

# DNS for Fun and Profit

Roy Arends  
Telematica Instituut  
roy@dnss.ec

Peter Koch  
DENIC e.G.  
koch@denic.de

12. DFN-CERT Workshop  
Hamburg  
03. März 2005

## Agenda

- DNS Finally Secure – DNSSEC Status
- The Protocol and Beyond – DNS Fingerprinting
- Security by Obscurity? – IP6 .ARPA Side Effects
- Open Relays and Open Resolvers – Anything over DNS

# DNSSEC Status Update

## DNSSEC Status

- RFC 2065 – January 1997
- RFC 2535 – March 1999
- DNSSEC-bis – IESG [approved](#) September 2004
- –bis support in BIND 9.3, NSD
- Testbeds, Secure Islands, (DLV)

## DNSSEC-bis – What's New?

- Protocol changes
  - Limited scope for KEY records
  - New DS record type
  - Type Code Rollover
  - New NSEC data format
  - EDNS0 support mandatory
- The zone walking problem
- Key management

## KEY Scope Limited

- KEY RR to carry (public) DNSSEC keys
- ... and others
- Problems
  - DNS [subtyping](#) problem – cannot ask specific questions
  - Signing KEYS you don't understand?
- $\rightsquigarrow$  KEY restricted to DNSSEC keys only
- Other applications (SSH, IPsec) may use dedicated (new) RR types

## Delegation Signer (DS) Record

- SIG at parent vs. SIG at child debate
- Do neither – insert one level of indirection
- DS contains signed hash of **Key Signing Key**
- $\rightsquigarrow$  Easier parent initiated key rollover
- KSK signs **Zone Signing Key**
- ZSK (or ZSKs) signs zone data
- $\rightsquigarrow$  Easier child initiated key rollover
- KSK and ZSK both DNSKEY RRs at the child zone apex

## Type Code Rollover

- *Jakob's Bug*: trouble with NXT after invention of DS
- New codes and mnemonics

old	KEY	SIG	NXT
new	DNSKEY	RRSIG	NSEC

- Internal structure remains (mostly) unaltered

## Zone Walking

- NSEC RR

```
example.net      NSEC  www.example.net  MX NS SOA RRSIG NSEC  
www.example.net NSEC  example.net     A  MX RRSIG NSEC
```

- Chaining through the zone – even with AXFR disabled
- Problem at the Registry level (privacy, data protection)
- $\rightsquigarrow$  *online signing*
- $\rightsquigarrow$  NSEC successor (probably hash based)

## DNSSEC Deployment

- Latest versions of BIND 9 and NSD support DNSSEC-bis
- Testbeds, workshops, operational recommendations
- EPP support in development
- Tutorials available (e.g. RIPE NCC)
- Registries are actively developing procedures
- Root signing is still under discussion
- Early deployment approaches

# DNS Fingerprinting

## Why?

- Built with *surveys* in mind
- Mostly interested in the DNS landscape
- You know `version.bind` `TXT` `CH`?
- You disabled it?

## How?

- Unspecified bogus data handling
- Incorrect handling of proper data
- Implementations have bugs
- Implementations fixed bugs
- Have (stopped having) features

## Fingerprinting Requirements

- Nothing may break
- Independent of data served
- Independent of config
- Least possible queries
- Least possible log entries

## How? (2)

- DNS message has 16 bits in header
- We use 15 bits (not QR bit (more later))
- DNS query for . (root domain), QTYPE A, QCLASS IN

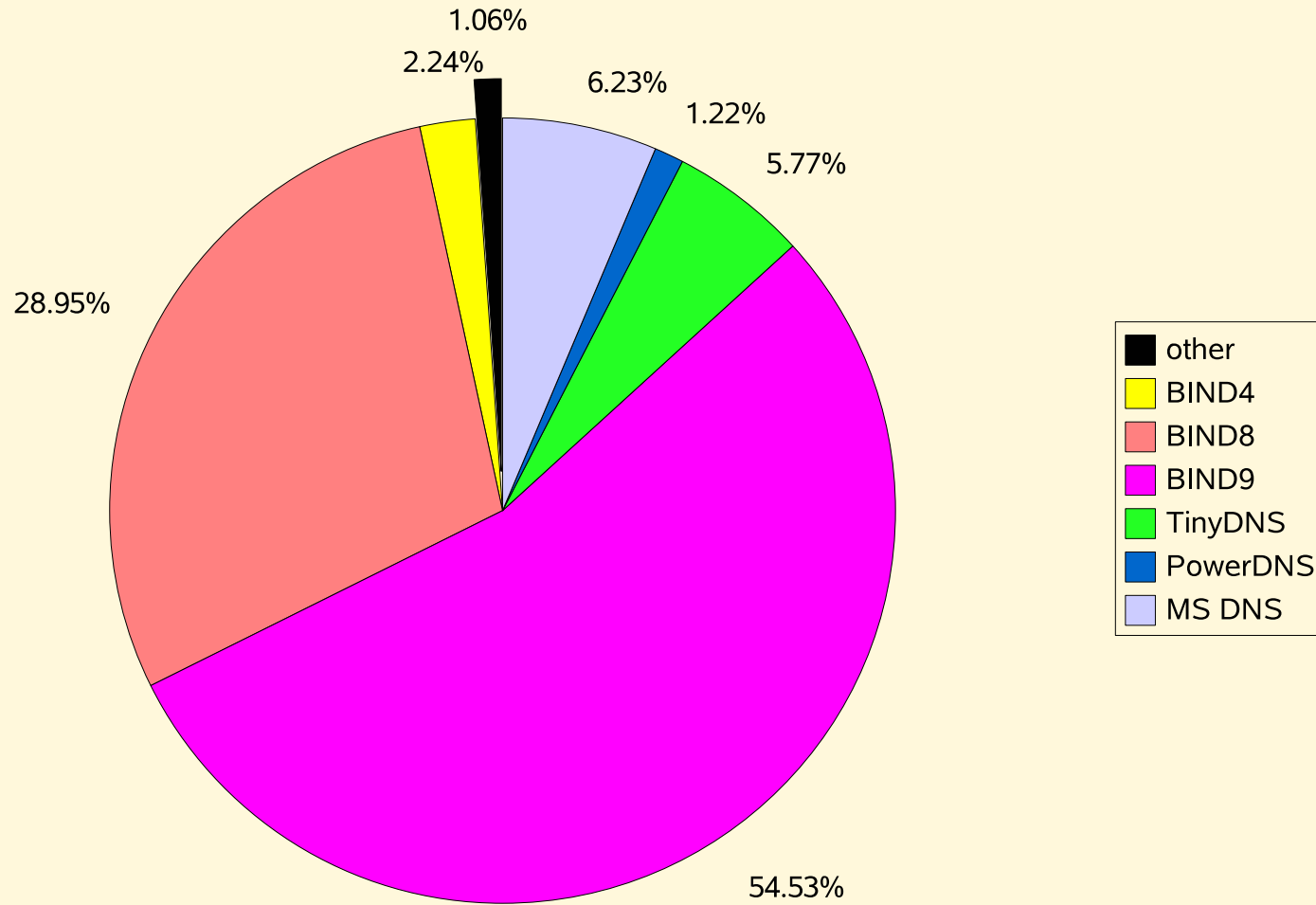
## How? (3)

- Lab setup:
  - BIND 8, BIND 9
  - MS DNS
  - djbdns
- Recorded all received responses in a matrix
- Some matrix crunching

**And the results are ... ?**

- VGRS-ATLAS
- BIND4,8,9
- eNom-DNS
- MARADNS
- MyDNS
- Nominum ANS,CNS
- NonSequitur DNS
- Pliant DNS Server
- PowerDNS
- QuickDNS
- Simple DNS plus
- javadns jnamed
- Nomde DNS tunnel
- Viking DNS server
- small HTTP server
- 4d WebSTAR
- Cisco Network Registrar
- NSD1,2
- DNS4me
- TinyDNS
- TotD
- UltraDNS
- pdnsd
- Rbldnsd
- Oak DNS
- Posadis
- Yaku-NS
- sheerdns
- dproxy
- dnrd
- JDNSS
- RaidenDNSD
- WinGate DNS
- dents
- Incognito DNS Commander
- MS Server NT4,2000,2003
- Net::DNS::Nameserver
- DeleGate DNS proxy
- Netnumber ENUM server
- Runtop Implementation
- Mikrotik Implementation
- Axis Video Network Implementation
- Fasthosts Envisage DNS server
- Ascenvision SwiftDNS
- Nortel Networks Instant Internet
- Nortel Networks Alteon ACEswitch
- Aethra ATOS Stargate ADSL
- 3Com Office Connect Remote
- Netopia Implementation
- Tzolkin DNS service
- jdns javadns service

# May 2004 Survey on DE



## What Does **Not** Help

- Active load balancers
- Firewalls check queries (cp-fw1-ai)
- Forwarders

## DNS Message Header: Extras

- QR bit 0: request
- QR bit 1: response
- Some implementations responded to responses (see nisc 758884)
- Most impls have been **fixed** (but not all)
- Can cause loops or query storms

## DNS Message Header: Extras(2)

- Some firewalls do reverse lookups of incoming DNS queries
- Some do reverse lookups of all UDP messages
- If you own the reverse space:  
reconnaissance method: *Hi firewall, I can see you :D*  
or just blame somebody else: spoof source address, its UDP remember?
- **TIP:** switch off all DNS lookups in your firewall. It is a denial of service method

## **IP6 .ARPA Side Effects**

## IPv6 Properties

- /48 assignment, 65536 /64 subnets,  $\leadsto 2^{64}$  addresses ( $10^{19}$ ) each
- (Port) scanning infeasible
- Addresses can be **hidden** ...
- ... well, *not really*
- Information leaks:
  - Address generation (Vendor ID)
  - Logs, traces
  - DNS on the wire queries
  - AXFR, NSEC walks



## Empty Non-Terminals

- `example.net` (SOA, NS, ...)
- `www.empty.example.net` (A, AAAA)
- `empty.example.net` may be *empty*
- Query yields **NOERROR** and **empty answer section**
- ...BIND 9 bugs notwithstanding

## Searching for 2001:DB8::42

```
0.0.0.0.0.0.0.0.0.8.B.D.0.1.0.0.2.IP6.ARPA
0.0.0.0.0.0.0.0.0.8.B.D.0.1.0.0.2.IP6.ARPA
0.0.0.0.0.0.0.0.0.8.B.D.0.1.0.0.2.IP6.ARPA
[... ]
0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.B.D.0.1.0.0.2.IP6.ARPA
0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.B.D.0.1.0.0.2.IP6.ARPA
4.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.B.D.0.1.0.0.2.IP6.ARPA
2.4.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.B.D.0.1.0.0.2.IP6.ARPA
1.0.0.0.0.0.0.0.0.8.B.D.0.1.0.0.2.IP6.ARPA
```

## Analysis

- Address space enumeration is feasible given IPv6 reverse mapping
- (Why) is this a **threat**?
- (When) is this a **problem**?



# Open Resolvers

## Shift in Security Consensus

### **Trend then:** Open relays

- Considered *not done* to operate closed relays
- Bandwidth, availability, infrastructure were expensive
- Service sharing was the gentlemen's approach

### **Internet now:**

- Considered very bad to operate open relays
- Bandwidth, availability, infrastructure not expensive
- Service sharing is considered security nightmare

## The Analogy

### Internet then:

- closed resolvers *not done*
- General view of DNS: availability is a must
- More users for a resolver: more efficient cache usage

### Internet now:

- It seems that focus **has not changed**
- Bulk of the authoritative DNS servers offer recursion

## Why is this Bad?

- Cache poisoning
- Cache probing
- DoS on the visibility of domains
- Store and forward bulk data

## Cache Poisoning

- Done by trial and error
- Open resolver increases the risk
- Simple test: when does the `widowupdate.example.net` record expire?
- Then: send a query to the resolver for `widowupdate.example.net`
- Now: hose the resolver with responses (Meanwhile DoS the authoritative servers for `example.net`)

## Cache Probing

- Check some cache for specific data
- Is some user looking at pr0n? Worse?

## Accidental DoS

- Resolving for the world will increase cache size/log size significantly
- This is accidental DoS  $\rightsquigarrow$  Service for *real users* slows down
- Users experience more latency – *network is slow*

## Black Hat DoS

- Reconnaissance: Scanning a /16 (class B) network for open resolvers is trivial
- Simple way: send DNS messages – wait for responses

## ***Intelligent Black Hat DoS***

- Send DNS messages with spoofed source address
- Query for a specific domain **under your control**
- Wait for incoming queries at the (your) server
- Much **faster**, much **harder to detect**

## Recruitment

- Now a Black Hat has a bulk of servers that it can use to resolve (redirect messages)
- These servers were **not recruited**
- They were politely *asked to participate*

## Result

- Now use the bulk of resolvers (say 32K) to query for random names under \$VICTIM\_DOMAIN
- Authoritative servers for \$VICTIM\_DOMAIN get hosed by queries
- Result: \$VICTIM\_DOMAIN is virtually disconnected
- Of course, hosing/DoSing higher level domains is much worse!
- These attacks currently happen *as we speak*

## Defense

- There is hardly any defense against these class of attacks
- Basically, the only defense is: [close the open resolvers!](#)

## Store and Forward

- Uses proper DNS messages to encapsulate bits of data
- Caches will store these bits of data for future use!
- Think streaming!
- Think bit-torrent seeds!
- Hard to detect
- Hard to defend against
- Simple defense: [close the open resolvers](#)

## Close Open Resolvers!

- Resolver can either ...
  - send back REFUSED
  - drop the query as a whole
  - (should not send back a referral to the root)

