Improving NAT-PT

marcelo bagnulo

v4v6 interim meeting

Montreal

RFC4966 issues with NAT-PT

- Issues Unrelated to an DNS-ALG
- Issues Exacerbated by the Use of DNS-ALG
- Issues Directly Related to Use of DNS-ALG

- Let’s go through them and see what can we improve

Issues Unrelated to an DNS-ALG (I)

- Issues with Protocols Embedding IP Addresses
  - Generic issue with any NAT
    - Address in data won't get trasnlated
  - "Solution" in v4: BCP for NAT traversal (ICE etc)
  - Reccomendation for NAT64: Make NAT64 compatible with NAT traversal techniques i.e. BEHAVE requirements
- NAPT-PT Redirection Issues
  - Works only for UDP and TCP
  - Reccomendation for NAT64: Define support for other transports DCCP, SCTP, IPSec
    - Follow guidelines from BEHAVE

Issues Unrelated to an DNS-ALG (II)

- NAT-PT Binding State Decay
  - What is worse, 2266 doesn't specify timer
  - Reccomendation for NAT64: Follow BEHAVE requirements for mapping refresh timer
  - At least apps know what to expect, can use ICE
- Loss of Information through Incompatible Semantics
  - Flow label, Extension headers, some ICMP
  - Not clear how severe this issue is (as per 4966)
  - Reccomendation for NAT64: none
  - Diffserv Code points
  - Reccomendation: copy DSCP value?

Issues Unrelated to an DNS-ALG (III)

- NAT-PT and Fragmentation
  - Reccomendation for NAT64: need more discussion, see Cullen's comments
- NAT-PT Interaction with SCTP and Multihoming
  - Reccomendation for NAT64: properly define how to handle SCTP in NAT64 (see sctp in behave)
- NAT-PT as a Proxy Correspondent Node for MIPv6
  - Reccomendation for NAT64: none (or define NAT64 to be CN)
- NAT-PT and Multicast
  - Reccomendation for NAT64: more study is needed

Issues Unrelated to an DNS-ALG preliminary conclusions

- Most of the issues can be solved or improved by doing a proper (more complete) specification
  - Compliant with BEHAVE requirements
  - Compatibe with NAT traversal techniques
  - Describing the support for all required protocols

Issues Exacerbated by the Use of DNS-ALG (I)

- Network Topology Constraints Implied by NAT-PT
  - DNS queries and data packets must flow the same path
  - May be solved for NAT64
    - Depends on the binding betwenn Pref64 and the NAT64 box
    - In NAT64 it is possible to place the DNS64 in the DNS server rahter than on path
  - More difficult to solve for dynamic NAT46
    - Seems intrinsic to the reduced v4 addr space
  - Reccomendation: separate NAT64 from NAT46, allowing to build NAT64 wihtout the problem, define DNS64 as a DNS server or resolver capability
- Scalability and Single Point of Failure Concerns
  - In NAT64 it is possible to decouple NAT64 and DNS64
  - It would be possible to define inter NAT64 protocols to deal with this
  - Reccomendation: decouple NAT64 from DNS64 and not clear we want to do more than this

Issues Exacerbated by the Use of DNS-ALG (II)

- Issues with Lack of Address Persistence
  - Timeout between the DNS query and the data packet
  - Can be solved for NAT64, not easy for NAT46
  - Also, timeout between different sessions
  - Reccomendation: make NAT64 separate from NAT46 and compliant with behave requirements, so apps know what to expect
- DoS Attacks on Memory and Address/Port Pools
  - Both data packet based and DNS query based
  - DNS query based can be solved for NAT64 (not easy for NAT46)
  - Reccomendation: make NAT64 separate from NAT46 and define heuristics to deal with the DoS attacks in data packets

Issues Exacerbated by the Use of DNS-ALG
Preliminary conclusions

- Most severe issues can be solved for NAT64 but not for NAT46
  - Major change from NATPT: define DNSALG as a function of the DNS server or DNS resolver rather than an on path spoofer
- We should separate NAT64 from NAT46
- This allows to have NAT64 boxes that are much less malign than full NATPT
- We need to think about the implications of the binding between Pref64 and NAT64 box though.
- Note that some scenarios don’t need DNSALG at all
  - Maybe we need to seprate the NAT64 from the DNS64 spec?

Issues Directly Related to Use of DNS-ALG (I)

- Address Selection Issues when Communicating with Dual-Stack End-Hosts
  - For v6/v4 only hosts initiating a communication to a dual stack
    - Can be mitigated at the DNS64 (no synthesis if real AAAA exists, possible when DNS64 is part of DNS server resolver)
  - For a dual stack communicaing to a v6 only node
    - Can be mitigated by playing with rfc3484 or with EDNS0 option
    - Difficult to provde automatically for legacy nodes though
    - Consider the tradeoffs for different options for Pref64
  - Reccomendation: use described tools for mitigate the issue
- Inappropriate Translation of Responses to A Queries
  - DNSALF don't allow hosts to see the real RR even if they want to
  - Non issue for DNS64 when located at the DNS server resolver

Issues Directly Related to Use of DNS-ALG (II)

- DNS-ALG and Multi-Addressed Nodes
  - Issue of creating multiple nat bindings for nodes with multiple addresses
  - Non issue for NAT64, since state is created with data packets when DNS64 is provided at the DNS server resolver
  - Real issue for NAT46
  - Reccomendation: solve the issue for NAT64 decoupling the DNS64 and putting it in the DNS server or resolver
- Limitations on Deployment of DNS Security Capabilities
  - Can be solved if the DNS64 is co-located with the valiating server or validating host
  - Difficult to solve for NAT46
  - Reccomendation: place DNS64 in the DNS server or resolver

Conclusions and reccomendations

- Conclusion: NAT64 and NAT46 are very different beasts with very different limitations
  - Many critical limitations can be mitigated for NAT64
  - Of the remaining, some only apply to DNS64 and not to NAT64
- Reccomendation: define NAT64 separated from NAT46
  - Even consider specifying NAT64 and DNS64 in separated specs
- Conclusion: Many of the limitations result from assuming DNSALG spoofs DNS queries
- Reccomendation: Define DNS64 as an additional functionality located in the DNS server or DNS resolver
- Conclusion: many of the limitations are due to lack of the NATPT specification itself
- Reccomendation: make a new spec, using BEHAVE