NAT64/DNS64 real life experiments, warnings and also one useful tool
...from go6lab.si and IPv6-lab.net
Problem statement and real world status

*(Six Degrees of Inner Turbulence)*

IPv6 and IPv4 are incompatible on the wire.

- We need transition and translation mechanisms between the two protocols
- Mobile operators are massively switching devices to IPv6-only connectivity
  - millions of users
  - using 464XLAT (Android) or plain NAT64 (Apple) as a transition tool to access IPv4 content
- Some people do weird stuff while adding AAAA to their DNS records
- Important questions:
  - Do content providers know how their content will be seen from such environments?
  - Do connectivity providers know what their users’ experience on IPv6-only would be?
Credits, acknowledgments and supporters

(Metropolis, Pt. 1: The Miracle and the Sleeper)

Projects like this and all the massive amount of work cannot be performed without great support of people and companies that we work with:

**Internet Society** - for Jan’s working time and funding of the numerous travels where we got the idea of the IPv6 issues while talking to operators and also future travels where we’ll talk about this topic (we are here, aren’t we? :)

**Go6 Institute Slovenia** - for funding and running the Go6lab where we got connectivity, HW, SW and place to experiment with all this things IPv6

**SJM Steffann** for Sander’s working time, coding skills, massive energy and also HW/SW in their ipv6-lab.net to make this tool redundant.

**Corinne Pritchard** for a lovely design of a tool’s front-end. Hugs :)
Go6lab NAT64/DNS64 public test-bed

(The Test That Stumped Them All)

Aimed at everyone who would like to test NAT64/DNS64 functionality
4 different implementations, 4 different instructions how to direct traffic there

Used by operators
  Testing the idea of providing NAT64/DNS64 and/or 464XLAT to their users

Used by application providers
  To see how their apps works in NAT64/DNS64 environment

Used by HW/SW vendors
  Testing their solutions against multiple NAT64 vendors

Gained quite some traction and momentum this days

Instructions: https://go6lab.si/current-ipv6-tests/nat64dns64-public-test/
Go6lab NAT64/DNS64 test-bed

NAT64/DNS64 public test

Go6lab is hosting a variety of NAT64/DNS64 solutions, open for general Internet public for testing.

Disclaimer: This setup is not intended for performance testing, just to see how NAT64/DNS64 operates, what applications breaks and what are differences between different implementations. If you need to do performance testing of NAT64/DNS64 send email to <zavod@go6.si> and schedule a test session in go6lab facility.

To test different NAT64/DNS64 setups you need to disable IPv4 on your device and set an IPv6 resolving DNS server, different one for each setup. Please send us your observations, specially about which applications breaks in IPv6-only/NAT64 environment. We would like to build a list of them.
A10 Networks NAT64 implementation: set your DNS to 2001:67c:27e4:15::6411
NAT64 implementation is running on a A10 vThunder virtual appliance.
NAT64 routed prefix: 2001:67c:27e4:642::/64
Quick ping6 test if up&running: ping6 2001:67c:27e4:642::5bef:6015

PaloAlto Networks Firewall NAT64 with BIND9 DNS64: set your DNS to 2001:67c:27e4::64
NAT64 implementation is running in PAN500 firewall box.
NAT64 routed prefix: 2001:67c:27e4:64::/64
Quick ping6 test if up&running: ping6 2001:67c:27e4:64::5bef:6015

Jool NAT64 with BIND9 DNS64: set your DNS to 2001:67c:27e4:15::64
Jool NAT64 implementation is running in a virtual container on proxmox server.
NAT64 routed prefix: 2001:67c:27e4:1064::/64
Quick ping6 test if up&running: ping6 2001:67c:27e4:1064::5bef:6015

Cisco ASR1000 NAT64 with BIND9 DNS64: set your DNS to 2001:67c:27e4::60
NAT64 implementation is running in Cisco ASR1001.
NAT64 routed prefix: 2001:67c:27e4:11::/64
Quick ping6 test if up&running: ping6 2001:67c:27e4:11::5bef:6015
Go6lab NAT64/DNS64 test-bed HW/SW
(Systematic Chaos)
Some DNS admins put “crap” in AAAA records
(The Enemy Inside)

Have you ever seen any of this values for AAAA record?

::
::1
::ffff:[IPv4_addr]
fe80::[some_value]
64:ff9b::[some_value]
2001:DB8::[some_value]

If you have seen something like this and you know who did it, talk to that people and tell them to fix it. This sort of thing isn’t useful to anybody and severely impacts user experience.

More: http://www.employees.org/~dwing/aaaa-stats/ (courtesy of Dan Wing)
Causing confusion with or without www
(Breaking All Illusions)

sander@MacPro:~$ host www.firstinsight.com
www.firstinsight.com is an alias for 160569.group19.sites.hubspot.net.
160569.group19.sites.hubspot.net is an alias for
cos2mdc.hubspot.net.mdc.edgesuite.net.
cos2mdc.hubspot.net.mdc.edgesuite.net is an alias for a1711.b.akamai.net.
a1711.b.akamai.net has address 88.221.254.18
a1711.b.akamai.net has address 88.221.254.10

sander@MacPro:~$ host firstinsight.com
firstinsight.com has IPv6 address ::
firstinsight.com mail is handled by 10 mx1.emailsrvr.com.
firstinsight.com mail is handled by 20 mx2.emailsrvr.com.
When deploying in real life you need “fixes”

(Build Me Up, Break Me Down)

So, what can we do about it?

We can figure out who this people are, contact them, warn them about the issue, educate them and ask them to fix the problem.

Remember: If you are not part of solution, you are part of the problem.

At the same time we can protect our users from bad user experience and set the “exclude” rules in our DNS64 servers. IANA allocated 2000::/3 as global unicast address pool, so whatever else is used in AAAA - it’s by default bogus and we can safely ignore that.
When deploying in real life you need “fixes”

(Stream of Consciousness)

**BIND9** example of DNS64 configuration in go6lab:

```
dns64 2001:67c:27e4:64::/96 {
    clients { any; };
    mapped { !rfc1918; any; };
    exclude { 0::/3; 4000::/2; 8000::/1; 2001:DB8::/32; };
    break-dnssec yes;
};
```

Explanation of “exclude” configuration parameter: If DNS64 server gets an AAAA record with a value of anything outside 2000::/3 - it ignores it and synthesizes the AAAA record from NAT64_prefix::IPv4_address
When deploying in real life you need “fixes”

(Moment Of Betrayal)

This fixes the most common errors and keeps your users relatively safe from bad user experience when using 464XLAT and NAT64.

This also prevent issues when AAAA record contains an IPv6 documentation prefix, that is inside 2000::/3 block.

“Break-dnssec yes” configuration directive:

By default, DNS64 module does not process queries that request DNSSEC data (DO = 1) or that have DNSSEC RRs in the answer. Setting break-dnssec yes will override this default and cause policy processing on all DNSSEC queries. However, the constructed response will not have any DNSSEC records added and therefore cannot be verified by the client (it may have the unintended consequence of looking like a bogus response or even an injection attack to the client).
macbook# dig AAAA 4.go6.si @2001:67c:27e4:15::64 +dnssec

; <<>> DiG 9.8.3-P1 <<>> AAAA 4.go6.si @2001:67c:27e4:15::64 +dnssec
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 13683
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 0, AUTHORITY: 4, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;4.go6.si. IN AAAA ???

;; AUTHORITY SECTION:
go6.si. 3502 IN SOA nsec1.go6lab.si. root.go6.si. 2017010300 86400 7200 36000000 3600
ngo6.si. 3502 IN RRSIG SOA 8 2 3600 20170117190417 20170103072329 23274 go6.si. GeTWHLNSb1aASVADyIxviIsZLq0SY AUCR364SWTzuS7vkghSlaubYHo/GHWjSCj8N4LiZMBsbFXUXQrcsbN9J 27c5SF5S/0de+Ty00TtNvL46ETjVlalKdETGDHMAnGuZ02Deful14C4PioVqEGCcNdKmNLxQ0BZORELOGGe T8gZruFJryrXyHo2YD83PERF7hpyp+ZIV91wOEMbznDi7131Igw3TNcwa V7aN80b6I8vfiqRZ62ONU2jBuEq yLWjw==
c0jvnrvuo3s4v19heogvskgfrvoro8nvdp.go6.si. 3502 IN RRSIG NSEC3 8 3 3600 20170113072032 2016122920157 23274 go6.si. fJc+IjWZdAZt+cvMjw++8qzOy01lq9T Qe3Xolbll11900933mzeAYwaBqBBHJejkwyFY0YnlZ5UyDw7TS9FSl00 d1sEa6XYigy7JIN6wxyFZy+CJ U1tx WD37/HTS3uBXeit/s9zgPfLttoQw3xMoqIWa97CkLDvFBr01lzM8SOKn zkduQuGIYrBJRRdWU9U65sJIIjogOE0MttxULgyzhrM24NVvN6aX Z0Ec1Yo1Tbpoy0jY3WJedsiGHpe1TTZjGRCB 0gjMiQ==
c0jvnrvuo3s4v19heogvskgfrvoro8nvdp.go6.si. 3502 IN NSEC3 1 0 5 82EEE992264B06E4 C83N2GVQ3KBH0KUQRMBHUD63IDHLBSU A

;; Query time: 51 msec
;; WHEN: Tue Jan 3 00:59:11 2017
;; MSG SIZE rcvd: 766
macbook# dig AAAA 4.go6.si @2001:67c:27e4:15::64 +dnssec
;
;; DiG 9.8.3-P1 <<> AAAA 4.go6.si @2001:67c:27e4:15::64 +dnssec
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 45896
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 1, AUTHORITY: 3, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;4.go6.si. IN AAAA

;; ANSWER SECTION:
4.go6.si. 3600 IN AAAA 2001:67c:27e4:1064::5bef:603d

;; AUTHORITY SECTION:
go6.si. 3600 IN NS nsec1.go6lab.si.
go6.si. 3600 IN NS nsec2.go6lab.si.
go6.si. 3600 IN RRSIG NS 8 2 3600 20170110170427 20161227211300 23274 go6.si. iWA2Cfu
uKS6Ve0PwY7WXUPclfQRFuVWamHv1bagVxo6zp86WAHUdGuitdX3xHC3 Klr+Ldi/5nLDJ1eFU9BJ1 fU4qe0FQjWaiY9zzv0MZjaIV8K k3GUDC
XLJ98BAIT6iDoSdYPbHBeSk2/uxVzwQXjynRDgN75VM40fUJHJhTikuI2D/2PenM6Lv17DNwx9FF44gM1GG7W60/4L50j/4Ey3xE+xuV4BS
m+ZP1nhQW9NR0Bf/Rugjv/1tlQ5 qYAJCQ==

;; Query time: 384 msec
;; WHEN: Tue Jan  3 10:06:24 2017
;; MSG SIZE  rcvd: 406

With break-dnssec yes;
(Enigma Machine)
When deploying in real life you need “fixes”

(Stream of Consciousness)

**Unbound** example of DNS64 configuration in go6lab:

```
server:
  module-config: "dns64 validator iterator"
  dns64-prefix: 2001:67c:27e4:64::0/96
  private-address: 0::/3
  private-address: 4000::/2
  private-address: 8000::/1
  private-address: 2001:DB8::/32
  # private-address: 64:FF9B::/96
  # private-address: ::ffff:0:0/96
  # private-address: ::1/128
  # private-address: ::/128
interface: 2001:67c:27e4::64
```
DNS64: DNS Extensions for Network Address Translation from IPv6 Clients to IPv4 Servers

Abstract

DNS64 is a mechanism for synthesizing AAAA records from A records. DNS64 is used with an IPv6/IPv4 translator to enable client-server communication between an IPv6-only client and an IPv4-only server, without requiring any changes to either the IPv6 or the IPv4 node, for the class of applications that work through NATs. This document specifies DNS64, and provides suggestions on how it should be deployed in conjunction with IPv6/IPv4 translators.
Oops: firewalls.com

IPv4

NAT64
IPv4

NAT64

Oops: firewalls.com

Still waiting for that timeout...
Oops: firewalls.com

Indeed, very secure...

IPv4

NAT64
Indeed, very secure, running on ::ffff:209.59.134.0…
Good DNS, bad server: notificaciones.060.es

IPv4
IPv6 and NAT64
Good DNS, bad geolocation tool: shrani.si
No AAAA, server still confused: uc.cn
Good server, bad content: ebay-kleinanzeigen.de

IPv4 and NAT64

IPv6
Good server, bad content: thegatewaypundit.com
Non working AAAA record = broken NAT64 😊
Working AAAA record and IPv4-only content
Working AAAA record and IPv4-only content
How to test for all of these things?

(The Looking Glass)

There are many things to test:

Test on NAT64

Test on IPv6-only

Did all resources (images, stylesheets, scripts) load ok?

Does it look good to the user?

Do we see any pMTUd issues?

One service to test them all: NAT64Check !!!
### nat64check

(Caught in a web)

Check: http://www.example.com/

<table>
<thead>
<tr>
<th>URL</th>
<th>When</th>
<th>NAT64</th>
<th>IPv6-only</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://dehumida.si">http://dehumida.si</a></td>
<td>Nov. 27, 2016, 10:27 p.m.</td>
<td>100%</td>
<td>Unreachable</td>
</tr>
<tr>
<td><a href="http://osvekadoina.si">http://osvekadoina.si</a></td>
<td>Nov. 27, 2016, 10:27 p.m.</td>
<td>100%</td>
<td>Unreachable</td>
</tr>
<tr>
<td><a href="http://puhar.si">http://puhar.si</a></td>
<td>Nov. 27, 2016, 10:27 p.m.</td>
<td>100%</td>
<td>Unreachable</td>
</tr>
<tr>
<td><a href="http://vi4-renzel.si">http://vi4-renzel.si</a></td>
<td>Nov. 27, 2016, 10:27 p.m.</td>
<td>100%</td>
<td>Unreachable</td>
</tr>
<tr>
<td><a href="http://domstarejsihidlia.si">http://domstarejsihidlia.si</a></td>
<td>Nov. 27, 2016, 10:27 p.m.</td>
<td>100%</td>
<td>Unreachable</td>
</tr>
<tr>
<td><a href="http://mirza.si">http://mirza.si</a></td>
<td>Nov. 27, 2016, 10:27 p.m.</td>
<td>100%</td>
<td>Unreachable</td>
</tr>
<tr>
<td><a href="http://program-za-trgovino.si">http://program-za-trgovino.si</a></td>
<td>Nov. 27, 2016, 10:27 p.m.</td>
<td>100%</td>
<td>Unreachable</td>
</tr>
<tr>
<td><a href="http://fitness-epicenter.si">http://fitness-epicenter.si</a></td>
<td>Nov. 27, 2016, 10:27 p.m.</td>
<td>100%</td>
<td>Unreachable</td>
</tr>
<tr>
<td><a href="http://vtuoz.si">http://vtuoz.si</a></td>
<td>Nov. 27, 2016, 10:27 p.m.</td>
<td>100%</td>
<td>Unreachable</td>
</tr>
<tr>
<td><a href="http://chill.si">http://chill.si</a></td>
<td>Nov. 27, 2016, 10:27 p.m.</td>
<td>100%</td>
<td>Unreachable</td>
</tr>
</tbody>
</table>
### nat64check

**(Caught in a web)**

Check: [http://www.example.com/](http://www.example.com/)

<table>
<thead>
<tr>
<th>URL</th>
<th>When</th>
<th>NAT64</th>
<th>IPv6-only</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://www.sdeval.si">http://www.sdeval.si</a></td>
<td>Nov. 27, 2016, 10:25 p.m.</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td><a href="http://myunikat.si">http://myunikat.si</a></td>
<td>Nov. 27, 2016, 10:21 p.m.</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td><a href="http://korsic.si">http://korsic.si</a></td>
<td>Nov. 27, 2016, 10:19 p.m.</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td><a href="http://graj-tenis.si">http://graj-tenis.si</a></td>
<td>Nov. 27, 2016, 10:16 p.m.</td>
<td>100%</td>
<td>97%</td>
</tr>
<tr>
<td><a href="http://www.vrticosbos.si">http://www.vrticosbos.si</a></td>
<td>Nov. 27, 2016, 10:08 p.m.</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>http:/89.si</td>
<td>Nov. 27, 2016, 10:06 p.m.</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td><a href="http://renovatio.si">http://renovatio.si</a></td>
<td>Nov. 27, 2016, 10:02 p.m.</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td><a href="http://ki-ki-do.si">http://ki-ki-do.si</a></td>
<td>Nov. 27, 2016, 10:02 p.m.</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td><a href="http://proekt-it.si">http://proekt-it.si</a></td>
<td>Nov. 27, 2016, 9:58 p.m.</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td><a href="http://fenstino.si">http://fenstino.si</a></td>
<td>Nov. 27, 2016, 9:58 p.m.</td>
<td>100%</td>
<td>97%</td>
</tr>
</tbody>
</table>
nat64check

www.netbk.co.jp

Image match

NAT64 85%

IPv4-only

IPv6-only

Resources match
(The Shattered Fortress)
We don’t “fix” brokenness

(Lines in the Sand)

For NAT64/DNS64 implementations in go6lab, we try to avoid AAAA bogus records and all our DNS64 configs are tuned like we have shown in previous slides.

For NAT64 Check tool, however, we don’t exclude anything. We are trying to show how it is in reality and not mask problems. Those people, causing problems needs to find out the situation and fix it.

https://nat64check.go6lab.si/
https://nat64check.ipv6-lab.net/
How does this work?

(Illumination Theory)

Four servers:

- Management server and web interface
- Server with only IPv4
- Server with only IPv6
- Server with NAT64

Go6lab specific: All servers are virtual machines (LIX) on Proxmox 4.2 virtualization cluster

IPv6-lab.net specific: All running on virtual machines (Ubuntu) on a VMware cluster

Using phantomjs as a command-line browser

- Keep track of all loaded (or not) resources
- Make a screenshot when the page has loaded

Compare images with each other

Check for resources that could not be loaded
How does this work?

(Images and Words)

IPv4-only Web client
IPv6-only Web client
IPv6-only + NAT64/DNS64 Web client

DNS64
NAT64

Master and web interface
IPv4 internet
IPv6 internet
Run the test and calculate the score

(Take the Time)

Enter the domain name or URL that you would like to test and submit it to the system.

The process of starting all needed browsers, get the results, grab the screen and calculate/compare all the results takes some time, so please be patient.
Run the test and calculate the score

*(Take the Time)*

You’ll see something like this for some time after you start the measurement...

The system will show you when the test was requested, when test was started and will also show you the result in the same page after some time.

No need to hit the “Back” button, it will happen, just give it some time. We promise. Scout’s word of honor. :)

*Image of a test interface showing a test in progress and a scheduled time for the test.*
Wash, rinse and repeat

(Lifting Shadows of a Dream)

• Test your website
  Are you sure all the images, analytics scripts etc. load over IPv6?
  If you don’t have IPv6 on your server:
    Did you hard-code any IPv4 addresses in your HTML code? In this case NAT64/DNS64 will fail. You might get away with it on 464XLAT, but don’t count on it.
    Why you don’t have IPv6 on your server yet?
  If things fail - make sure you are persistent enough to find the issues. There are millions of potential viewers of your content on IPv6-only connection at this precise moment, make sure you don’t mess it up and that our tool shows the 100% good result.

• Test again
• Repeat until your website is no longer broken
• If you are fixing a DNS misconfiguration - it might take some time before our tool gets and sees the new and fixed AAAA value. DNS propagation is slow.
Non properly working target domain = no content!

(Learning To Live)
NAT64Check code available on GITHUB
(In Constant Motion)

• All code is free and open and it lives on GitHub.
• Feel free to use it and/or contribute:
  https://github.com/sjm-steffann/nat64check
Conclusions, questions, suggestions?
(Act II: Scene Eight: The Spirit Carries On)

If you are content provider: test how people sees your content from different environments

If you are connectivity provider: test how IPv6-only and 464XLAT/NAT64-DNS64 users sees content

If both: all of the above 😊

You might be surprised. Well, you probably will be surprised.

https://nat64check.go6lab.si/
https://nat64check.ipv6-lab.net/

Jan Žorž
zorz@isoc.org

Sander Steffann
sander@steffann.nl