IPv6-Only and DNS[SEC|64]

Jen Linkova

furry13@gmail.com

RIPE72, May 2016
Traditional Dual-Stack Network
Taking IPv4 away....
The Problem

IPV4 Network

IPv6-only Network

www.example.net
192.0.2.10

Router

IPv6-only user
2001:db8::2

no communication possible as
the host does not have IPV4
address to initiate a
connection to IPV4 address
and there is no v6 address to
connect to...:(((
The Solution: DNS64
DNS64 + DNSSEC (Validating Client)

Validating client

DNS64

www.example.net AAAA?

www.example.net AAAA?
www.example.net A?

www.example.net
AAAA 64:ff9b::192.0.2.10

AAAA: "no such name"
A: 192.0.2.10

NS for v4-only example.net
What is 'Validating Client'? 

- **Security-aware** resolver: accepts/understands DNSSEC security RRs.
  - "DNSSEC OK" (DO) bit set to '1'
- **Validating Resolver**: performs validation using DNSSEC security RRs
  - “Checking Disabled” (CD) bit
    - CD = 1 instructs server to disable validation (client will validate)
DNSSEC and DNS64 AAAA Synthesis (RFC6147)

For all IPv4-only names?

…..or…..

For DNSSEC-enabled names ONLY?

Some DNS64 MAY NOT return AAAA for IPv4-only DNSSEC-enabled names (e.g. BIND9: break-dnssec = yes)

For all IPv4-only names?

…..or…..

For DNSSEC-enabled names ONLY?
Standards vs. Implementation: \( DO = 1, \ CD = 1 \)

**RFC 6147 (DNS64)**

- Both DO and CD bits are set: DNS64 MUST NOT perform synthesis
  - not 100% clear if it applies if DNSSEC RRs are available or not
- Validation behind the DNS64: the validator must know how to perform the DNS64 function itself

**Reality**

Some DNS64 **perform** synthesis in the absence of DNSSEC RRs

```plaintext
furry@Wintermute:~> dig +dnssec +cdflag www.amazon.com aaaa +short
64:ff9b::36ef:1a80
furry@Wintermute:~>
```
Problem Space for Validating Clients

- DNSSEC-enabled domains
- IPv4-only names

OR

- Problem Area
- IPv4-only names

"Relaxed" DNS64 Implementations

"Strict" RFC6147 Implementation
In God We Trust, All Others Bring Data

Or

How Big is the Problem?
IPv6 & DNSSEC Adoption (Alexa 1M)

Alexa 1 000 000 web site names

16788 names with signed A/AAAA

56610 names with AAAA RRss
16788 names with signed A and/or AAAA

13324 IPv4-only names

3464 IPv6-enabled
IPv6 & DNSSEC Adoption (Alexa 1M)

IPv6 Adoption
- 5.7% of all sites
- 21% of DNSSEC-enabled sites

DNSSEC Adoption
- 1.7% of all sites
- 6% of IPv6-enabled sites
incompatible with ANY DNS64 implementation

incompatible with strict RFC6147 DNS64 implementation
Don’t Panic!

...just enable IPv6...
Validating Stub Resolvers & DNS64: Solution

Discover NAT64 prefix to perform DNS64 (RFC7050)

```plaintext
furry@Wintermute:~> dig +nacdflag ipv4only.arpa aaaa +short
64:ff9b::c000:aa
64:ff9b::c000:ab
furry@Wintermute:~>
```

Issue #1: If negative response for “AAAA” validates and (Do = 1 & CD = 0) DNS64 MAY perform synthesis

```plaintext
furry@Wintermute:~> dig +dnssec +nacdflag ipv4only.arpa aaaa +short
furry@Wintermute:~>
```

Issue #2: SECURITY?
NAT64 prefix 2001:db8::/96

1. Discovery of the IPv6 Prefix Used for IPv6 Address Synthesis
2. 2001:db8::c000:ab PTR nat64.example.net.
3. Nat64.example.net. AAAA 2001:db8::c000:ab
4. Nat64.example.net. AAAA RRSIG ...

CD = 0

ipv4only.arpa AAAA ?

ipv4only.arpa AAAA 2001:db8::c000:ab

2001:db8::c000:ab PTR ?

DNS64

Validating resolver

b.a.0.0.0.0.c.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

PTR NAT64.example.net

nat64.example.net AAAA ?

Nat64.example.net AAAA 2001:db8::c000:ab

Nat64.example.net RRSIG AAAA.....
Conclusions

- Non-DNS64 aware validating Client behind NAT64:
  - Failure rate ~1.3% ... or 94%....

- Service owners:
  - enable IPv6 (especially if DNSSEC is enabled!)

- DNSSEC-aware and validating stub resolvers SHOULD be DNS64-aware
  - Discover NAT64 prefix
  - Perform DNS64 function
QUESTIONS?
Backup Slides
IPv6-enabled Sites Distribution: Alexa 1M
IPv6-enabled Sites Distribution: Alexa 10K
DNSSEC-enabled Names Distribution: Alexa 1M
DNSSEC-Enabled Sites Distribution: Alexa 10K
DNSSEC-Enabled IPv4-only Names (Alexa 1M)