in-addr.arpa är upplagd.

Unique local address

Unique local address (ULA) är motsvarigheten till privata IP-adresser, men IPv6 istället:

- fd00::/8

Om man använder dessa så kan man behöva ha fungerande lokal baklängesuppslagning för dessa på samma sätt som för privata IPv4-adresser.

▶Om presentationen

[Till Innehåll]

Internets domännamnssystem

Denna presentation är framtagen 2019–2023 av Mats Dufberg (mats.dufberg@internetstiftelsen.se) på Internetstiftelsen (https://internetstiftelsen.se/). Den är en del av undervisningsmaterialet för kursen ”Internets domännamnssystem” vid Kungliga tekniska högskolan, KTH (kurskod HI1037) resp. Karlstads universitet, KAU (kurskod DVGC28).

Licens

Dokumenthistorik

- Rev A: Ursprunglig version HT 2023

Slut.