DNSSEC in the networks with a NAT64/DNS64

Martin Hunek and Zdenek Pliva
Institute of Information Technology and Electronics
Technical University of Liberec
Liberec, Czech Republic
Email: martin.hunek@tul.cz

***Abstract*—This paper describes the problems with using both Domain Name System Security (DNSSEC) (security extension to domain name system) validating Domain Name System (DNS) resolvers and NAT64/DNS64 transition mechanism. In this paper we also propose a solution how to solve the problem of such combination. The foreign (synthesized) AAAA record as well as the broken trust chain in such records in secure way which doesn't breach DNSSEC.**

A current widely used solution comes from RFC 7050 [1] with conjunction with RFC 6146 [2] and RFC 6147 [3]. In such case the end node will detect Domain Name System 6-to-4 (DNS64) by asking for well-known Internet Protocol version 4 (IPv4) only domain, if detected end node would disable DNSSEC validation. This solves previously mentioned problem of foreign AAAA record and such domain would be reachable. However this also brakes DNSSEC validation and it does not allow operator to control over the prefix preference.

Our proposed solution supplies the end node with secondary DNSSEC chain to validate DNS64 synthesized records from information already presented to the node by Neighbor Discovery or Dynamic Host Configuration Protocol version 6 (DHCPv6), in the way that network operator can have a control over the prefixes and DNS resolvers used by the end node for NAT64/DNS64 transition mechanism.

***Index Terms*—IPv6, NAT64, DNS64, DNSSEC.**

I. Introduction

This paper deals with conjunction of two technologies. One is the security extension to domain name system – DNSSEC [4], the second is a transition mechanism between internet protocol version 4 and version 6 – the Network Address Translation 6-to-4 (NAT64) [2] and its integral part DNS64 [3].

The main problem of such conjunction is the DNS64 part of the transition mechanism. Due to its nature the DNS64 synthesize Internet Protocol version 6 (IPv6) – AAAA record for domain name which has got only IPv4 – A record, is effectively pointing the communication towards the network address translation node – the NAT64.

On the other hand the DNSSEC is preventing undetected manipulation to the zone which may get manipulated by synthesized AAAA records produced by DNS64. In other words these technologies are effectively working against each other. Usual way to handle this situation is to disable one of them, either loosing ability of communication between IPv4 and IPv6 nodes by disabling DNS64 or by loosing security aspects of a DNS by disabling DNSSEC validation.

II. Theoretical background

When the internet protocol version 6 has been designed, It has been decided that instead of just expanding IP address space by extending the IP header, entirely new protocol should be designed. This let to inability of IPv4 only node in communication directly with IPv6 node and vice versa. Due to this limitation, the tunneling and translation mechanisms has been invented.

One of the translation mechanisms is the NAT64/DNS64, which consists of two components. The first component is the NAT64, which stands for Network Address Translation IPv6 to IPv4. It basically does the same thing as the Network Address Translation 4-to-4 (NAT44) or the Network Address Translation (NAT) [5] in short. It extracts the IP header and replaces it by new one. In this case the transformation is between two different protocols.

The second part – DNS64 is responsible for pointing the end nodes to use NAT64 gateway. If the target does have only IPv4 (A) record in DNS, the DNS64 resolver synthesize an IPv6 (AAAA) record which point to network prefix used by NAT64. This effectively point end node to NAT64 and whole communication in the infrastructure of operator network would go through the IPv6 protocol (due to its priority over older IPv4). After the transition on the NAT64 L4, data would be transported over IPv4 to target IPv4 node. Vice versa, the data from target to end node would be transported over IPv4 to the NAT64 box to its IPv4 address and then the response would be translated back to IPv6 and send to the originating end node.

Because the NAT64/DNS64 is based on the modification of DNS responses - effectively working on the same schema as the Man in the Middle (MitM) attack, it opens some security vulnerabilities. These include Denial of Service (DoS), end node flooding and MitM attacks. To overcome this problem the DNSSEC must be used and for DNSSEC usage, the node must know the trusted domain list. Standard does not specify the correct way how the trusted domain list should be determined, however it might use some of these sources:

ISBN 978-80-261-0722-4, © University of West Bohemia, 2018

- End user maintained list.
- ISP maintained list.
- Autoconfiguration via Stateless Address Autoconfiguration (SLAAC) - Domain Name System Search List (DNSSL) option
- Autoconfiguration via DHCPv6 - option 24
- Autoconfiguration via DHCPv4

III. Current solution

Current solution outlined by the RFC 7050 [1], use well-known domain "*ipv4only.arpa.*" which has got only two A records *192.0.0.170* and *192.0.0.171*. However, when the end node asks the DNS64 enabled resolver, the response would be a IPv6 AAAA record pointing to NAT64 pool ending by hexadecimal representation of above mentioned addresses (*C000:AA* or *C000:AB*). By this way the end node knows, that network uses DNS64 and should use NAT64.

This also should trigger either DNSSEC enabled end node stub resolver or the DNSSEC enabled caching resolver to keep the "Checking Disabled" flag set to zero. This action informs the DNS64 resolver to synthesize AAAA record which otherwise would be enabled. So in such case the IPv6 only nodes would not be able to access the IPv4 only nodes - they would not receive the AAAA record pointing to the NAT64 box.

To overcame possible security vulnerabilities, introduced by this "legal" modification of DNS records. The RFC 7050 [1] came up with DNSSEC validation of provided NAT64 prefix. However method proposed by RFC 7050 [1] is quite complex in the sense of number of needed steps and phases and it also has got a lower manageability.

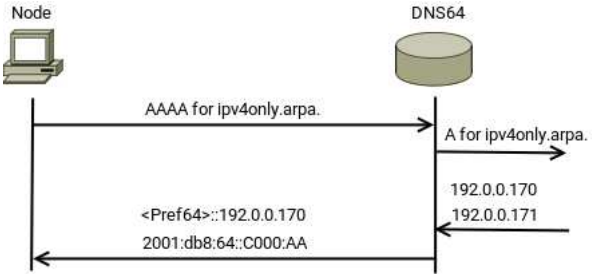

Figure 1. Detection of NAT64 prefix according to RFC 7050 [1]

First stage of NAT64/DNS64 discovery is the detection of NAT64 prefix. This is shown in the figure 1. In this figure it can be seen, that method proposed by current RFC does not require the node to have any specific knowledge about its network. This is the bright side of current approach, however the the well-known address has to be served in the arpa domain. The rest of this part of this process is quite straight forward. The DNS64 box translates the well-known address to the NAT64 prefix according to RFC 6147 [3]. The address received from the arpa domain had to match with the standard, otherwise record received by the node would have been ignored. By this step, the detection of NAT64 prefixes ends. The non-validating node can start to use received prefix for accessing IPv4 only nodes, however the DNSSEC would not be available and the end node could be subjected to the race condition type of DoS attack, MitM attack or can participate on flooding attack. To leverage DNSSEC for protection against such attack the end node must verify all of the received NAT64 prefixes.

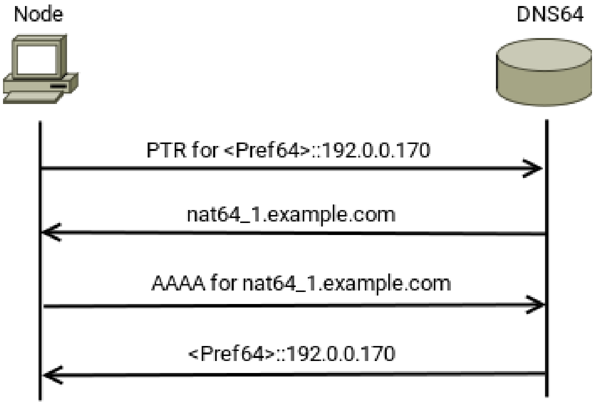

Figure 2. Validation of NAT64 prefix according to RFC 7050 [1]

The DNSSEC validating node continues in the process according to the figure 2. In the first step the node asks for the reverse record (PTR) for every detected prefix – well-known encoded address outside of well-known NAT64 prefix (that can't be validated by DNSSEC and it is supposed to be safe). When the node receives the PTR reply, it had to compare the received domain name with the list of trusted domains. This require the end node knowledge about its network prior to successful validation. The RFC 7050 [1] does not explicitly describe the way for the node how to acquire such a list but it is supposed to be either set by user/operator or by autoconfiguration (SLAAC or DHCPv6).

If the domain in PTR record matches a domain from the trusted list, node have to ask for an AAAA record of every matching PTR. After that the node must validate every response and the address in an AAAA response must match the previously discovered ones. If everything checks out fine, the discovery has been successfully completed and validated prefix is marked as trusted.

IV. Proposed solution

In the contrast with the RFC 7050 [1] solution of this problem, we propose to reverse its logic for faster and simpler process of NAT64/DNS64 discovery. Supposing that the node has got the trusted domain list and that it would be able to get an "active" domain list by autoconfiguration (e.g. SLAAC – DNSSL or DHCPv6 option 24). Then the node can match those lists and start asking for proposed Service (SRV) records.

The node would have to ask first for SRV for *_nat64._ipv6* in trusted and active domains as it is shown in the figure 3. As a response, the node would receive a list of all prefixes with their priorities and weights. This is one of the major differences between proposed solution and the RFC 7050 [1], which does not provide a way for network operator to specify

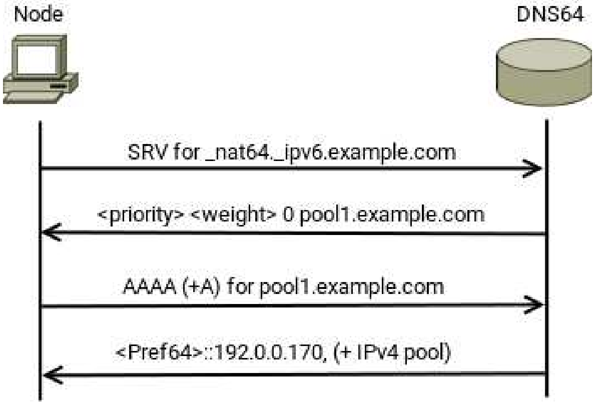
Node
DNS64
SRV for _nat64._ipv6.example.com
<priority> <weight> 0 pool1.example.com
AAAA (+A) for pool1.example.com
<Pref64>::192.0.0.170, (+ IPv4 pool)

Figure 3. Proposed NAT64 SRV record and NAT64 discovery

NAT64 pool priorities. The number reserved for port number can then be optionally used for indicating pool sizes both for IPv6 and IPv4 or set to zero. When it is non-zero it must indicate the length of network masks for both protocols, IPv4 appended decadically after IPv6 (for example 09632 – meaning NAT64 IPv6 prefix has length 96 bits and it is translated to single IPv4 address). Then the node might additionally ask for A record of such pool, determining its public IPv4 address (or size of dynamic pool), if needed by application. Otherwise only AAAA record would be needed to determine NAT64 IPv6 pool.

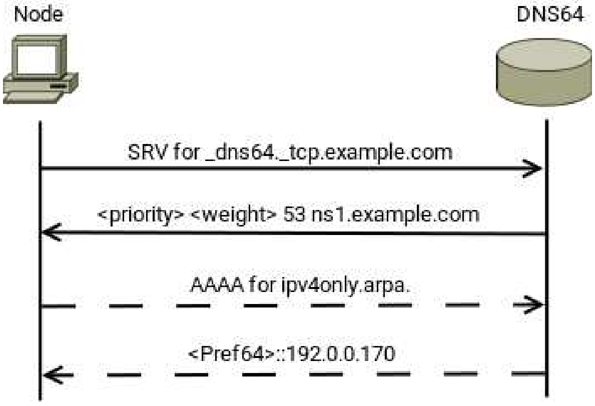
Node
DNS64
SRV for _dns64._tcp.example.com
<priority> <weight> 53 ns1.example.com
AAAA for ipv4only.arpa.
<Pref64>::192.0.0.170

Figure 4. Proposed DNS64 SRV record and DNS64 discovery

In addition to the NAT64 SRV records, we also propose the DNS64 SRV record. This adds the possibility for network operator to run DNS64 service outside of the primary DNS infrastructure. This way the network operator might choose to provide DNS64 service only to this new standard capable nodes. By this way operator may effectively solve possible problems with old DNSSEC implementations. The process of the DNS64 server detection is shown in the figure 4.

The above mentioned figure also shows optional validation of DNS64 box function. However subsequent query and DNSSEC validation of PTR records is not necessary due to the signature of SRV record. If the DNS64 SRV record is not present the node should fall back to process outlined by the RFC 7050 [1].

Of course the whole proposed solution requires the same prerequisites as the RFC 7050 [1] does. The domain used for NAT64 prefix discovery must be DNSSEC secured and the DNSSEC validating node must ensure that all responses are valid. The PTR records should still match corresponding AAAA records, however it is not required by proposed method so there is also no requirement concerning DNSSEC deployment in reverse zone. Due to the absence of PTR record queries, there is no difference between processing network specific NAT64 prefixes and well-known NAT64 prefix. All of them are validated by signatures of SRV and AAAA records in the trusted domain. Secure transmission of trusted domains and security of routing NAT64 prefixes remains within responsibility of network operator and it is out of scope of proposed NAT64 prefix discovery method.

V. Conclusion

Our proposed method of NAT64 prefix discovery extends the current standard in use (the RFC 7050 [1]) by adding alternative means of secure prefix discovery. It utilizes the well-known IPv4 only record in ARPA domain as well as the well-known IP address and provides compatibility with above mentioned standard as a fallback option. Node, unaware of this method, would not be impacted by the proposed method. Network not utilizing the new method would make penalty to method aware nodes in total length of processing one SRV query and corresponding NODATA and NSEC(/NSEC3) response.

When implemented, our proposed method should be used before the method outlined in the RFC 7050 [1]. The first query should be for NAT64 SRV record, then the node may ask for DNS64 SRV record or continue with AAAA query for *ipv4only.arpa* for current resolver and fallback to SRV record method only if its current resolver does not provide DNS64 service.

Main contribution of proposed method lays in the added possibility of network operator to provide sorted list of NAT64 prefixes by their priority. This allows network operator controlled load balancing, which is not possible with current standard. The same applies to DNS64 service record, which also provides a possibility to run DNS64 service outside of main DNS infrastructure. This might help to overcome possibly broken implementations of current standard in DNSSEC validating nodes.

Proposed method doesn't suggest changes to DNS itself nor adds any new record type. As such, it does not introduce any new vectors of attack or performance impact in usual DNS operations. Only suggested changes are made in the NAT64/DNS64 detection algorithm. From security perspective, it even closes one security bug, the signature of well-known prefix, however this vector of attack is purely academical.

Acknowledgment

This paper has been supported by Student Grant Scheme (SGS 2018) at Technical University of Liberec.

Acronyms

DHCPv6 Dynamic Host Configuration Protocol version 6. 1, 2

DNS Domain Name System. 1–3

DNS64 Domain Name System 6-to-4. 1–3

DNSSEC Domain Name System Security. 1–3

DNSSL Domain Name System Search List. 2

IPv4 Internet Protocol version 4. 1–3

IPv6 Internet Protocol version 6. 1–3

MitM Man in the Middle. 1, 2

NAT Network Address Translation. 1

NAT44 Network Address Translation 4-to-4. 1

NAT64 Network Address Translation 6-to-4. 1–3

SLAAC Stateless Address Autoconfiguration. 2

SRV Service. 2, 3

References

[1] T. Savolainen, J. Korhonen, and D. Wing, “Discovery of the IPv6 Prefix Used for IPv6 Address Synthesis”, RFC Editor, RFC 7050, Nov. 2013, pp. 1–22. DOI: 10.17487/RFC7050. [Online]. Available: https://www.rfc-editor.org/rfc/rfc7050.txt.

[2] M. Bagnulo, P. Matthews, and I. van Beijnum, “Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers”, RFC Editor, RFC 6146, Apr. 2011, pp. 1–45. DOI: 10.17487/RFC6146. [Online]. Available: https://www.rfc-editor.org/rfc/rfc6146.txt.

[3] M. Bagnulo, A. Sullivan, P. Matthews, and I. van Beijnum, “DNS64: DNS Extensions for Network Address Translation from IPv6 Clients to IPv4 Servers”, RFC Editor, RFC 6147, Apr. 2011, pp. 1–32. DOI: 10.17487/RFC6147. [Online]. Available: https://www.rfc-editor.org/rfc/rfc6147.txt.

[4] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose, “DNS Security Introduction and Requirements”, RFC Editor, RFC 4033, Mar. 2005, pp. 1–21. DOI: 10.17487/RFC4033. [Online]. Available: https://www.rfc-editor.org/rfc/rfc4033.txt.

[5] P. Srisuresh and K. B. Egevang, “Traditional IP Network Address Translator (Traditional NAT)”, RFC Editor, RFC 3022, Jan. 2001, pp. 1–16. DOI: 10.17487/RFC3022. [Online]. Available: https://www.rfc-editor.org/rfc/rfc3022.txt.