

NAT64/DNS64 detection via SRV Records

draft-hunek-v6ops-nat64-srv-01

Martin Huněk

Technical University of Liberec

2022-03-21

Why?

Current solutions

- ▶ RFC 7050: DNS Well-Known Name (updated by RFC 8880)
- ▶ RFC 7225: Port Control Protocol
- ▶ RFC 8115: DHCPv6 Option
- ▶ RFC 8781: RA Option

Current solutions

- ▶ RFC 7050: DNS Well-Known Name (updated by RFC 8880)
 - ▶ Hard to implement correctly but implemented somehow
 - ▶ Does not work with third-party DNS providers
- ▶ RFC 7225: Port Control Protocol
 - ▶ Ignored by ISPs
- ▶ RFC 8115: DHCPv6 Option
 - ▶ Ignored by Android
- ▶ RFC 8781: RA Option
 - ▶ Ignored by routers (so far)

Later 3 are not usable in user-space. Applications usually do not speak PCP, DHCPv6 or RA and mandatory access control issues.

RFC 7050 + RFC 8880

In order to be secure requires:

- ▶ DNSSEC signed NAT64 FQDN
- ▶ Corresponding PTR
- ▶ Secure Channel between Node and resolver*
- ▶ Trusted domain list*
- ▶ No user input*
- ▶ Stub resolver must distinguish between configuration sources of rDNS*
- ▶ Only autoconfiguration sources allowed to resolve WKN
- ▶ Recursive DNS resolver is an interface-specific*

* Are problematic

How?

Goals

- Goal 1 No new protocol or alteration of an existing one.
- Goal 2 Utilize widely supported protocols.
- Goal 3 Utilize information already provided by a network.
- Goal 4 Must work with foreign DNS.
- Goal 5 Must not require DNS64 synthesis on a host.
- Goal 6 Must not require prior provisioning (BYOD).
- Goal 7 Must provide secure detection over an insecure channel.
- Goal 8 Must be able to run in user-space.

Result?

- ▶ Every application should be able to talk to DNS
- ▶ A node knows its IP address - can have PTR
- ▶ The information must be in the global DNS tree
- ▶ DNSSEC provides data authenticity (host knows the root)
- ▶ SRV record is good for that (structured, priorities, weights, and TTL)
- ▶ It can be up to host-specific level

SRV record

Format of an SRV record

`<service>`.`<proto>`._`<domain>`. `<TTL>` IN SRV `<priority>` `<weight>` `<port>`
`<target>`

Proposed SRV records

_nat64._ipv6.example.com. 84000 IN SRV 5 0 9632 nat64prefix
_dns64._udp.example.com. 84000 IN SRV 5 0 53 dns64
_dns64._tcp.example.com. 84000 IN SRV 5 0 53 dns64

News?

Changelog

v01:

- ▶ Detailed process of local domain detection

v00:

- ▶ PTR records instead of DNSSSL
- ▶ Reasons for having another method
- ▶ Interactions with other methods and 464XLAT
- ▶ Any transport method could be used
- ▶ Negative answer
- ▶ TTL behaviour
- ▶ Multicast support
- ▶ Proof of concept code
- ▶ Spelling and grammar

Thank you for your attention.



Figure: Github repository <https://github.com/hunator/draft-v6ops-nat64-srv>