Servizi NIC e LIR
Tutorial sul DNS
(Seconda Parte)

Vincenzo Puglia (Consortium GARR) | 01/10/2025

Argomenti trattati

- DNS - Le Deleghe
- DNS - IPv6
- Sicurezza su DNS
- DNSSEC
- RPKI (Resource Public Key Infrastructure)

Le deleghe sul DNS

deleghe

Quando il GARR-NIC riceve la richiesta di registrazione di un dominio, lo registra con i nameserver indicati dagli Enti richiedenti e che, al tempo della richiesta, sono di fatto autoritativi del dominio, quindi rispondono alle query per un certo dominio.

All'atto della registrazione, presso le Authority del .it e .eu, **vengono definite le deleghe** sui nameserver autoritativi del .it e .eu, **verso i nameserver autoritativi dei domini di secondo livello** che di volta in volta vengono registrati per gli Enti GARR richiedenti.

Il concetto di delega (1)

La gestione del DNS dei domini può essere delegata ad altre organizzazioni o ad altri dipartimenti della stessa organizzazione.
Questo significa che chi definisce e registra un dominio non necessariamente deve gestirne il DNS ma può delegare tale attività ad un'altra organizzazione (struttura gerarchica dello spazio dei nomi). **Decentralizzare la responsabilità amministrativa attraverso il meccanismo della delega**.

Un'organizzazione di grande dimensioni (ad es. INFN) è probabile che abbia un dominio istituzionale ed un numero elevato di sottodomini (ad es. bo.infn.it o na.infn.it). Gestire i sottodomini può diventare oneroso, soprattutto se questi sono distribuiti geograficamente.

Il concetto di delega (2)

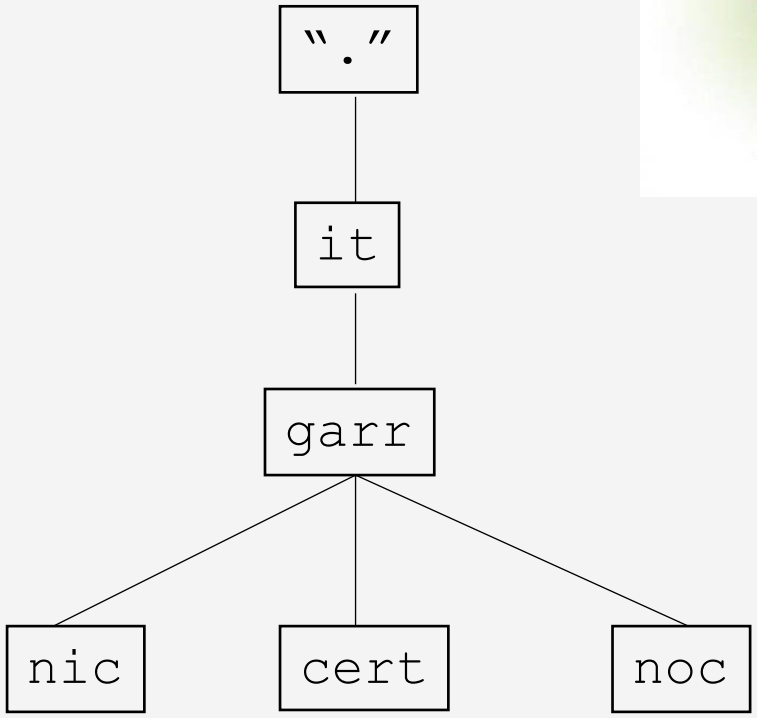
- **La delega consente di bilanciare il carico delle query**, dirette ad un dato dominio, dividendo una zona più grande **(zona padre)** in zone più piccole **(zone figli)**
    - Non tutte le queries saranno dirette ai nameserver della zona padre

- Ciò consente di delegare (e quindi distribuire) la gestione di parti di un determinato nome a dominio a diversi dipartimenti della stessa organizzazione

- **Al di sotto di un determinato nome a dominio potrebbe essere necessario aggiungere vari sottodomini** da utilizzare per scopi diversi:
    - noc.garr.it
    - nic.garr.it
    - cert.garr.it
    - pippo.example.com

Come configurare la delega

- ogni qualvolta è necessario creare un dominio figlio (es: noc.garr.it) è necessario inserire le opportune informazioni nel dominio padre (es: garr.it).
- le deleghe si effettuano mediante i record NS



Un esempio di delega (file di zona padre: garr.it)

```
$ORIGIN .
$TTL 86400      ; 1 day
garr.it         IN SOA  ns1.garr.net. nic.garr.it. (
                        2014101700 ; serial
                        86400      ; refresh (1 day)
                        7200       ; retry (2 hours)
                        2592000    ; expire (4 weeks 2 days)
                        345600     ; minimum (4 days)
                        )
$TTL 604800     ; 1 week
                        NS      ns1.garr.net.
                        NS      ns2.garr.net.
$TTL 86400      ; 1 day
                        MX      10 lx1.dir.garr.it.
                        MX      20 lx5.dir.garr.it.
$ORIGIN garr.it.
$TTL 86400      ; 1 day
;;; Delegation to subdomain
nic                             NS      srv1.garr.net.
                                NS      srv2.garr.net.
cert                            NS      dxgarr.dir.garr.it.
                                NS      srv1.garr.net.
                                NS      srv2.garr.net.
```

Esempio delega di una zona di reverse IPv4/v6 (file di zona padre: 204.193.soa (classe b)

```
File: 204.193.soa
$TTL 86400      ; 1 day
@               IN SOA  ns1.garr.net. staff-dns.garr.net. (
                        2024121001 ; serial
                        28800      ; refresh (8 hours)
                        7200       ; retry (2 hours)
                        1209600    ; expire (2 week)
                        7200       ; minimum (2 hours)
                        )
;
                NS      ns1.garr.net.
                NS      ns2.garr.net.
;
;
0               NS      vm-publicdns.ieo.it.
                NS      ns1.garr.net.
1               NS      terri1.oa-abruzzo.inaf.it.
                NS      dns.ced.inaf.it.
2               NS      dnsm.bo.infn.it.
                NS      dnsi.bo.infn.it.
                NS      server2.infn.it.
.....
```

Per le reti stesso discorso, quando si assegna una rete IPv4, ad es. una /24, occorre aggiornare il file di zona di reverse padre (/16) aggiungendo i records NS relativi ai Nameserver autoritativi della zona assegnata

Lame delegation

- **Quando si registra un nome, l'amministratore del dominio (o sottodominio)** di fatto **richiede formalmente una delega per la gestione di una parte dell'albero del DNS**
- Se però le informazioni sul dominio (ad esempio I NS) non vengono correttamente comunicate all'amministratore del dominio padre (in modo che le registri) si ottiene una "delega zoppa" (lame delegation)

- Cosa accade:
  - Se un utente cerca di contattare un host nel dominio lame, il NS non rifiuterà la query ma non riceverà una risposta
  - Il DNS ritenterà la query per molte volte, e questo coinvolgerà sia il **NS** in questione che i **NS di root**

Risoluzione inversa - Classless Delegation (RFC-2317)

- Per collegarsi a Internet, un ente ha bisogno di indirizzi IP pubblici, generalmente assegnati da un provider di servizi Internet (ISP).
- Problema:

  In alcuni contesti è necessario **dividere una /24 tra due enti differenti** (ad es. enti con pochi dipendenti), **cioè suddividere una classe di indirizzamento (/24) su domini differenti.**

  Bisogna quindi **consentire una gestione amministrativa autonoma ai vari domini** e **garantire una corretta risoluzione inversa**

- Soluzioni:

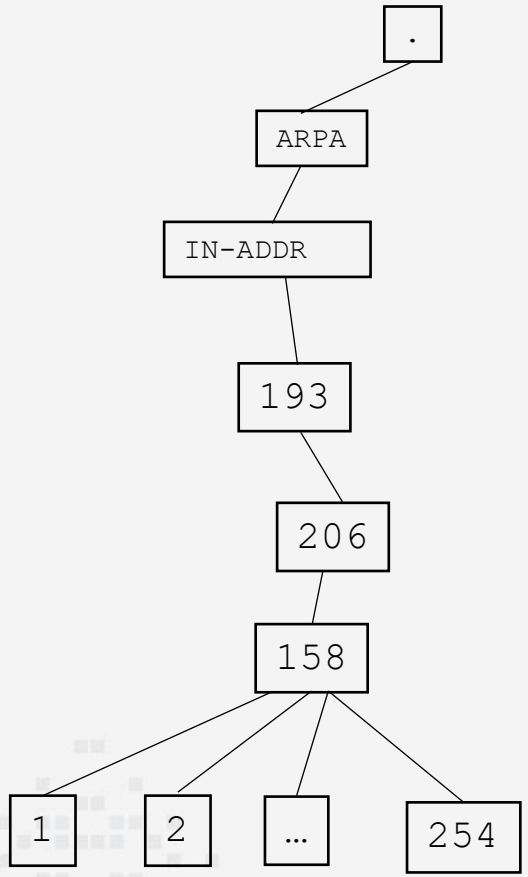
  Reti in classe B: si procede normalmente alla delega delle singole subnet equivalenti ciascuna ad una rete in classe “C” o “ /24 “ (classfull)

  Reti in classe C: e’ possibile effettuare la delega della risoluzione inversa anche di **“parti” più piccole di una rete /24 (classless)**

Risoluzione inversa - caso «classico» subnet /24

Classe B – zona Padre: file di zona 206.193.soa

```
$TTL 86400      ; 1 day
@       IN SOA  ns1.garr.net. staff-dns.garr.net. (
                        2025071600 ; serial
                        28800      ; refresh (8 hours)
                        7200       ; retry (2 hours)
                                   .....      )
                NS      ns1.garr.net.
                NS      ns2.garr.net.
;(utente1)
0               NS      alpha1.murst.it.
                NS      ns1.garr.net.
;(utente2)
1               NS      alpha1.murst.it.
                NS      ns1.garr.net.
;(utente3)
2               NS      beta2.cnr.it.
                NS      ns1.garr.net.
....
158             NS      dxgarr.dir.garr.it.
                NS      ns1.garr.net.
                NS      ns2.garr.net.
....
```

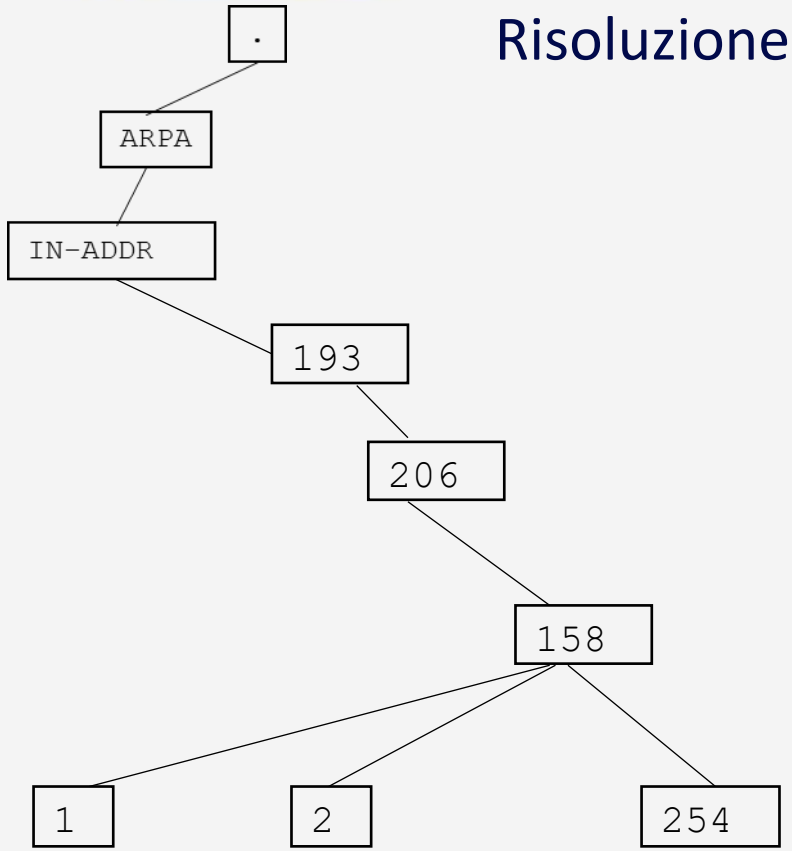


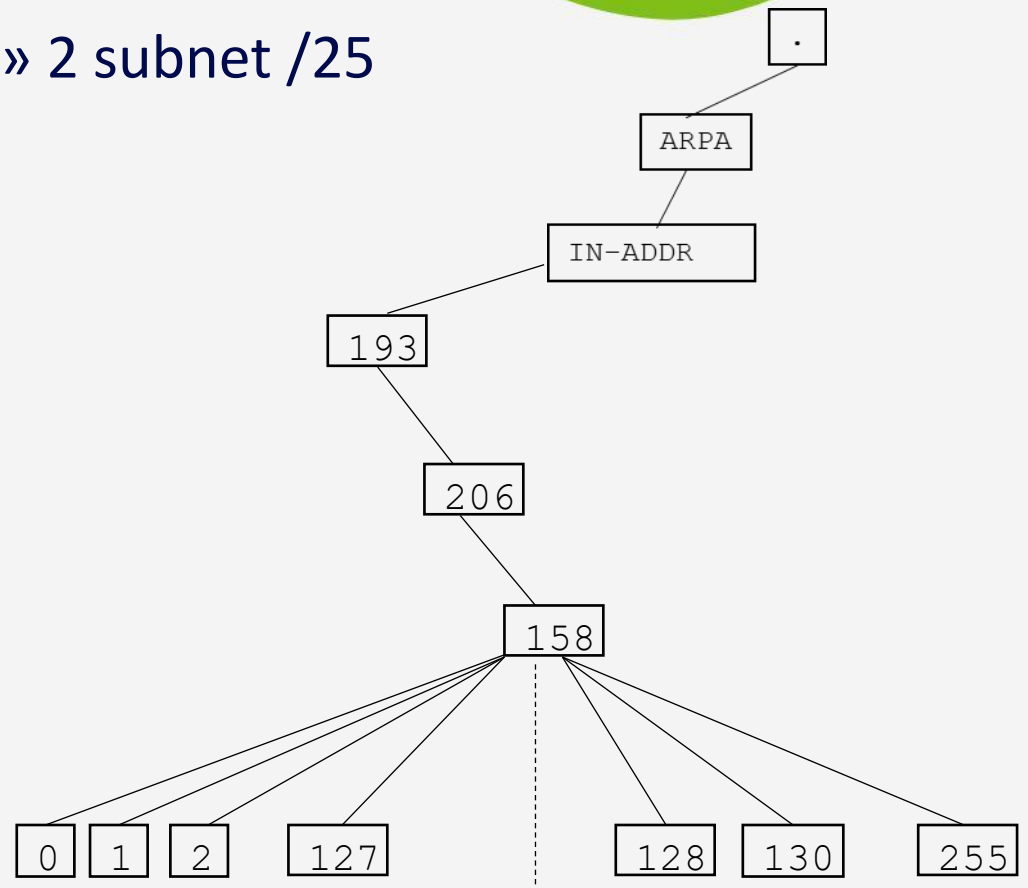
Classe C (zona Figlio): File di zona di reverse della 193.206.158.0/24:

158.205.193.soa

```
$ORIGIN 206.193.in-addr.arpa.
158             IN      SOA     dxgarr.dir.garr.it.
root.dxgarr.dir.garr.it. (
                2008021501 86400 3600 604800 86400 )
                IN      NS      dxgarr.dir.garr.it.
                IN      NS      ns1.garr.net.
                IN      NS      ns2.garr.net.
;
$ORIGIN 158.206.193.in-addr.arpa.
;
1               IN      PTR     dxgarr.dir.garr.it.
2               IN      PTR     lx1.dir.garr.it.
3               IN      PTR     lx2.dir.garr.it.
.
241             IN      PTR     vpn01.dir.garr.it.
242             IN      PTR     vpn02.dir.garr.it.
...
254             IN      PTR     pcgarr.dir.garr.it.
```

Risoluzione inversa - caso «classless» 2 subnet /25





- Si crea un nuovo “point of delegation” aggiungendo un ulteriore livello nell’albero IN-ADDR.ARPA

Esempio: suddivisione della 193.206.158.0/24 in due sottoreti /25:

193.206.158.**0**/25 (fino a 193.206.158.**127**) e 193.206.158.128/25 (fino a .255)

Risoluzione inversa - caso «classless» (2)

Configurazione lato **“zona padre” (ISP)** modificare named.conf con direttiva slave per la zona come segue:

Zona /24

```
zone "158.206.193.in-addr.arpa" {
        type master;
        file "master/net/158.206.193.soa";
        allow-transfer { 193.206.141.42; };
        allow-query { any; };
};
```

E due Zone /25 (solo se ns1 fa anche da secondario)

```
zone "128-25.158.206.193.in-addr.arpa" {
        type slave;
        file "slave/net/128-25.158.206.193.for";
        masters {
                193.206.158.130;
        };
        allow-query { any; };
        allow-transfer { none; };

};
```

Configurazione file-zona IP classless reverse delegation file: **158.206.193.soa** (RFC2317 - http://www.ietf.org/rfc/rfc2317.txt) :

```
$TTL 86400      ; 1 day
@           IN SOA ns1.garr.net. staff-dns.garr.net. (
                        2009111805 ; serial
                        28800      ; refresh (8 hours)
                        7200       ; retry (2 hours)
                        1209600    ; expire (2 week)
                        7200       ; minimum (2 hours)
                        )
                NS      ns1.garr.net.
                NS      ns2.garr.net.
;
128-25          NS      fox.inaf.it.
                NS      ns1.garr.net.
;
129             CNAME   129.128-25
130             CNAME   130.128-25
131             CNAME   131.128-25
132             CNAME   132.128-25
133             CNAME   133.128-25
...
```

Risoluzione inversa - caso «classless» (RFC-2317) (3)

Configurazione lato "zona figlio" (lato utente – fox.inoa.it)

- aggiungere a named.conf la zona reverse per la subnet classless

```
zone " 128-25.158.206.193.in-addr.arpa " {
    type master;
    file "/etc/bind/128-25.158.206.193.rev";
     allow-transfer 193.206.141.38;

};
```

- creare in /etc/namedb il file di zona di reverse per la subnet classless

```
#vi /etc/bind/128-25.158.206.193.rev
$ttl 38400
@       IN      SOA     dns.bo.ingv.it. postmaster.bo.ingv.it. (
                        2007100401
                        86400
                        3600
                        604800
                        86400 )
@       IN      NS      fox.inoa.it.
@       IN      NS      ns1.garr.net.
;
129     IN      PTR     fox1.inoa.it.
130     IN      PTR     fox2.inoa.it.
...
255     IN      PTR     fox255.inoa.it.
```

Il DNS su IPv6

- Il DNS su IPv6: cosa c'è di nuovo
- L'IPv6 e la struttura gerarchica del DNS
- L'IPv6 e i Root Nameserver
- Interazione tra IPv6 e IPv4 nel DNS
- File di zona con IPv6

Cosa c'è di nuovo nel DNS con IPv6

- L'utilizzo di IPv6 non modifica i meccanismi di base del Domain Name System
- Anche con IPv6, nel protocollo standard DNS, vengono individuate e distinte **2** tipologie di risoluzioni: DIRETTA e INVERSA

- Risoluzione diretta un nuovo resource record per associare un indirizzo IPv6 ad un nome
  - Corrispondenza **nome host** – **indirizzo IPv6 (AAAA - QuadA)**

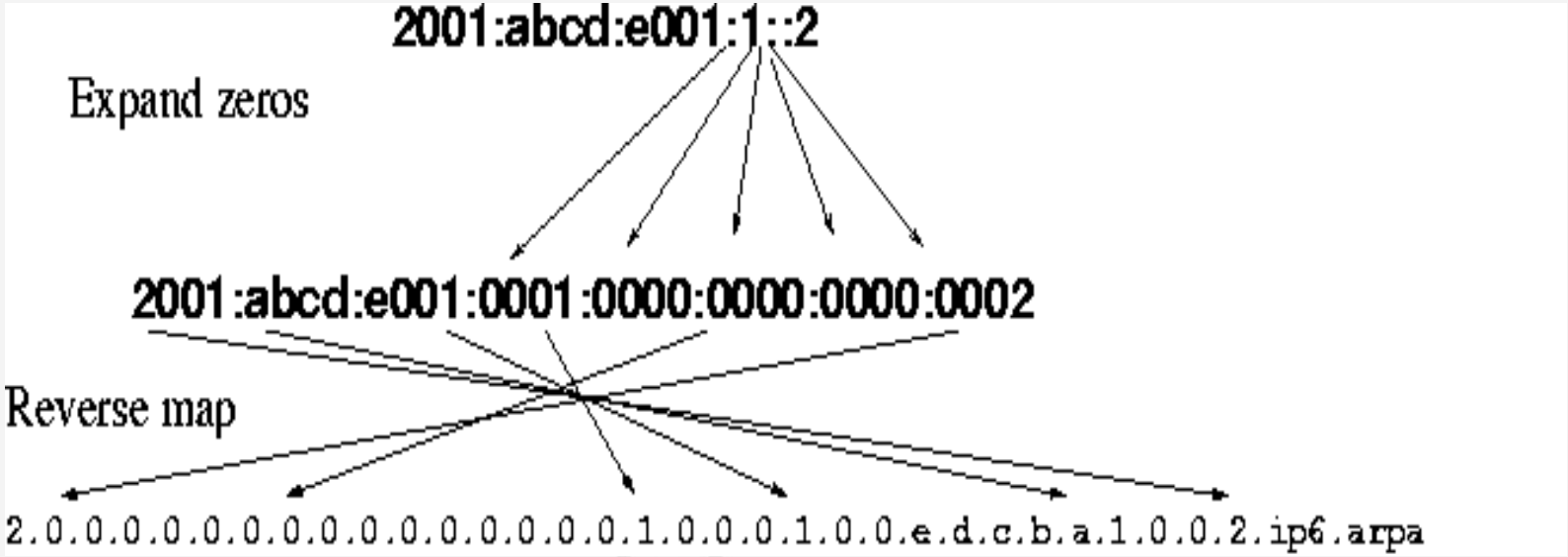
- Risoluzione **inversa**:
  - Corrispondenza indirizzo IPv6 – **nome host**
  - L'RFC 3152 indica l'ip6.arpa. come dominio da utilizzare per la mappatura del reverse lookup in IPv6 invece di in-addr.arpa.

… Se in IPv4 l'obiettivo era dare un nome ai calcolatori in IPv6 diventa un'esigenza ….

2001:0b00:0c18:0001:0290:27ff:fe17:fc1d

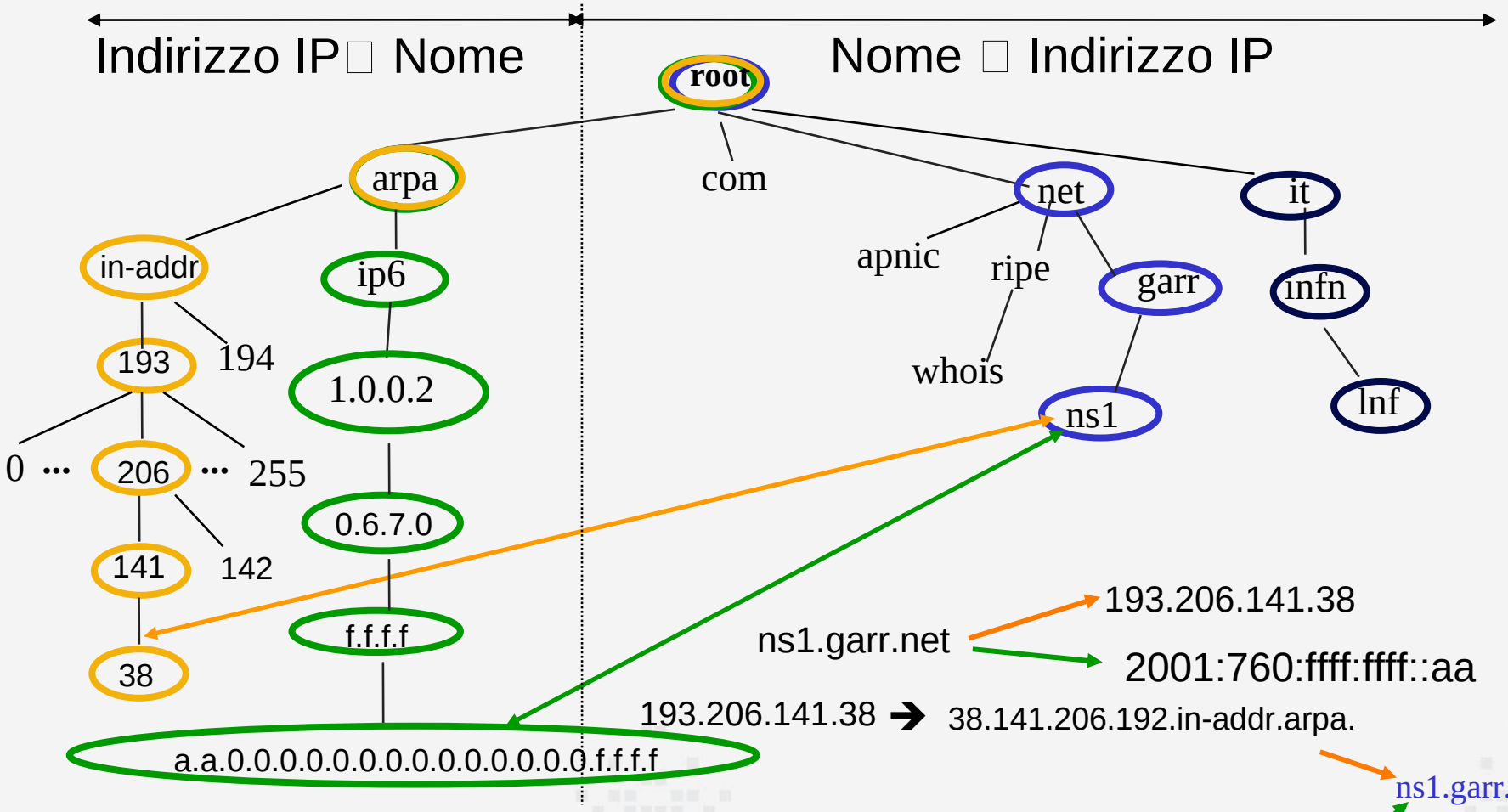
Il reverse DNS lookup

- Il **reverse** di un indirizzo IPv6 è espresso, nel dominio "**ip6.arpa**", da una sequenza di otto cifre esadecimali scritte in ordine inverso; numeri e lettere sono separati da un punto. Un indirizzo è formato da **8** campi di **16** bits ciascuno (**128** in tutto).



Consortium GARR THE ITALIAN EDUCATION & RESEARCH NETWORK

Corrispondenza tra IPv6, IPv4 e Nomi nell'albero del DNS



I root nameservers e IPv6

- I root nameserver sono gli stessi dell'ipV4, 13 in tutto il mondo (10 negli USA)
- A partire dal 4 febbraio 2008 sono stati introdotti records nella configurazione di 6 dei 13 root nameserver
  - ***http://www.iana.org/reports/2008/root-aaaa-announcement.html***

- Attualmente tutti i root NS sono configurati in dual stack (servizio IPv4/v6)

I root nameservers e IPv6

Letter	IPv4 address	IPv6 address	Old name	Operator	Location
A	198.41.0.4	2001:503:ba3e::2:30	ns.internic.net	VeriSign	Dulles, Virginia, U.S.
B	192.228.79.201		ns1.isi.edu	USC-ISI	Marina Del Rey, California, U.S.
C	192.33.4.12		c.psi.net	Cogent Communications	distributed using anycast
D	128.8.10.90		terp.umd.edu	University of Maryland	College Park, Maryland, U.S.
E	192.203.230.10		ns.nasa.gov	NASA	Mountain View, California, U.S.
F	192.5.5.241	2001:500:2f::f	ns.isc.org	ISC	distributed using anycast
G	192.112.36.4		ns.nic.ddn.mil	Defense Information Systems Agency	Columbus, Ohio, U.S.
H	128.63.2.53	2001:500:1::803f:235	aos.arl.army.mil	U.S. Army Research Lab	Aberdeen Proving Ground, Maryland, U.S.
I	192.36.148.17		nic.nordu.net	Autonomica	distributed using anycast
J	192.58.128.30	2001:503:C27::2:30		VeriSign	distributed using anycast
K	193.0.14.129	2001:7fd::1		RIPE NCC	distributed using anycast
L	199.7.83.42 (since November 2007; was 198.32.64.12)			ICANN	distributed using anycast
M	202.12.27.33	2001:dc3::35		WIDE Project	distributed using anycast

Il Record AAAA

Il Record AAAA

- Gestisce la risoluzione diretta dei nomi associati ad un indirizzo IPv6
  - $ORIGIN prova.it
  - www IN AAAA 2001:760:1010::3
- Equivalente al record A utilizzato in IPv4

Il Record PTR

- Record Pointer IPv6 [RFC 1035]
- Gestisce la risoluzione inversa di un indirizzo IPv6 associando un indirizzo ad un nome

$ORIGIN 1.0.0.0.8.1.c.0.0.0.b.0.e.f.f.3.ip6.arpa

d.1.c.f.7.1.e.f.f.f.7.2.0.9.2.0 IN PTR www.6net.garr.it

- Lo stesso tipo di record utilizzato per IPv4
  - Un nuovo modello di top level usato per IPv6: ip6.arpa

Config di un file di zona diretta «prova.it» con IPv6

```
prova.it        IN SOA  ns1.prova.it. staff-dns.prova.it. (
                        2008011101 ; serial
                        86400      ; refresh (1 day)
                        7200       ; retry (2 hours)
                        2592000    ; expire (4 weeks 2 days)
                        345600     ; minimum (4 days)
                        )
$TTL 86400
                NS      ns1.prova.it.
                NS      ns2.prova.it.
$ORIGIN   prova.it.
srv1                            IN      A       10.0.0.1
                                IN      AAAA    2001:760:1010::1
srv2                            IN      A       10.0.0.2
                                IN      AAAA    2001:760:1010::2
www                             IN      A       10.0.0.3
                                IN      AAAA    2001:760:1010::3
ns1                             IN      A       10.0.0.4
                                IN      AAAA    2001:760:1010::7
```

Config del file di zona di reverse con IPv6

```
@   IN SOA 0.1.0.1.0.6.7.0.1.0.0.2.ip6.arpa.  hostmaster.prova.it. (
                    200905260      ; Serial number (YYYYMMdd)
                    24h                         ; Refresh time
                    30m                         ; Retry time
                    2d                          ; Expire time
                    3d                          ; Default TTL
)
                                                IN    NS    ns1.prova.it.
                                                IN    NS    ns2.prova.it.


; IPv6 PTR entries
; Subnet #1
$ORIGIN 0.0.0.0.0.1.0.1.0.6.7.0.1.0.0.2.ip6.arpa.

1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0    IN    PTR    srv1.prova.it.
2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0    IN    PTR    srv2.prova.it.
3.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0    IN    PTR    www.prova.it.
7.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0    IN    PTR    ns1.prova.it.
```

Sicurezza su DNS

Denial of service (1)

Descrizione:

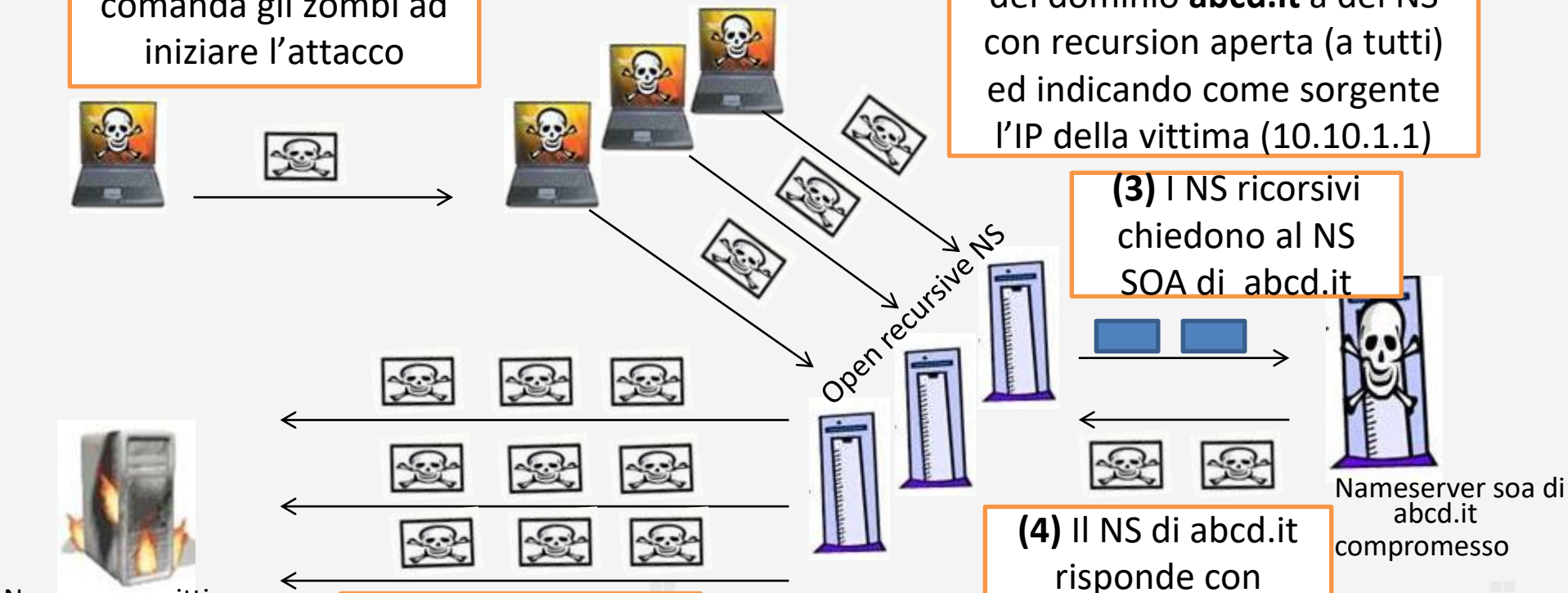
- Questo tipo di attacco **mira ad impedire il funzionamento del name server** (‘ denial of service’). Tipicamente vengono condotti **inviando, al server DNS vittima, delle query artefatte (‘malformed’)** che provocano in primo luogo la saturazione del collegamento del nameserver e in alcune vecchie versioni di Bind accessi errati ai dati in memoria o a volte il **crash del processo named**.
- L’attaccante lancia l'attacco da **sistemi (zombi)** su cui acquisisce il controllo non autorizzato mediante worm.
- Vengono sfruttati **NS con recursion aperta**.
- Viene fatto uso di amplificatori per aumentare il volume del traffico dell’attacco.
- Viene **“spoofato” l’indirizzo sorgente** del traffico **utilizzando l’IP della vittima.**

denial of service (2)

(1) L'attaccante comanda gli zombi ad iniziare l'attacco

(2) Tutti gli zombi inviano query per un determinato RR del dominio **abcd.it** a dei NS con recursion aperta (a tutti) ed indicando come sorgente l'IP della vittima (10.10.1.1)

(3) I NS ricorsivi chiedono al NS SOA di abcd.it



(4) Il NS di abcd.it risponde con pacchetti DNS da 4000byte

(5) I DNS con recursion aperta inviano le risposte di 4000byte al NS vittima

Denial of service: contromisure

- Mantenere **aggiornato BIND**
- Applicare regole (**rate-limit**) che filtrano **pacchetti DNS** particolarmiente **grandi (>512 bytes)**
- **Non consentire traffico** proveniente da **NS con recursion abilitata**
- **Abilitare la recursion (sui propri NS) solo per un numero ristretto di host (quelli della popria lan)**
- Definire ACL (sul router) che consenta traffico in uscita solo se i pacchetti contengono un IP sorgente appartenente alla propria LAN **(antispoofing).**

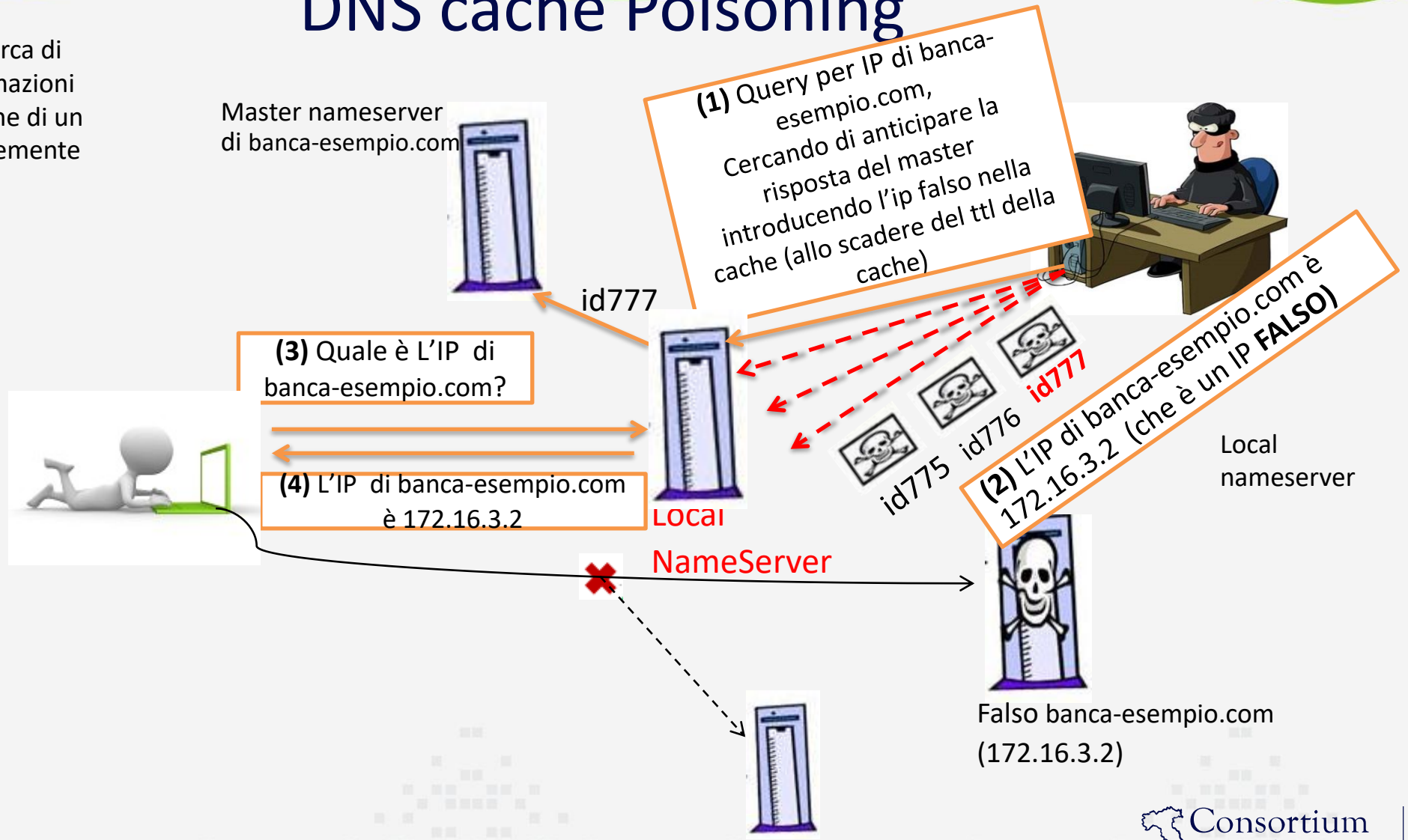
DNS Cache Poisoning

- Cache Poisoning – L'attaccante inserisce un **record DNS falso nella cache di un resolver** DNS, così che tutti gli utenti che lo utilizzano vengano reindirizzati verso un sito malevolo.
- Local Host Poisoning – L'attaccante modifica il **file *host* di un computer** per ingannare l'utente a livello locale.
- Man-in-the-Middle (MITM) DNS Attack – **un hacker** intercetta e **modifica le risposte DNS tra il client e il server** per reindirizzare il traffico.

- Cosa accade:
  - Un utente digita banca-esempio.com nel browser.
  - Il suo server DNS locale ha memorizzato un record alterato da un attacco di DNS poisoning.
  - Invece di risolvere correttamente il sito, reindirizza l'utente a un sito fraudolento che assomiglia a quello originale.
  - L'utente inserisce le proprie credenziali, che vengono rubate dagli hacker.

DNS cache Poisoning

L'attaccante cerca di mettere informazioni false nella cache di un NS precedentemente compromesso



Cache Poisoning: contromisure

- Mantenere **aggiornato BIND**
- Abilitare la ricorsione **(recursion solo alle macchine 'di fiducia', generalmente quelle della propria lan)**
- Evitare DNS pubblici non affidabili – **Utilizzare DNS sicuri come Google (8.8.8.8) o Cloudflare (1.1.1.1)**.
- **Svuotare regolarmente la cache DNS** con ipconfig /flushdns (Windows) o systemd-resolve --flush-caches (Linux).
- **Configurare**, quando possibile, **DNSSEC**.


DNSSEC

Introduzione al DNSSEC

- Descritto nel RFC 2065
- Il DNS Security Extensions aggiunge funzionalità di sicurezza al DNS
- Pensato per **garantire l'autenticità ed integrità dei dati**
- **Protegge** i propri nameserver **da attacchi specifici come il DNS Cache Poisoning**
- Il DNSSEC controlla le risposte ad ogni livello della gerarchia dello spazio dei nomi attraverso quella che è conosciuta come una "**chain of trust**", catena di fiducia. **Ogni nameserver** che supporta **DNSSEC** e ha zone firmate correttamente **contribuisce a garantire l'autenticità delle risposte**.

Cosa DNSSEC non può fare

- DNSSEC non può proteggere contro:
  - Packet **sniffing** (del traffico DNS)
  - Attacchi di tipo **DDoS**
  - IP **spoofing**
  - Alcune forme di phishing e pharming, ad esempio il **typosquatting** (tecnica fraudolenta che sfrutta gli errori di digitazione degli utenti)
- Inoltre non esegue controlli sui **trasferimenti di zona** dal nameserver master allo slave

34

DNSSEC: utilizzo delle firme digitali

- DNSSEC utilizza:
- **Criptografia asimmetrica** (chiavi pubbliche e private)
  - **ZSK**: Zone Signing Key (privata) — non visibile, utilizzata per firmare i RR all' interno del file di zona
  - **KSK**: Key Signing Key (pubblica) — la chiave pubblica **deve essere** invece **messa a disposizione dei nameserver che vogliono validare i dati relativi di quella determinata zona**
  - Le firme generate con la porzione di chiave privata possono essere validate con la porzione di chiave pubblica
- **DNSSEC non fa criptografia dei dati DNS**. Il contenuto del pacchetto DNS resta integro. I dati vengono firmati digitalmente su ogni livello gerarchico dello spazio dei nomi
- **Ogni DNS ricorsivo, con DNSSEC abilitato, deve essere configurato in modo che conosca la chiave pubblica dei namesever di root** (**avviene in fase di startup - Trust Anchor** del DNSSEC)

Come funziona DNSSEC

- Funzionamento è analogo a quello del DNS: il resolver invia una query ad un nameserver per un determinato RR.
- **Il nameserver interrogato**, se non autoritativo del RR richiesto, innesca il processo di risoluzione **richiedendo al NS autoritativo**, **oltre alle informazioni sul RR, anche la chiave (pubblica) DNSSEC associata alla zona**
- La chiave **KSK (pubblica) consente al nameserver richiedente di verificare che le informazioni ricevute**, relative al nome per il quale è stata fatta la query, **siano identiche a quelle presenti sul nameserver autoritativo** (evitando cosi gli attacchi cache poisoning)

Nuovi RR in DNSSEC

- RRSIG (Firma digitale)
- DNSKEY (Chiave pubblica)
- DS (Delegations Signer, parent-child)
- NSEC (prova di non esistenza)

Il record RRSIG

- Una volta abilitato DNSSEC, **ogni risposta DNS** (A, PTR, MX, SOA, DNSKEY, etc.) **sarà fornita con un RRSIG** ovvero una firma digitale dei RR.

- I record **RRSIG** vengono aggiunti al file di zona in automatico mediante specifici comandi per **firmare le zone**, non si fa manualmente e **sono generati mediante** la **chiave privata**

Il record DNSKEY

- **Per poter fare la validazione degli RRSIG in DNSSEC è necessaria la chiave pubblica** che deve essere quindi visibile, mentre la chiave privata è segreta
- Anche la chiave pubblica viene inclusa nel file di zona come record DNSKEY
  - È utilizzata per verificare i dati di zona (records) ad ogni query DNSSEC

Il record DS (Delegations Signer)

- **Il record DS** costituisce l'informazione verificabile (**è generato dalla chiave pubblica della zona figlio**) **conservata nella zona padre** come parte della catena di fiducia
- Uno degli aspetti fondamentali di DNSSEC è che **la zona padre, tramite il record DS, è in grado di garantire la validità della chiave pubblica della zona figlio** e quindi l'attendibilità di tutti i dati ricevuti forniti dalla zona figlio.

Consortium GARR THE ITALIAN EDUCATION & RESEARCH NETWORK

I record NSEC ed NSEC3 (Next SECure)

- Servono per dimostrare la NON esistenza del dato
- Svolgono lo stesso ruolo ma con presupposti leggermente differenti

- **Se** al DNS **viene chiesto l'indirizzo IP di un dominio che non esiste**, **esso restituisce una risposta vuota** (NXDOMAIN)
- Ciò rappresenta un problema se si desidera comunque autenticare la risposta, poiché non è presente alcun messaggio DNS da firmare.
- DNSSEC corregge questo problema aggiungendo **i record NSEC e NSEC3** che **consentono un denial of existence autenticato (risposta vuota autenticata)**

Registrare un nome a dominio con DNSSEC

2

Parent Zone
(.eu)

Record DS(.it)

1

3

File di Zona
(nostro dominio)

www

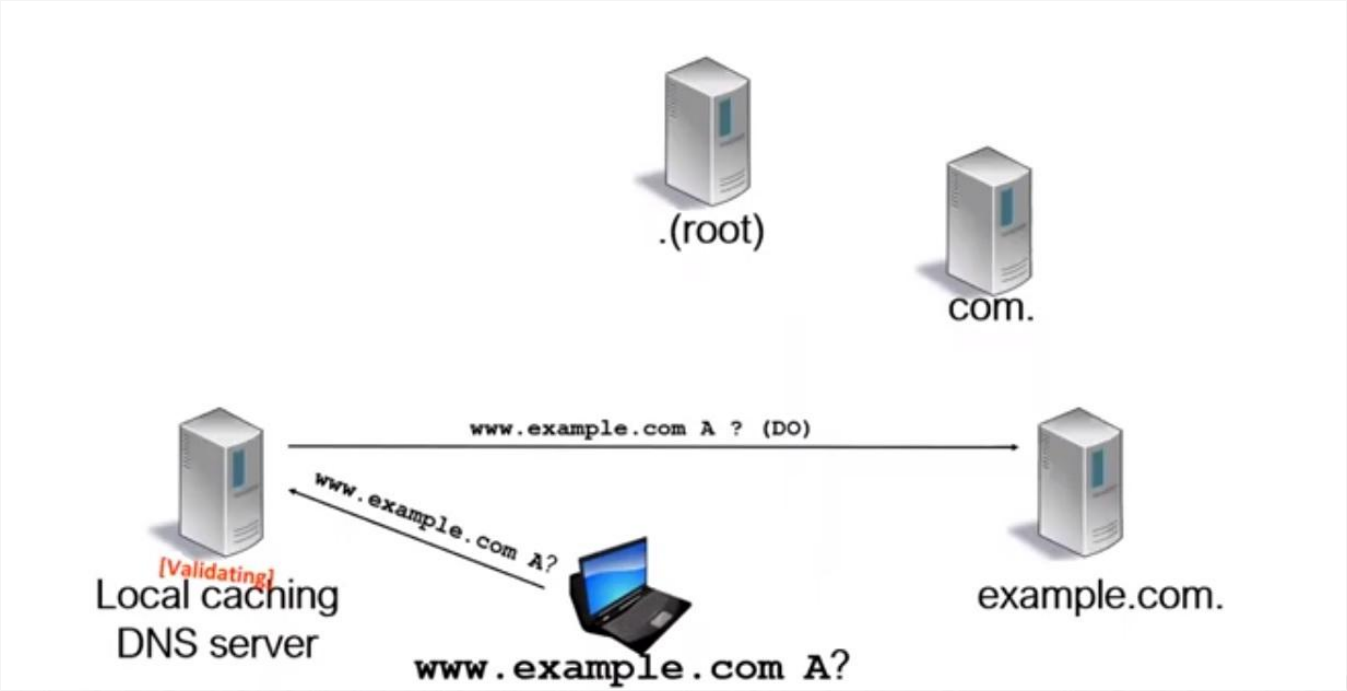
1 – Generazione delle chiavi (lo fa l'utente)
2 – Upload chiave pubblica (si crea il record DS) nella zona padre (lo fa il Registrar, in fase di Registrazione del dominio)
3 - Inserimento chiave pubblica nel file di zona (DNSKEY) + firma digitale dei RR mediante chiave privata (RRSIG)

- DNSSEC Resolution

DNSSEC Resolution (1)

- Una volta giunto a chiedere la risoluzione di www.example.com al nameserver autoritativo, il nostro nameserver invia nell'header del message DNS un nuovo bit: **il bit DO (DNSSEC OK)**

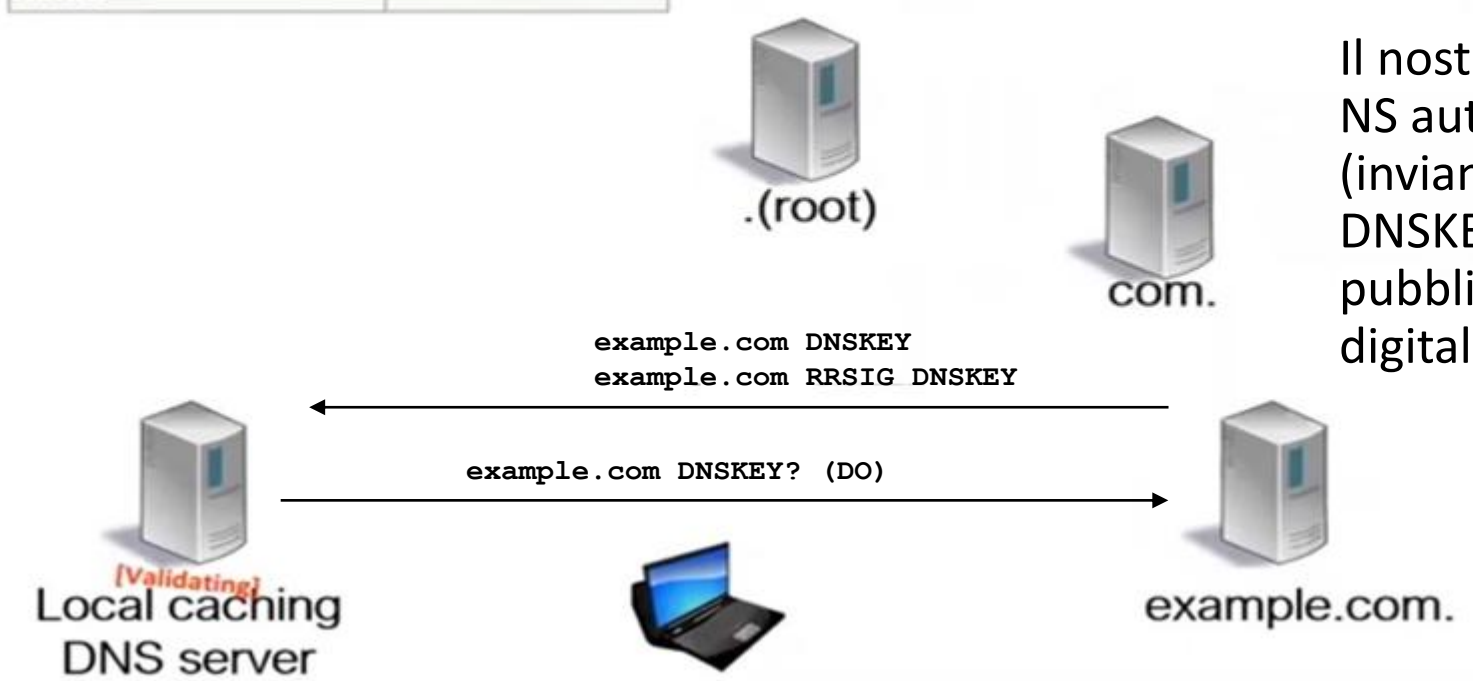
- Serve per segnalare al DNS autoritativo di example.com che il Local caching NS è predisposto a ricevere le firme digitali della zona (si comunica che DNSSEC è abilitato)



DNSSEC Resolution (2)

Se il client ha il bit DO (DNSSEC OK) a 1, il nameserver autoritativo e con DNSSEC abilitato, **fornisce** sempre un record di tipo **RRSIG (creato con la porzione privata)** in aggiunta al normale record DNS

Record	Function
www.example.com A	IPv4 Address
www.example.com RRSIG A	Signature ↑
example.com DNSKEY	Public Key
example.com RRSIG DNSKEY	Signature ↑



Il nostro nameserver richiede al NS autoritativo di example.com (inviando un DO) anche la DNSKEY, porzione di chiave pubblica(per decriptare la firma digitale).

DNSSEC Resolution (3)

- C’è un problema in tutto questo meccanismo:
- Le firme digitali ricevute per validare i RR richiesti, provengono dal nameserver autoritativo di example.com sul quale non abbiamo nessuna garanzia che non sia stato già manipolato

- Se l’attaccante può falsificare i dati del nameserver potrebbe essere in grado anche di falsificare le chiavi che stiamo ricevendo dal nameserver autoritativo

DNSSEC Resolution (4)

- Per **poter provare che la chiave pubblica per una determinata zona «example.com» è stata creata** ed inserita ***dall'effettivo responsabile*** della zona è necessario l'utilizzo di un nuovo record:
  - **DS: Delegation Signer**
  - Risiede nella zona padre (creato durante la registrazione del nome)
  - Contiene un hash (miscuglio) della chiave pubblica presente nella zona figlio.
    - **Per compromettere la chiave pubblica di una zona, l'hacker deve essere in grado di compromettere anche il record DS della zona padre**

- **Per verificare la validità della *chiave pubblica della zona figlio*, il resolver la confronta con il *record DS depositato nella zona padre***. Se corrispondono, il resolver può presumere che la chiave pubblica non sia stata manomessa, il che significa che può considerare attendibile il record proveniente dalla zona figlio.

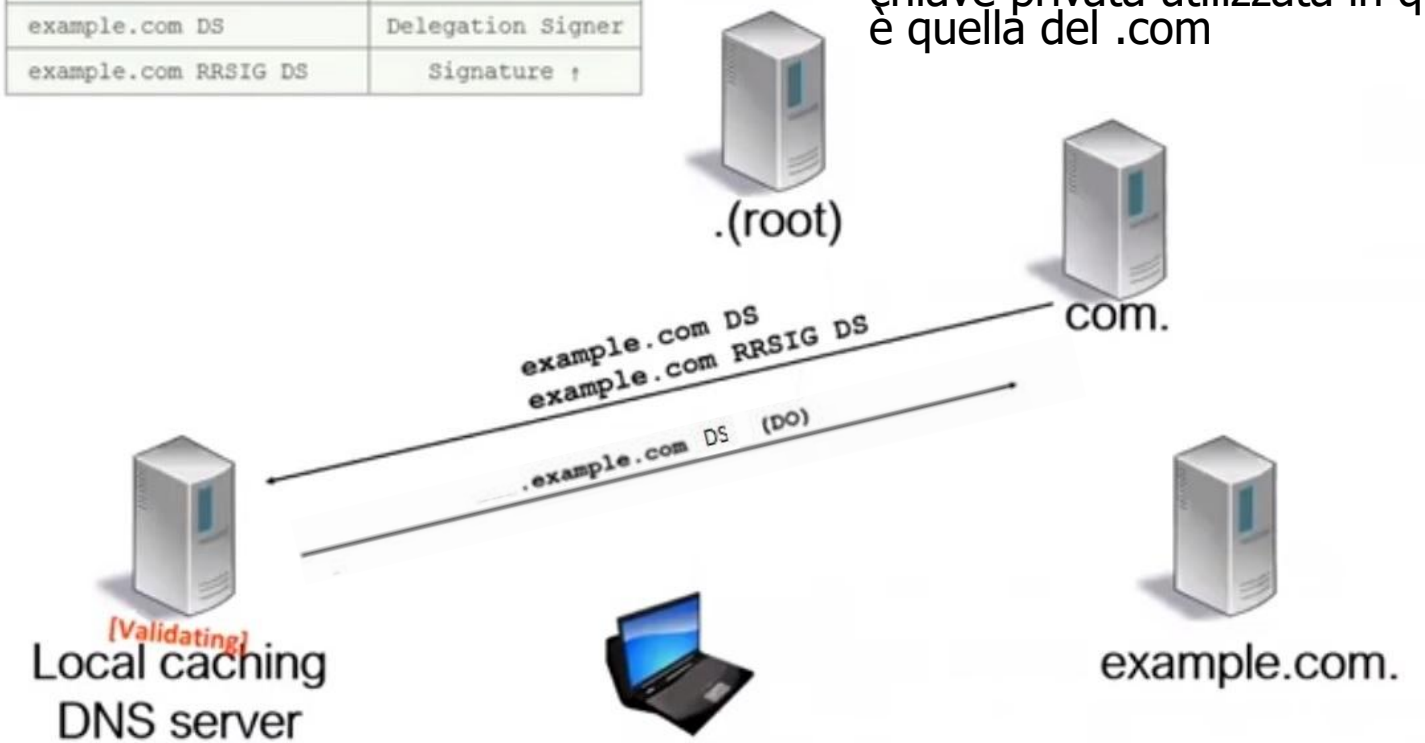
Consortium GARR | THE ITALIAN EDUCATION & RESEARCH NETWORK

DNSSEC Resolution (5)

Record	Function
www.example.com A	IPv4 Address
www.example.com RRSIG A	Signature ↑
example.com DNSKEY	Public Key
example.com RRSIG DNSKEY	Signature ↑
example.com DS	Delegation Signer
example.com RRSIG DS	Signature ↑

In questo modo riceviamo il **record DS** presente nella zona .com **e la sua firma digitale**.

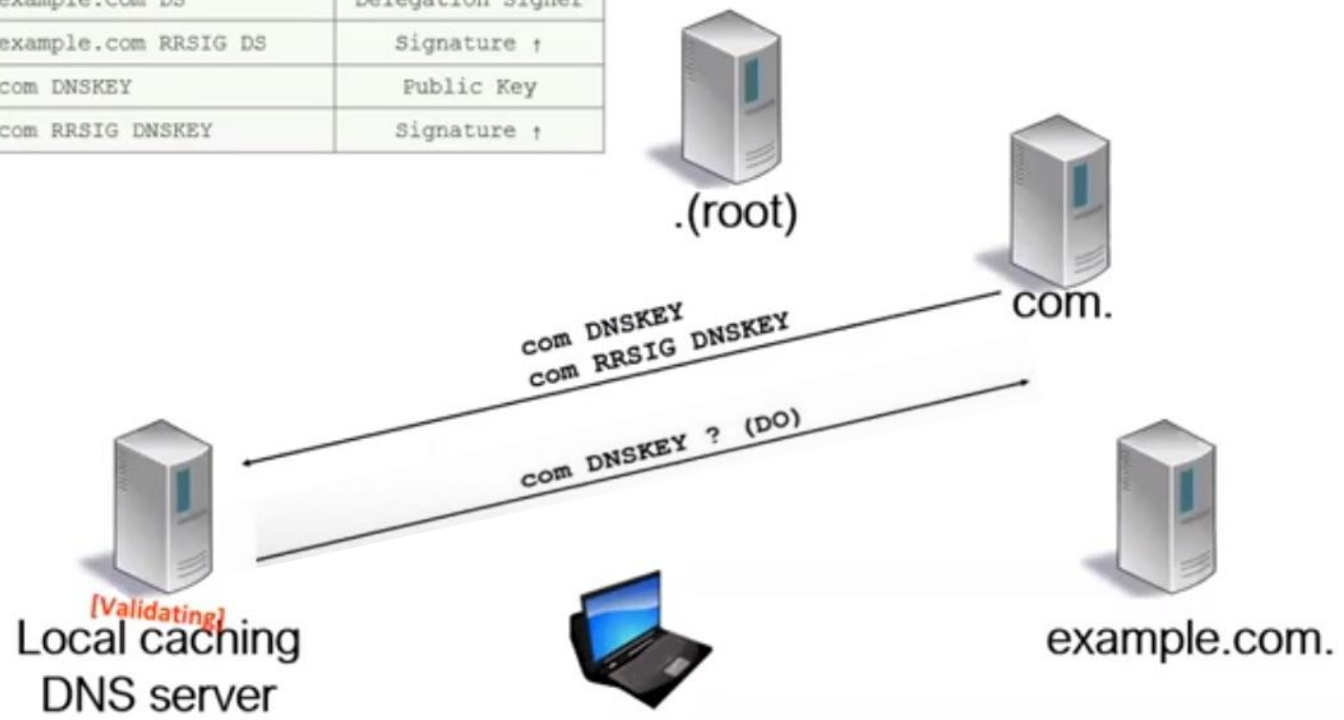
Ma quale chiave è stata utilizzata per generare quella firma(RRSIG)? La chiave privata utilizzata in questo caso è quella del .com



DNSSEC Resolution (6)

Record	Function
www.example.com A	IPv4 Address
www.example.com RRSIG A	Signature ↑
example.com DNSKEY	Public Key
example.com RRSIG DNSKEY	Signature ↑
example.com DS	Delegation Signer
example.com RRSIG DS	Signature ↑
com DNSKEY	Public Key
com RRSIG DNSKEY	Signature ↑

Ci serve la DNSKEY del .com per risolvere l'RRSIG

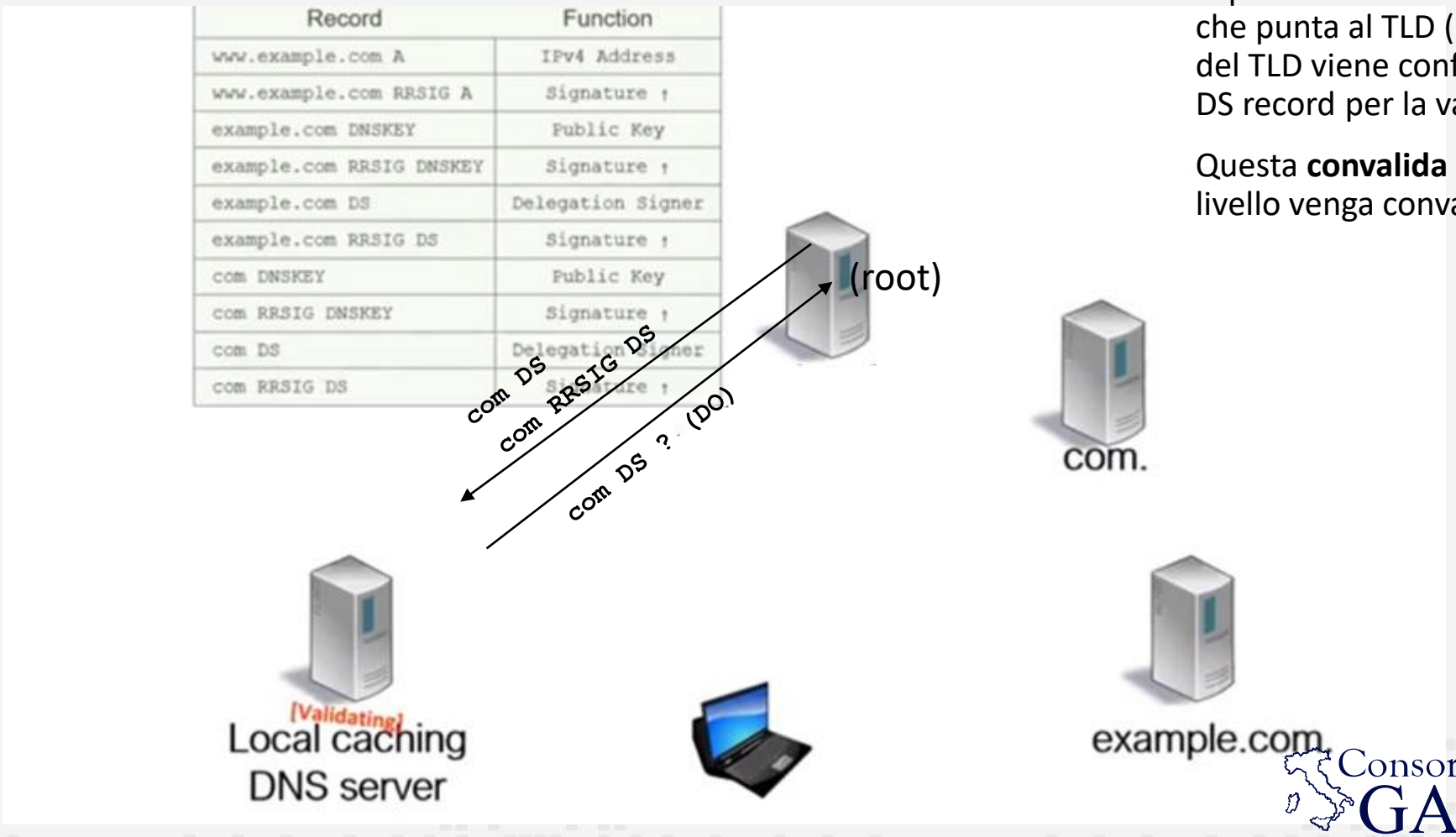


In questo modo quando il resolver DNS interroga il server di "example.com" e ottiene la KSK (pub), può confrontare la KSK con il record DS del **TLD** e assicurarsi che corrispondano.

(praticamente che la chiave pubblica ottenuta corrisponde a quella ottenuta dalla stringa crittografata)

DNSSEC Resolution (7)

Record	Function
www.example.com A	IPv4 Address
www.example.com RRSIG A	Signature ↑
example.com DNSKEY	Public Key
example.com RRSIG DNSKEY	Signature ↑
example.com DS	Delegation Signer
example.com RRSIG DS	Signature ↑
com DNSKEY	Public Key
com RRSIG DNSKEY	Signature ↑
com DS	Delegation Signer
com RRSIG DS	Signature ↑

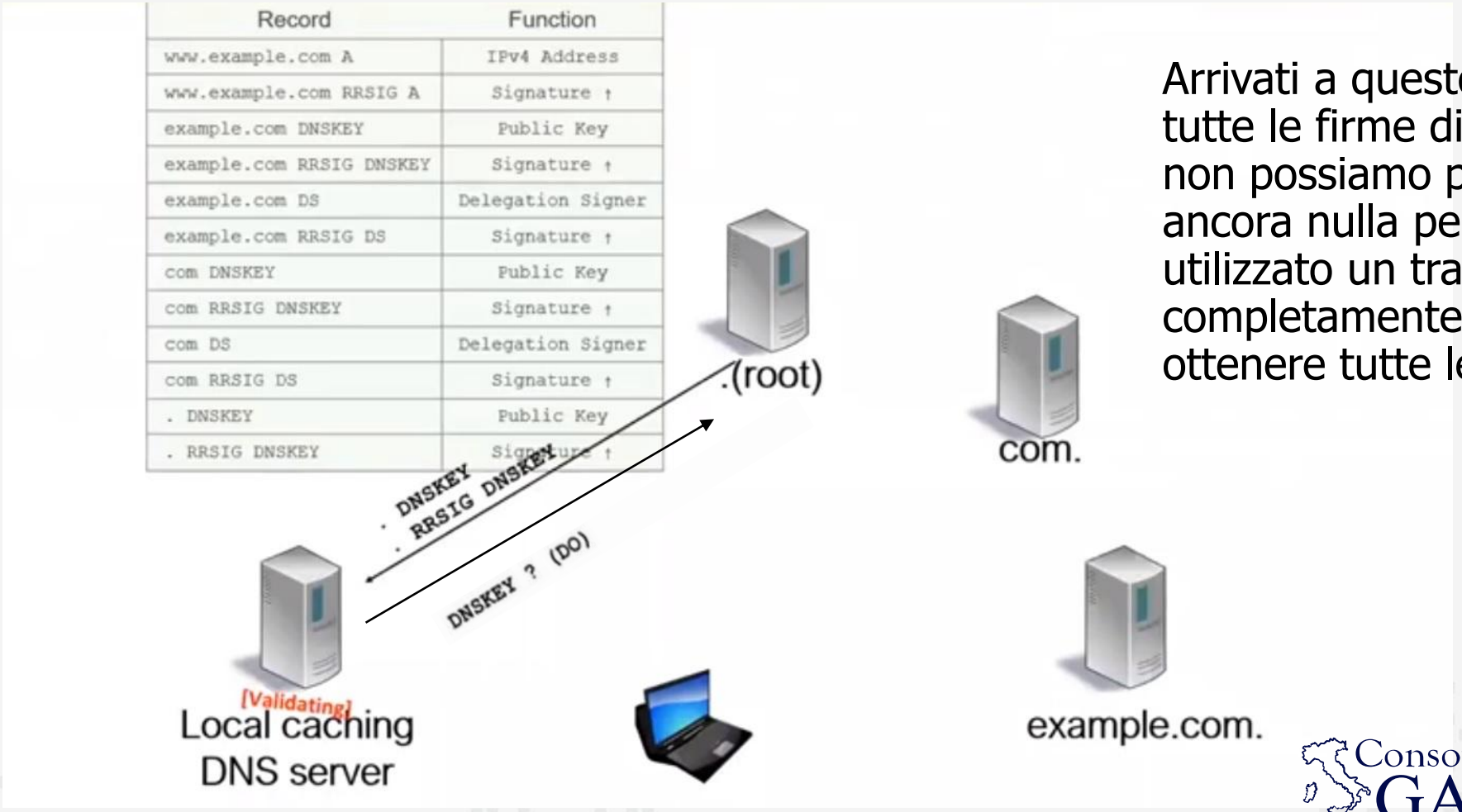


La stessa cosa succede al livello superiore. Il **root server** ha un DS record che punta al TLD (.com). La KSK (pub) del TLD viene confrontata con questo DS record per la validazione.

Questa **convalida a catena** fa sì che ogni livello venga convalidato.

DNSSEC Resolution (8)

Record	Function
www.example.com A	IPv4 Address
www.example.com RRSIG A	Signature ↑
example.com DNSKEY	Public Key
example.com RRSIG DNSKEY	Signature ↑
example.com DS	Delegation Signer
example.com RRSIG DS	Signature ↑
com DNSKEY	Public Key
com RRSIG DNSKEY	Signature ↑
com DS	Delegation Signer
com RRSIG DS	Signature ↑
. DNSKEY	Public Key
. RRSIG DNSKEY	Signature ↑

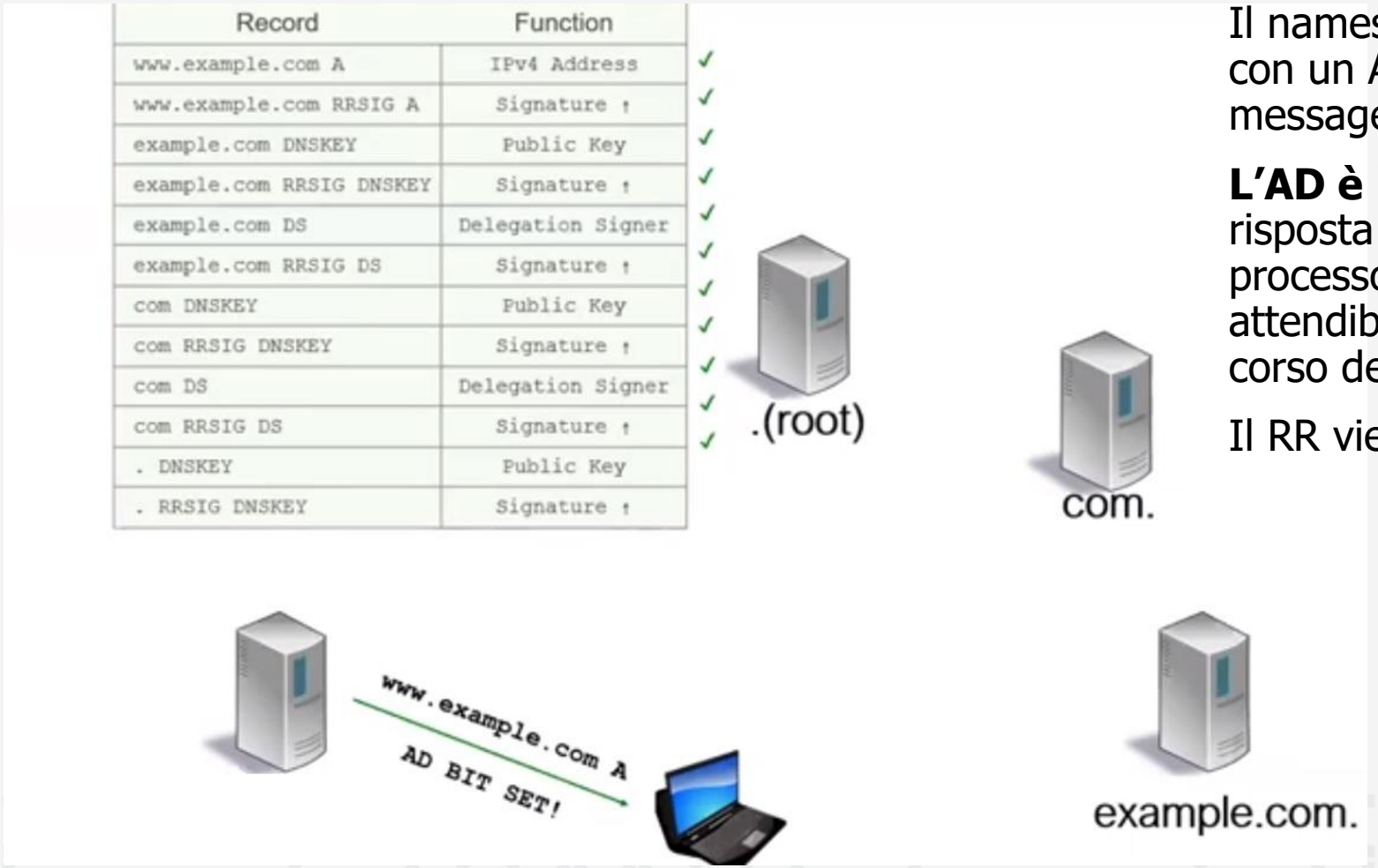


Arrivati a questo punto abbiamo tutte le firme di tutte le zone ma non possiamo provare in realtà ancora nulla perché stiamo utilizzato un trasporto completamente aperto (UDP) per ottenere tutte le firme

Trust Anchors

- **Non esiste una zona padre per la zona "." Quindi non abbiamo un record DS per verificare la chiave pubblica del NS di root**
- Chi garantisce per la chiave pubblica del nameserver di root?
    - **Il nostro nameserver**, se è configurato per il DNSSEC, **dispone già della chiave pubblica di root (Trust Anchor) che viene ricevuta in fase di startup di DNSSEC**
        - https://downloads.isc.org/isc/bind9/keys/9.11/
        - https://dnsinstitute.com/documentation/dnssec-guide/ch03s04.html
        - Ricevuta la **risposta dal root nameserver (DNSKEY), la confronta con quella di cui è già in possesso** (Trust Anchor)
    - Se le chiavi coincidono possiamo fidarci della risposta ottenuta dal root namserver
    - Quindi possiamo fidarci anche della risposta ottenuta sulla zona .com e su example.com
- Questa è nota come **"chain of trust"** in DNSSEC

DNSSEC Resolution (9)



Il nameserver locale risponde al client con un AD bit nell'header del message DNS

L'AD è l'Authenticated Data. La risposta ricevuta ha superato il processo di convalida quindi è attendibile e non è stata alterata nel corso del processo di risoluzione.

Il RR viene consegnato all'utente.

Considerazioni sull’attivazione di DNSSEC

- Complessità di implementazione e gestione
- Grandezza dei pacchetti DNS in risposta
- Grandezza dei file di zona - Generazione e caricamento
- Impatto delle query sulle perfòrmance di BIND
- Attenzione al furto delle chiavi
- Tempo validità delle firme digitali

Configurazione di BIND con DNSSEC

```
options {
                    directory "/var/named";
                    dnssec-enable yes;
                    key-directory "keys";
          };
logging {
    channel query_log {
         file "/var/log/named/named_querylog"
         versions 5 size 500M;
         print-time yes;
      };
          category queries {query_log; };
};

zone “prova.it" {
        type master;
        file "/etc/bind/prova-it-master.soa";
        allow-query { any; };
        allow-transfer { 90.148.80.2;
                         2001:760:1010::8; }; //consente di fare l’edit del file di zona come un file senza dnssec
              inline-signing yes;                // firma di nuovo automaticamente la zona
              auto-dnssec maintain;};
```

Generazione delle chiavi DNSSEC

- `# mkdir /var/named/keys`
- `# chown named /var/named/keys`

Directory per le chiavi

ZSK

- `# dnssec-keygen -a ECDSA-P256-SHA-256 -K /var/named/keys -n ZONE dnssec.example.com`
- `# dnssec-keygen -f KSK -a  ECDSA-P256-SHA-256 -K /var/named/keys -n ZONE dnssec.example.com`
- `# chown named /var/named/keys/*`

KSK

Modifica dei permessi, il processo BIND deve essere abilitato in lettura dei files che contengono le chiavi

Firma della zona e creazione dei record NSEC3 e DS

- `# rndc sign dnssec.example.com`

Firma il file di zona, quando non usiamo i comandi automatici nel «Named.com»

- `# rndc signing nsec3param 1 0 100  A5F7B1CD dnssec.example.com`

Aggiunge l'NSEC3 al file di zona

- `# dnssec-dsfromkey -2 /var/named/keys/Kdnssec.example.com.+013+38320.key`

Crea il record DS

(III) - Domande

- Quando un utente prova a contattare un host nel dominio lame (e cioè quando la delega nell'albero del DNS non è correttamente configurata) cosa succede? Vengono coinvolti i DNS di root?

- A cosa serve il Record DS?

- Il record DS corrisponde all'ultimo anello della Chain of Trust (catena di fiducia)?


RPKI

RPKI

Implementazione sulle risorse IP GARR

RPKI (Resource Public Key Infrastructure) è un'infrastruttura progettata per proteggere il sistema di instradamento di Internet, in particolare il protocollo **BGP (Border Gateway Protocol)**, da attacchi e configurazioni errate come il **prefix hijacking** (dirottamento di prefissi IP)

- **RPKI consente** ai titolari di indirizzi IP (come LIR, ISP o organizzazioni) **di dichiarare**, in modo critto-graficamente verificabile, **quale AS (Autonomous System) è autorizzato ad annunciare un certo prefisso IP**.

Benefici dell'RPKI sul routing (1)

- Consente di dimostrare se gli annunci di determinati prefissi provengono dal legittimo assegnatario

- Prevenire il routing hijacking
  - Un "BGP hijack" **consiste nell'annuncio di un prefisso pubblico IPv4/v6** da una rete ad un'altra **senza il consenso del titolare della risorsa**

- Il "BGP hijack" può avere diversi obiettivi:
  - **Fuorviare le forze dell'ordine** dalla reale origine dei pacchetti legati ad attività illegali
  - **Utilizzo temporaneo dello spazio di indirizzamento senza alcun costo** di manutenzione associato (es. abbonamento RIR)
  - **Impersonare reti e servizi di terze parti e Intercettazione del traffico** di reti di terze parti (es. attacchi man-in-the-middle)
  - **Interruzione delle comunicazioni IP tra** due o **più reti** di terze parti

Benefici dell'RPKI sul routing (2)

- Prevenire *mis-origination*
  - Prefissi originati per errore da altri AS
    - A causa di semplici errori di battitura in fase di configurazione può accadere di originare, dal proprio AS, prefissi che dovrebbero essere originati da altri AS

Componenti principali

- **ROA (Route Origin Authorization):** viene generato dal LIR o dal ISP, un documento firmato digitalmente che specifica *quale AS può annunciare un determinato prefisso IP*.
- **ROV (Route Origin Validation):** si implementa sui router e sui server di validazione (**validatori**), *software che* scarica e *verifica i ROA, presenti nei Repository dei RIR, per validare gli annunci BGP*

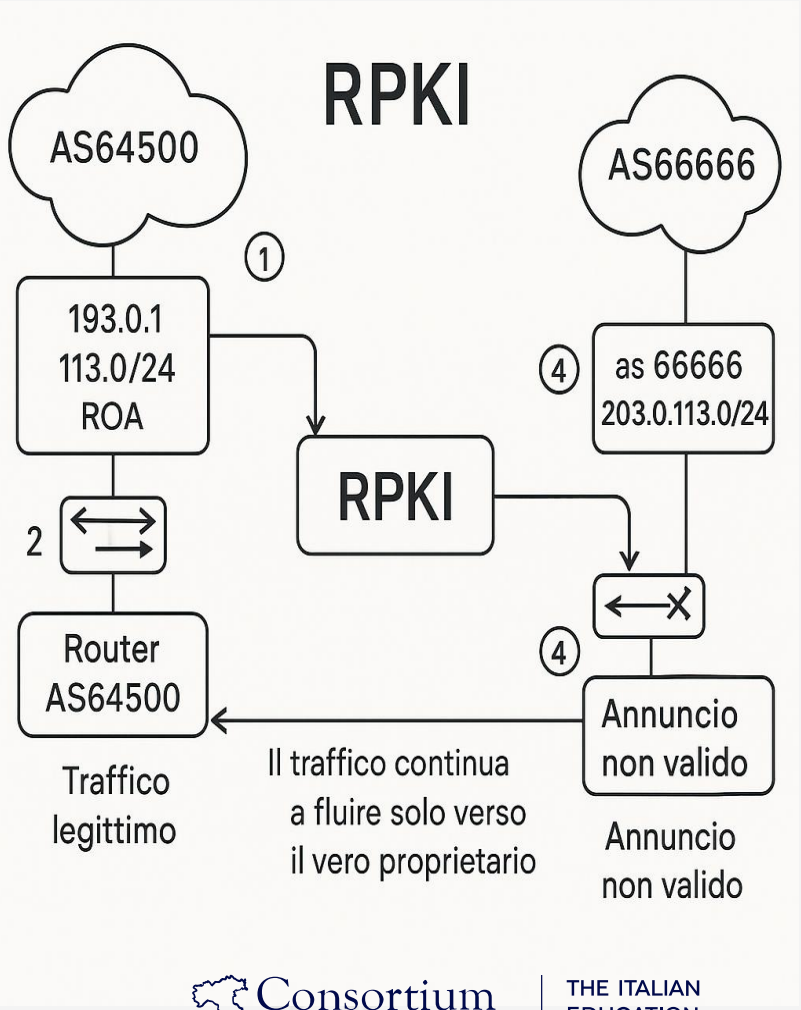
- Le **Autorità di Certificazione (CA):** di solito sono i **RIR** (come RIPE NCC, ARIN, etc.) che *rilasciano certificati associati alle risorse IP*.

Esempio: Scenario senza RPKI

- **Un operatore** di rete malevolo **annuncia al mondo un prefisso IP 203.0.113.0/24** che in realtà **non gli appartiene**.
  - **I router BGP** di altre reti accettano questo annuncio, pensando che sia legittimo, e **iniziano a inoltrare il traffico** verso l'attaccante.
- Questo è un **hijack**, e può causare:
  - Interruzione dei servizi
  - Intercettazione del traffico
  - Problemi di sicurezza

Esempio: Scenario con RPKI

- **Il proprietario legittimo del prefisso** 203.0.113.0/24 crea un **ROA** che dice:
  - "Solo l'AS64500 è autorizzato ad annunciare questo prefisso".
- Questo ROA è **firmato digitalmente e pubblicato** tramite infrastruttura RPKI.
- I provider e i gestori di reti che usano **validatori RPKI** scaricano i ROA e controllano ogni annuncio BGP.
- Quando l'**attaccante** (AS66666) prova ad annunciare 203.0.113.0/24:
  - I router **vedono che AS66666 non è autorizzato**.
  - L'annuncio viene **scartato automaticamente**.
- Il traffico continua a fluire **solo verso il vero proprietario**.
- Grazie a RPKI, l'attacco fallisce, e il traffico non viene dirottato.



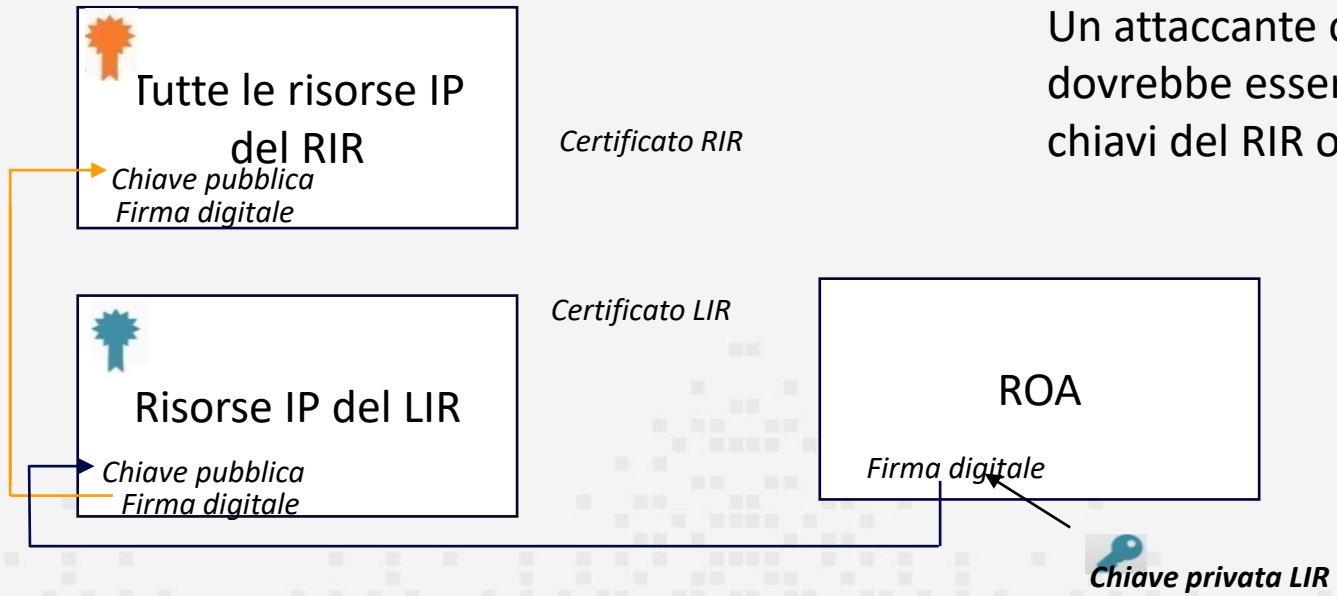
Consortium GARR THE ITALIAN EDUCATION & RESEARCH NETWORK

RPKI Chain of Trust

- **Ogni RIR dispone di un certificato di root associato a tutte le risorse IP**. Il RIR ha una **chiave pubblica**, una **chiave privata** ed una **firma digitale**.
- La chiave privata di root è utilizzata per firmare le risorse IP di ogni sponsoring LIR.

I Route Origin Authorisations (ROAs) (1)

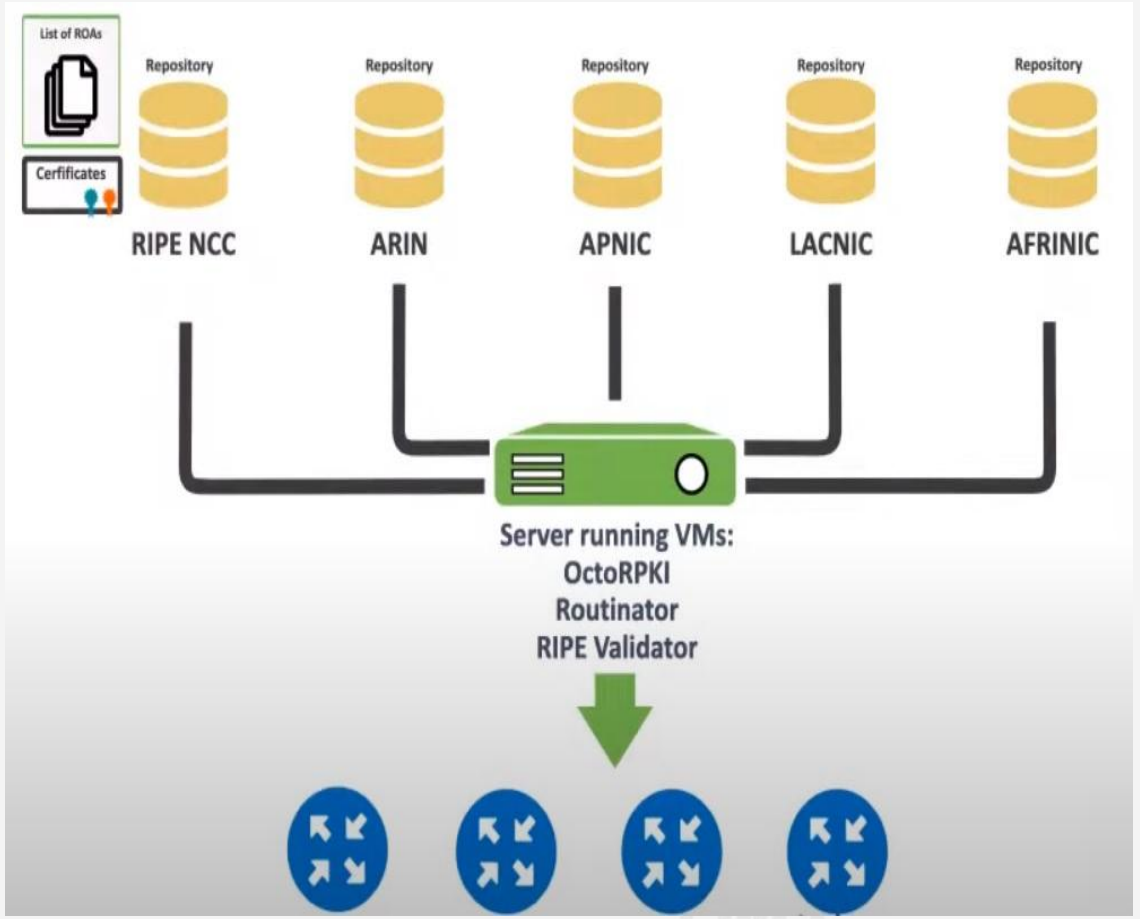
- Il LIR crea **un ROA per ogni classe di indirizzamento gestita, il Roa viene depositato (entro circa 5 minuti dalla sua creazione) in un repository del RIR**, che consiste in un grande database che include tutti i ROAs e gli altri oggetti RPKI
  - Ci sono vari repository, uno presso ogni RIR
- **ROV - Tutti i dati inclusi nei repository sono collezionati dai così detti Validators**:
  - *Si tratta di un software da installare su dei server facenti parte dell'infrastruttura di rete geografica della sponsoring LIR* (nel nostro caso nei PoP GARR)



Un attaccante che volesse falsificare delle informazioni, dovrebbe essere in grado di compromettere anche le chiavi del RIR oltre a quelle del LIR.

I Route Origin Authorisations (ROAs) (2)

- I validator **servono per prendere delle "*decisioni sul routing*"**
  - I software per erogare il servizio di *validator* sono open source e sono sviluppati da diverse organizzazioni (tra cui RIPE NCC)
- I validator sono collegati ai routers di Backbone della rete
- **Forniscono informazioni ai router** stessi per consentirgli di prendere decisioni sul routing BGP



Come creare i ROAs

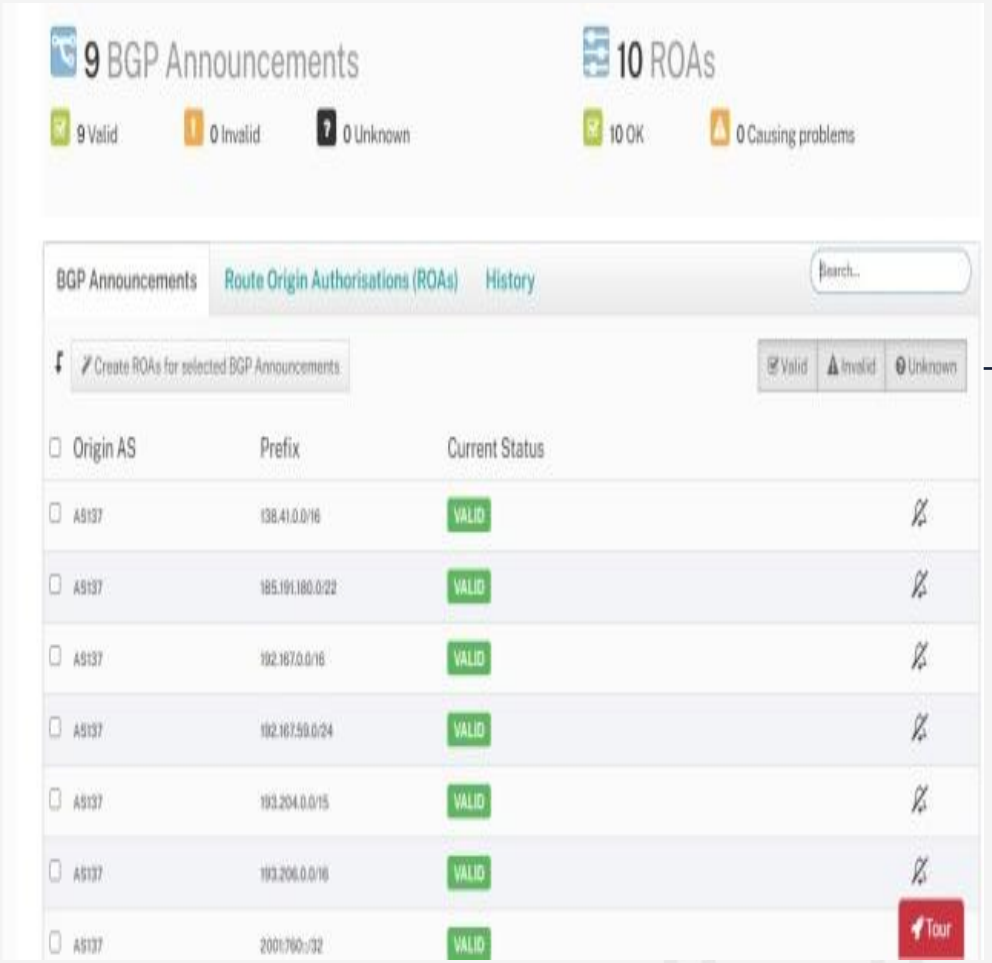
- Dunque per un LIR, *il **primo step*** per beneficiare delle potenzialità *dell'RPKI è **definire i ROAs per tutte le classi di indirizzamento** gestite*

- La creazione dei ROAs può avvenire:
  - **mediante portale messo a disposizione dal RIR di appartenenza;**
  - autonomamente. Lo sponsoring LIR può decidere di svolgere la funzione di Certificate Authority, creare e conservare la propria chiave crittografata.
    - I LIR che decidono di gestire i ROAs autonomamente fanno uso del pacchetto **software Krill**:
      - https://labs.ripe.net/author/alexband/krill-gains-powerful-roa-management-based-on-bgp-routing/
    - La lista di chi ha deciso di svolgere autonomamente la funzione di Certificate Authority è disponibile a questa URL:
      - https://rpki-validator.ripe.net/trust-anchors

Creazione dei ROAs su interfaccia WEB di RIPE (1)

- Per la creazione dei ROAs, il **GARR-LIR ha utilizzato l'interfaccia WEB** messa a disposizione **da RIPE NCC**
- **Una volta collegati** all'interfaccia WEB, **le informazioni visualizzabili sono esclusivamente quelle relative agli annunci BGP dei prefissi amministrati dallo sponsoring LIR, collezionate da RIPE** mediante i suoi **route collectors**
- Appena collegati all'interfaccia, lo stato degli **annunci BGP elencati sarà *Unknown***: il **prefisso** dell'annuncio **non** è **coperto da un ROA** esistente
- Una volta creato il ROA per un annuncio, lo **status cambierà in *Valid*** perché l'annuncio BGP risulterà essere coperto da un ROA

Creazione dei ROAs su interfaccia WEB di RIPE (1)



Creazione dei ROAs su interfaccia WEB di RIPE (3)

- Quando si crea un **ROA**, occorre definire il paramentro ***max length*** che stabilisce la grandezza massima del prefisso attribuibile al ROA affinchè possa essere considerato valido in BGP
  - Se si definisce in BGP un annuncio più specifico di quanto consentito dalla lunghezza massima impostata nel ROA, lo status del ROA risulterà *Invalid*
  - **La lunghezza massima del ROA deve corrispondere alla subnet con cui si origina il prefisso in BGP**. Se si decide di configurare in BGP annunci più specifici dello stesso prefisso, occorrerà definire ROA specifici per ogni nuovo annuncio

ROAs per prefissi annunciati da più di un AS

- E’ anche possibile eseguire degli ***overlaps*** sui ROA quando è necessario annunciare un **prefisso da più di un AS**
  - In questo caso occorrerà creare ROA differenti per lo stesso prefisso annunciato dai differenti AS
  - L’eventuale ROA in overlap può essere creato solo dal reale assegnatario della classe di indirizzamento, non dall’assegnatario del secondo AS

Il Route Origin Validation (ROV) (1)

- **ROV è il processo**, lato router BGP, **di verificare se un annuncio BGP ha** un'origine valida in base ai **ROA pubblicati**.
- In sintesi, al momento, ci sono **cinque Repository** gestiti dai 5 RIR (RIPE NCE, ARIN, APNIC, LACNIC e AFRINIC).
- Un **validator** presente nella rete **comunica** con questi **repository** e **scarica un database ROA** per popolare la cache. Si tratta di una **copia unificata** della RPKI, che viene periodicamente recuperata/aggiornata direttamente o indirettamente dalla RPKI globale.
- **I Validator confrontano gli annunci BGP ricevuti dalla rete, con la tabella RPKI e comunicano il risultato ai router per prendere una decisione sicura.**

Il Route Origin Validation (ROV) (2)

- Il **protocollo** utilizzato per lo scambio di informazioni tra *Router* e *Validator* è definito nell'RFC6810 ed è noto come **"RTR"** (RPKI to Router).

- L'implementazione del ROV non è stata ancora predisposta sui router di Backbone GARR

  Il software ***validator* più utilizzato è il «*routinator*»**, necessita di meno risorse di sistema rispetto ad altri software *validator* ma non dispone di un'interfaccia grafica **(solo command line)**

  https://github.com/NLnetLabs/routinator

RPKI: esempi pratici e monitoring

- Per **verificare se gli annunci BGP del vostro ISP sono validi** è possibile fare un check da questo portale:
  - **https://isbgpsafeyet.com/**
- Il portale fornisce anche **esempi pratici sul BGP hijack** ed informazioni sull'infrastruttura RPKI e molto altro…

- Per vedere **l'andamento dei ROA nel tempo**:

  - **https://rpki-monitor.antd.nist.gov/**

Fine – Parte II