v6ops
Internet-Draft
Intended status: Best Current Practice
Expires: April 11, 2018

J. Palet Martinez
Consulintel, S.L.
October 8, 2017

464XLAT Deployment Guidelines in Operator Networks
draft-palet-v6ops-464xlat-deployment-00

Abstract

This document describes how 464XLAT ([RFC6877]) can be deployed in an IPv6 operator network and the issues to be considered.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 11, 2018.

Table of Contents

1. Introduction

464XLAT ([RFC6877]) describes an architecture that provides IPv4 connectivity across a network, or part of it, when it is only natively transporting IPv6.

In order to do that, 464XLAT ([RFC6877]) relies on the combination of existing protocols:

1. The customer-side translator (CLAT) is a stateless IPv4 to IPv6 translator (NAT46) ([RFC7915]) implemented in the end-user device or CE, located at the "customer" edge of the network.

2. The provider-side translator (PLAT) is a stateful NAT64 ([RFC6146]), implemented typically at the opposite edge of the operator network, that provides access to both IPv4 and IPv6 upstreams.

3. Optionally, DNS64 ([RFC6147]), implemented as part of the PLAT allows an optimization (a single translation at the NAT64, instead of two translations - NAT46+NAT64), when the application at the end-user device supports IPv6 DNS (uses AAAA RR).

464XLAT ([RFC6877]) is a very simple approach to cope with the major NAT64+DNS64 drawback: Not working with applications or devices that use literal IPv4 addresses or non-IPv6 compliant APIs.

464XLAT ([RFC6877]) has been used initially in IPv6 cellular networks, so providing an IPv6-only access network, the end-user

device applications can access IPv4-only end-networks/applications, despite those applications or devices use literal IPv4 addresses or non-IPv6 compliant APIs.

In addition to that, in the same example of the cellular network above, if the User Equipment (UE) provides tethering, other devices behind it will be presented with a traditional NAT44, in addition to the native IPv6 support.

Furthermore, 464XLAT ([RFC6877]) can be used in non-cellular IPv6 wired (xDSL, DOCSIS, FTTH, Ethernet, ...) and wireless (WiFi) network architectures, by implementing the CLAT functionality at the CE.

The remaining sections of this document, despite of any specific examples being used, are applicable to any operator network architecture, and introduces possible issues and general deployment guidelines to be considered when deploying 464XLAT ([RFC6877]) in an IPv6 network.

2. DNSSEC Considerations

As indicated in Section 8 of [RFC6147] (DNS64, Security Considerations), because DNS64 modifies DNS answers and DNSSEC is designed to detect such modifications, DNS64 can break DNSSEC.

If a device connected to an IPv6-only WAN queries for a domain name in a signed zone, by means of a recursive name server that supports DNS64, and the result is a synthesized AAAA record, and the recursive name server is configured to perform DNSSEC validation and has a valid chain of trust to the zone in question, it will cryptographically validate the negative response from the authoritative name server. So, the recursive name server actually lie to the client device, however in most of the cases, the client will not notice it, because generally they don’t perform validation themselves as instead rely on their recursive name servers.

If the client device performs DNSSEC validation on the AAAA record, it will fail as it is a synthesized record.

Similarly, if the client querying the recursive name server is another name server configured to use it as a forwarder, and is performing DNSSEC validation, it will also fail on any synthesized AAAA record.

There are several possible solutions to avoid breaking DNSSEC:

2.1. DNSSEC validator aware of DNS64

In general, DNS servers with DNS64 function, by default, will not synthesize AAAA responses if the DNSSEC OK (DO) flag was set in the query. In this case, as only an A record is available, it means that the CLAT will take the responsibility, as in the case of literal IPv4 addresses, to keep that traffic flow end-to-end as IPv4, so DNSSEC is not broken.

2.2. Stub validator

If the DO flag is set and the client device performs DNSSEC validation, and the Checking Disabled (CD) flag is set for a query, as the DNS64 recursive server will not synthesize AAAA responses, the client could perform the DNSSEC validation with the A record and then may query the network for a NAT64 prefix ([RFC7050]) in order to synthesize the AAAA ([RFC6052]). This allows the client device to avoid using the CLAT and still use NAT64 even with DNSSEC.

Some devices/OSs may implement, instead of CLAT, a simliar function by using Bump-in-the-Host ([RFC6535]). In this case, the considerations in the above paragraphs are also applicable.

2.3. CLAT with DNS proxy and validator

If a CE includes CLAT support and also a DNS proxy, as indicated in Section 6.4 of [RFC6877], the CE could behave as a stub validator on behalf of the client devices, following the same approach described in the precedent section (Stub validator). So the DNS proxy actually lie to the client devices, which in most of the cases will not notice it unless they perform validation themselves. Again, this allow the clients devices to avoid using the CLAT and still use NAT64 with DNSSEC.

2.4. ACL of clients

In cases of dual-stack clients, stub resolvers should send the AAAA queries before the A ones. So such clients, if DNS64 is enabled, will never get A records, even for IPv4-only servers, and they may be in the path before the NAT64 and accesible by IPv4. If DNSSEC is being used for all those flows, specific addresses or prefixes can be left-out the DNS64 synthesis by means of ACLs.

2.5. Mapping-out IPv4 addresses

If there are well-known specific IPv4 addresses or prefixes using DNSSEC, they can be mapped-out of the DNS64 synthesis.

Even if this is not related to DNSSEC, this "mapping-out" feature is actually quite commonly used to ensure that [RFC1918] addresses (for example used by LAN servers) are not synthesized to AAAA.

3. Using 464XLAT with/without DNS64

In the case the client device is IPv6-only (either because the stack is IPv6-only, or because it is connected via an IPv6-only LAN) and the server is IPv4-only (either because the stack is IPv4-only, or because it is connected via an IPv4-only LAN), only NAT64 combined with DNS64 will be able to provide access among both. Because DNS64 is then required, DNSSEC validation will be only possible if the recursive name server is validating the negative response from the authoritative name server and the client is not performing validation.

However, when the client device is dual-stack and/or connected in a dual-stack LAN by means of a CLAT (or has the built-in CLAT), DNS64 is an option.

1. With DNS64: If DNS64 is used, most of the IPv4 traffic (except if using literal IPv4 addresses or non-IPv6 compliant APIs) will not use the CLAT, so will use the IPv6 path and only one translation will be done at the NAT64. This may break DNSSEC, unless measures as described in the precedent section are taken.

2. Without DNS64: If DNS64 is not used, all the IPv4 traffic will make use of the CLAT, so two translations are required (NAT46 at the CLAT and NAT64 at the PLAT), which adds some overhead in terms of the extra NAT46 translation, however avoids the AAAA synthesis and consequently will never break DNSSEC.

When clients in an operator network use DNS from other networks, for example manually configured by users, they may support or not DNS64, so the considerations in this section will apply as well.

4. DNS64 and Reverse Mapping Considerations

When a client device, using a name server configured to perform DNS64, tries to reverse-map a synthesized IPv6 address, the name server responds with a CNAME record pointing the domain name used to reverse-map the synthesized IPv6 address (the one under ip6.arpa), to the domain name corresponding to the embedded IPv4 address (under in-addr.arpa).

This is the expected behaviour, so no issues to be considered regarding DNS reverse mapping.

5. CLAT Translation Considerations

As described in Section 6.3 of [RFC6877] (IPv6 Prefix Handling), if the CLAT can be configured with a dedicated /64 prefix for the NAT64 translation, then it will be possible to do a more efficient stateless translation.

However, if this dedicated prefix is not available, the CLAT will need to do a stateful translation, for example performing stateful NAT44 for all the IPv4 LAN packets, so they appear as coming from a single IPv4 address, and then in turn, stateless translated to a single IPv6 address.

The obvious recommended setup, in order to maximize the CLAT performance, is to configure the dedicated translation prefix. This can be easily achieved automatically, if the CE or end-user device is able to obtain a shorter prefix by means of DHCPv6-PD ([RFC3633]), so the CE can use a /64 for that.

The above recommendation is often not posible for cellular networks, when connecting UEs (some broadband cellular use DHCPv6-PD ([RFC3633]), but smartphones, in general, not), as they provide a single /64 for each PDP context and use /64 prefix sharing ([RFC6877]). So in this case, the UEs typically have a build-in CLAT client, which is doing a stateful NAT44 before the stateless NAT46.

6. Summary of deployment recommendations for 464XLAT

As indicated in the introduction of this document, operators willing to deploy 464XLAT ([RFC6877]), MUST to support, at least, the provider-side translator (PLAT).

In the case it is a non-cellular network and the operator is providing the CEs to the customers, or suggesting them some specific models, they MUST support the customer-side translator (CLAT).

If the operator offers DNS services, in order to increase performance by reducing the double translation for all the IPv4 traffic, and avoid breaking DNSSEC, they MAY support DNS64. In this case, if the DNS service is offering DNSSEC validation, then it MUST be in such way that it is aware of the DNS64. This is considered de simpler and safer approach, and MAY be combined as well with the other possible solutions described in this document:

- Devices running CLAT SHOULD follow the indications in the "Stub validator" section recommendation. However, most of the time, this is out of the control of the operator.

- o CEs SHOULD include a DNS proxy and validador. This is relevant if the operator is providing the CE or suggesting it to customers.

- o ACL of clients and Mapping-out IPv4 addresses MAY be considered by each operator, depending on their own infrastructure.

The ideal configuration for CEs supporting CLAT, is that they support DHCPv6-PD ([RFC3633]) and internally reserve one /64 for the stateless NAT46 translation. The operator MUST ensure that the customers get allocated prefixes shorter than /64 in order to support this optimization. One way or the other, this is not impacting the performance of the operator network.

As indicated in Section 7 of [RFC6877] (Deployment Considerations), operators MAY follow those suggestions in order to take advantage of traffic engineering.

In the case of cellular networks, the considerations regarding DNSSEC may appear as out-of-scope, because UEs OSs, commonly don’t support DNSSEC, however applications running on them may do, or it may be an OS "built-in" support in the future. Moreover, if those devices offer tethering, other client devices may be doing the validation, hence the relevance of a proper DNSSEC support by the operator network.

Furthermore, cellular networks supporting 464XLAT ([RFC6877]) and "Discovery of the IPv6 Prefix Used for IPv6 Address Synthesis" ([RFC7050]), allow a progressive IPv6 deployment, with a single APN supporting all types of PDP context (IPv4, IPv6, IPv4v6), in such way that the network is able to automatically serve all the possible combinations of UEs.

Finally, if the operator choose to secure the NAT64 prefix, it MUST follow the advise indicated in Section 3.1.1. of [RFC7050] (Validation of Discovered Pref64::/n).

7. Security Considerations

This document does not have any new specific security considerations.

8. IANA Considerations

This document does not have any new specific IANA considerations.

9. Acknowledgements

The author would like to acknowledge the inputs of TBD ... Christian Huitema inspired working in this document by suggesting that DNS64 should never be used, during a discussion regarding the deployment of CLAT in the IETF network.

10. Normative References

[RFC1918] Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot, G., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, DOI 10.17487/RFC1918, February 1996, <https://www.rfc-editor.org/info/rfc1918>.

[RFC3633] Troan, O. and R. Droms, "IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6", RFC 3633, DOI 10.17487/RFC3633, December 2003, <https://www.rfc-editor.org/info/rfc3633>.

[RFC6052] Bao, C., Huitema, C., Bagnulo, M., Boucadair, M., and X. Li, "IPv6 Addressing of IPv4/IPv6 Translators", RFC 6052, DOI 10.17487/RFC6052, October 2010, <https://www.rfc-editor.org/info/rfc6052>.

[RFC6146] Bagnulo, M., Matthews, P., and I. van Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers", RFC 6146, DOI 10.17487/RFC6146, April 2011, <https://www.rfc-editor.org/info/rfc6146>.

[RFC6147] Bagnulo, M., Sullivan, A., Matthews, P., and I. van Beijnum, "DNS64: DNS Extensions for Network Address Translation from IPv6 Clients to IPv4 Servers", RFC 6147, DOI 10.17487/RFC6147, April 2011, <https://www.rfc-editor.org/info/rfc6147>.

[RFC6535] Huang, B., Deng, H., and T. Savolainen, "Dual-Stack Hosts Using "Bump-in-the-Host" (BIH)", RFC 6535, DOI 10.17487/RFC6535, February 2012, <https://www.rfc-editor.org/info/rfc6535>.

[RFC6877] Mawatari, M., Kawashima, M., and C. Byrne, "464XLAT: Combination of Stateful and Stateless Translation", RFC 6877, DOI 10.17487/RFC6877, April 2013, <https://www.rfc-editor.org/info/rfc6877>.

[RFC7050] Savolainen, T., Korhonen, J., and D. Wing, "Discovery of the IPv6 Prefix Used for IPv6 Address Synthesis", RFC 7050, DOI 10.17487/RFC7050, November 2013, <https://www.rfc-editor.org/info/rfc7050>.

[RFC7278] Byrne, C., Drown, D., and A. Vizdal, "Extending an IPv6 /64 Prefix from a Third Generation Partnership Project (3GPP) Mobile Interface to a LAN Link", RFC 7278, DOI 10.17487/RFC7278, June 2014, <https://www.rfc-editor.org/info/rfc7278>.

[RFC7915] Bao, C., Li, X., Baker, F., Anderson, T., and F. Gont, "IP/ICMP Translation Algorithm", RFC 7915, DOI 10.17487/RFC7915, June 2016, <https://www.rfc-editor.org/info/rfc7915>.

Author’s Address

Jordi Palet Martinez
Consulintel, S.L.
Molino de la Navata, 75
La Navata - Galapagar, Madrid 28420
Spain

Email: jordi.palet@consulintel.es
URI: http://www.consulintel.es/