Network Working Group
Request for Comments: 3424
Category: Informational

L. Daigle, Ed.
Internet Architecture Board
IAB

November 2002

IAB Considerations for UNilateral Self-Address Fixing (UNSAF) Across Network Address Translation

Status of this Memo

Copyright Notice

Abstract

As a result of the nature of Network Address Translation (NAT) Middleboxes, communicating endpoints that are separated by one or more NATs do not know how to refer to themselves using addresses that are valid in the addressing realms of their (current and future) peers. Various proposals have been made for "UNilateral Self-Address Fixing (UNSAF)" processes. These are processes whereby some originating endpoint attempts to determine or fix the address (and port) by which it is known to another endpoint - e.g. to be able to use address data in the protocol exchange, or to advertise a public address from which it will receive connections.

This document outlines the reasons for which these proposals can be considered at best as short term fixes to specific problems and the specific issues to be carefully evaluated before creating an UNSAF proposal.

1. Introduction

As a result of the nature of Network Address (and port) Translation (NAT) Middleboxes, communicating endpoints that are separated by one or more NATs do not know how to refer to themselves using addresses that are valid in the addressing realms of their (current and future) peers - the address translation is locked within the NAT box. For some purposes, endpoints need to know the addresses (and/or ports) by which they are known to their peers. There are two cases: 1) when the client initiates communication, starting the communication has the side effect of creating an address binding in the NAT device and

allocating an address in the realm that is external to the NAT box; and 2) a server will be accepting connections from outside, but because it does not initiate communication, no NAT binding is created. In such cases, a mechanism is needed to fix such a binding before communication can take place.

"UNilateral Self-Address Fixing (UNSAF)" is a process whereby some originating process attempts to determine or fix the address (and port) by which it is known - e.g. to be able to use address data in the protocol exchange, or to advertise a public address from which it will receive connections.

There are only heuristics and workarounds to attempt to achieve this effect; there is no 100% solution. Since NATs may also dynamically reclaim or readjust translations, "keep-alive" and periodic re-polling may be required. Use of these workarounds MUST be considered transitional in IETF protocols, and a better architectural solution is being sought. The explicit intention is to deprecate any such workarounds when sound technical approaches are available.

2. Architectural issues affecting UNSAF Systems

Generally speaking, the proposed workarounds are for cases where a standard protocol communication is to take place between two endpoints, but in order for this to occur, a separate step of determining (or fixing) the perceived address of an endpoint in the other endpoint's addressing realm is required. Proposals require that an endpoint seeking to "fix" its address contact a participating service (in a different address realm) to determine (reflect) its address. Thus, there is an "UNSAF client" partnering with some form of "UNSAF service" that may or may not be associated with the target endpoint of the actual desired communication session. Throughout this memo, the terms "UNSAF server" and "UNSAF service" should be understood to generically refer to whatever process is participating in the UNSAF address determination for the originating process (the UNSAF client).

Any users of these workarounds should be aware that specific technical issues that impede the creation of a general solution include:

- there *is* no unique "outside" to a NAT - it may be impossible to tell where the target endpoint is with respect to the initiator; how does an UNSAF client find an appropriate UNSAF server to reflect its address? (See Appendix C).

- specifically because it is impossible to tell where address realms are bounded ("inside" or "outside", "private" or "public", or several "private" realms routing traffic), an address can only be determined relative to one specific point in the network. If the UNSAF service that reflected an UNSAF client's address is in a different NAT-masqueraded subnet from some other service X that the client wishes to use, there is _no_ guarantee that the client's "perceived" address from the UNSAF partner would be the same as the address viewed from the perspective of X. (See Appendix C).

- absent "middlebox communication (midcom)", there is no usable way to let incoming communications make their way through a middlebox (NAT, firewall) under proper supervision. By circumventing the NAT, UNSAF mechanisms may also (inadvertently) circumvent security mechanisms. The particular danger is that internal machines are unwittingly exposed to all the malicious communications from the external side that the firewall is intended to block. This is particularly unacceptable if the UNSAF process is running on one machine which is acting on behalf of several.

- proposed workarounds include the use of "ping"-like address discovery requests sent from the UNSAF client (initiator) to the UNSAF server (listener), to which the listener responds with the transport address of the initiator - in the address realm of the listener. However, with connection-less transports, e.g. UDP, IPsec ESP, etc., an UNSAF process must take care to react to changes in NAT bindings for a given application flow, since it may change unpredictably.

- if the UNSAF client uses periodic retries to refresh/reevaluate the address translation state, both the UNSAF client and the UNSAF server are required to maintain information about the presumed state of the communication in order to manage the address illusion.

- since the UNSAF server is not integrated with the middlebox, it can only operate on the assumption that past behavior is a predictor of future behavior. It has no special knowledge of the address translation heuristic or affecting factors.

- the communication exchange is made more "brittle" by the introduction of other servers (UNSAF servers) that need to be reachable in order for the communication to succeed - more boxes that are "fate sharing" in the communication.

Workarounds may mitigate some of these problems through tight scoping of applicability and specific fixes. For example:

- rather than finding the address from "the" outside of the NAT, the applicability of the approach may be limited to finding the "self-address" from a specific service, for use exclusively with that service.

- limiting the scope to outbound requests for service (or service initiation) in order to prevent unacceptable security exposures.

3. Practical Issues

From observations of deployed networks, it is clear that different NAT box implementations vary widely in terms of how they handle different traffic and addressing cases.

Some of the specific types of observed behaviors have included:

- NATs may drop fragments in either direction: without complete TCP/UDP headers, the NAT may not make the address translation mapping, simply dropping the packet.

- Shipping NATs often contain Application Layer Gateways (ALGs) which attempt to be context-sensitive, depending on the source or destination port number. The behavior of the ALGs can be hard to anticipate and these behaviors have not always been documented.

- Most NAT implementations with ALGs that attempt to translate TCP application protocols do not perform their functions correctly when the substrings they must translate span across multiple TCP segments; some of them are also known to fail on flows that use TCP option headers, e.g. timestamps.

- NAT implementations differ markedly in their handling of packets. Quite a few only really work reliably with TCP packets, not UDP. Of the ones that do make any attempt to handle UDP packets, the timers aging out flows can vary widely making it challenging to predict behavior.

- Variation in address and port assignments can be quite frequent - on NATs, port numbers always change, and change unpredictably; there may be multiple NATs in parallel for load-sharing, making IP address variations quite likely as well.

4. Architectural Considerations

By distinguishing these approaches as short term fixes, the IAB believes the following considerations must be explicitly addressed in any proposal:

1. Precise definition of a specific, limited-scope problem that is to be solved with the UNSAF proposal. A short term fix should not be generalized to solve other problems. Such generalizations lead to the the prolonged dependence on and usage of the supposed short term fix -- meaning that it is no longer accurate to call it "short term".

2. Description of an exit strategy/transition plan. The better short term fixes are the ones that will naturally see less and less use as the appropriate technology is deployed.

3. Discussion of specific issues that may render systems more "brittle". For example, approaches that involve using data at multiple network layers create more dependencies, increase debugging challenges, and make it harder to transition.

4. Identify requirements for longer term, sound technical solutions; contribute to the process of finding the right longer term solution.

5. Discussion of the impact of the noted practical issues with existing deployed NATs and experience reports.

5. Security Considerations

As a general class of workarounds, UNSAF proposals may introduce security holes because, in the absence of "middlebox communication (midcom)", there is no feasible way to let incoming communications make their way through a firewall under proper supervision: respecting the firewall policies as opposed to circumventing security mechanisms.

Appendix A. IAB Members at the time of this writing:

Harald Alvestrand
Ran Atkinson
Rob Austein
Fred Baker
Leslie Daigle
Steve Deering
Sally Floyd
Ted Hardie
Geoff Huston
Charlie Kaufman
James Kempf
Eric Rescorla
Mike St. Johns

Appendix B. Acknowledgements

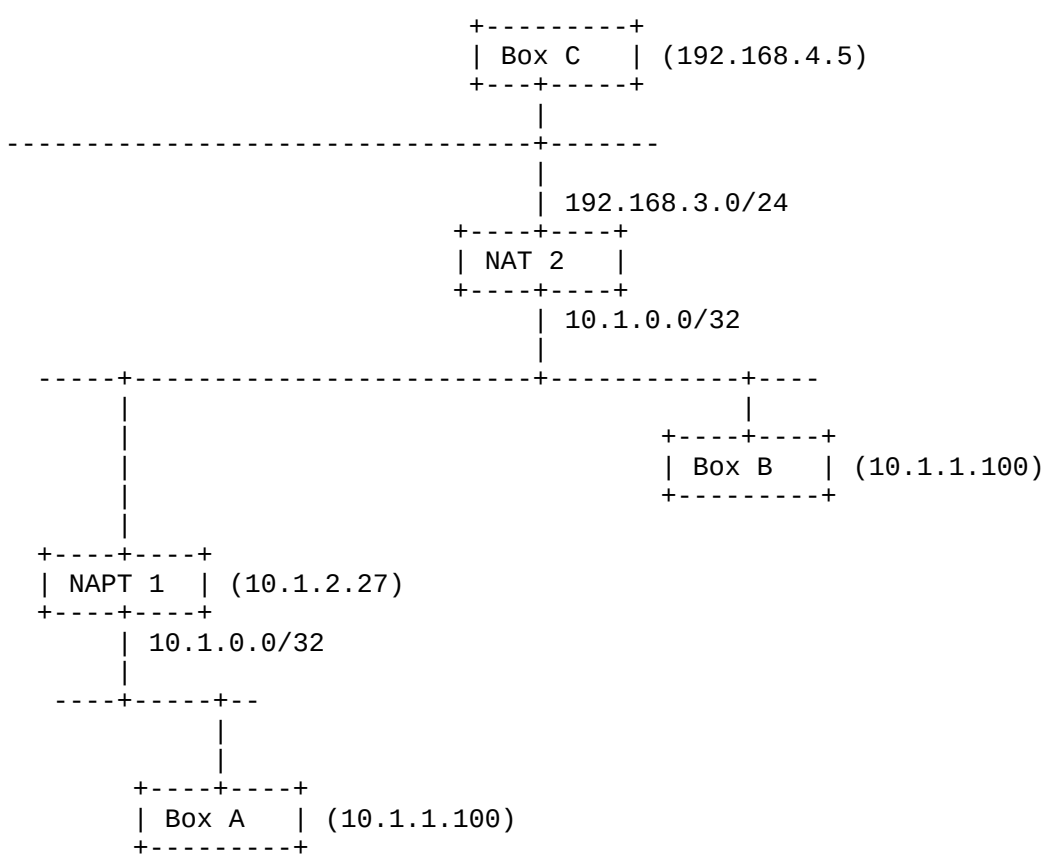
This document has benefited greatly from detailed comments and suggestions from Thomas Narten, Bernard Aboba, Keith Moore, and James Woodyatt.

This document was originally drafted when the following people were part of the IAB: Steve Bellovin, Brian Carpenter, Jon Crowcroft, John Klensin and Henning Schulzrinne; it has benefited considerably from their contributions and review.

Appendix C. Example NAT Configuration Scenario

C.1 Generic NATed Network Configuration

Here is one sample scenario wherein it is difficult to describe a single "outside" to a given address realm (bridged by NAPTs). This sort of configuration might arise in an enterprise environment where different divisions have their own subnets (each using the same private address space); the divisions are connected so that they can pass traffic on each others' networks, but to access the global Internet, each uses a different NAPT/firewall:



From the perspective of Box B, Box A's address is (some port on) 10.1.2.27. From the perspective of Box C, however, Box A's address is some address in the space 192.168.3.0/24.

C.2 Real World Home Network Example

James Woodyatt provided the following scenario, based on current examples of home networking products:

- the customer has existing Internet service from some broadband service provider, using e.g. a DSL line connected to an appliance that integrates a DSL modem with a NAT router/firewall.
- these devices are sometimes packaged with automated provisioning firmware, so the customer may view them as part of what their ISP provides them.
- later, the customer wants to use a host with only a wireless LAN interface, so they install a wireless access point that ships in its default configuration with NAT and a DHCP server enabled.
- after this, the customer has a wired LAN in one private address realm and a wireless LAN in another private address realm.

Furthermore, most customers probably have no idea what the phrase "address realm" means and shouldn't have to learn it. All they often know is that the printer server is inaccessible to the wireless laptop computer. (Why? Because the discovery protocol uses UDP multicast with TTL=1, but that's okay because any response would just be dropped by the NAT anyway, because there's no ALG.)

Authors' Addresses

Leslie Daigle
Editor

Internet Architecture Board
IAB
EMail: iab@iab.org

Full Copyright Statement

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.