Network Working Group
Internet-Draft
Expires: September 10, 2009

C. Vogt
Ericsson

March 9, 2009

Qualifying the Harmfulness of Address Translation
draft-vogt-address-translation-harmfulness-01.txt

Status of this Memo

Copyright Notice

Abstract

Address translation is widely considered harmful because its existing variants conflict with well-established design principles of the Internet engineering community. Still, address translation has become common practice despite technical problems because it constitutes an easy-to-deploy solution to a set of common operational needs. Since some of these needs will continue to exist in IP version 6, there is concern within the Internet engineering community about the potential proliferation of harmful technology from IP version 4 to IP version 6. This document addresses this concern. It compares feasible address translator designs with respect to harmful implications, explains why the problems of address translation, as used today, are to a significant extent specific to IP version 4, and shows how the problems can be mitigated in IP version 6.

Table of Contents

1. Introduction

One of the design principles most well-heeded by the Internet engineering community is that addresses have end-to-end validity and do not change in packets en route. This principle is being challenged by the widespread use of address translation on the Internet. Address translators rewrite addresses in packets en route, typically at network borders, to satisfy network operators' desire for provider independence, topology concealment, or conservation of global addresses. The incentives for deploying address translation are strong, even though the technique, as used today for IP version 4, has profound drawbacks. Since the incentives are furthermore partly independent of the IP version, there is concern within the Internet engineering community about the potential proliferation of harmful technology from IP version 4 to IP version 6.

This document addresses this concern by qualifying the harmfulness of feasible address translator designs in IP version 4 and 6. The document makes three contributions to this end: First, it compares potentially harmful implications of different address translator designs. Second, it infers that many of the problems with address translation as deployed today are due to address overloading, a technique that helps conserving global addresses, rather than because addresses are rewritten in packets en route. Third, the document argues that, while address overloading is inevitable in IP version 4 due to the shortage of global addresses [REF], address translation in IP version 6 does not require address overloading and could hence, if designed rightly, be considerably less problematic than address translation in IP version 4.

The following sections of this document are organized as follows: Section 2 explains the purposes for which address translation is used, and section 3 identifies the components of address translation that are necessary to achieve these purposes. Section 4 describes the implications of address translation, and it shows that many resulting problems can be attributed to address overloading being one of the components. Section 5 concludes that these problems are avoidable in address translation for IP version 6, where address overloading is dispensable.

2. Purposes of Address Translation

Network operators frequently apply address translation to separate the addresses they use locally in their networks from the global addresses at which the networks are reachable from the Internet. They do this for any of the following three purposes:

- Provider independence: Network operators desire the flexibility to change providers at low cost, in order to avoid lock-in to any particular provider.

- Topology concealment: Network operators may want to hide a network's local topology from the rest of the Internet for security reasons.

- Global address conservation: Network operators see increasing pressure to conserve global IP version 4 addresses due to the imminent runout of unallocated global IP version 4 addresses [REF].

A usecase of address translation to achieve provider independence is in residential networks or small enterprise networks, which either cannot afford, or are not eligible for, global provider-independent addresses. Internet registries restrict assignments of global provider-independent addresses to networks of sufficient size in an effort to prevent excessive load on the global routing system. This is deemed necessary because global provider-independent addresses cannot be aggregated as efficiently as provider-assigned addresses, and hence increase the load of the global routing system [REF]. The recommended practice for these networks is to use address space assigned by their providers, and to renumber in the event of a provider change. Renumbering can be a smooth process in sufficiently optimized networks. Unfortunately, though, experience [REF] shows that the process often involves substantial manual labor, and is hence costly and time-consuming. Although the addresses of hosts could be changed automatically via DHCP, routers and servers still typically have manually configured addresses, and would therefore have to be renumbered manually. Old addresses may also be preconfigured in applications, firewalls, and operations and management systems [REF]. Address translation helps avoiding network renumbering without global provider-independent addresses. Network operators can use local provider-independent addresses internally, and they translate those onto global addresses assigned by their providers.

A security-related usecase of address translation to for denial-of-service attack protection. The usual network-topological assignment of addresses provides a means to infer the topology of a network by remote hosts. Attackers may use this information to identify attack targets. For example, a denial-of-service attack against a server may more easily be executed via a host on the server's link, and such a host can typically be identified based on comparing its IP address to the IP address of the server in question. Address translation can conceil the internal topology of a network, by mapping local and global addresses such that the topological structuring of local

addresses cannot be derived from global addresses.

The conservation of global addresses provides a third usecase for address translation. It is of common interest among operators of IP version 4 networks, for which the dire shortage of global IP version 4 addresses makes network expansion difficult. This, in turn, can have an negative impact on revenue. Address translation helps conserving global addresses because it allows multiple hosts with separate local addresses to share one global address.

3. Functional Components of Address Translation

In order to accomplish the purposes identified above, address translation incorporates two functional components:

- Address rewriting: The local and global addresses of a network are mapped, and swapped accordingly in packets leaving or entering the network.

- Address overloading: Multiple local addresses are mapped onto a single global address. To enable demultiplexing of packets received at a global address back onto the right local address, address translators store the corresponding local address as connection-specific state, and they use port numbers in the received packets as indexes into this state.

Address rewriting affords provider independence and topology concealment. Provider independence is achieved through the decoupling of a network's local addresses from the global addresses assigned by the network's provider. The local addresses hence do not need to change if the network changes providers. This eliminates the need to renumber. Topology hiding can be achieved either by overloading a single global address with a large portion of local addresses, or by choosing, and keeping secret, a non-trivial permutation based on which local and global addresses are mapped.

Simple address rewriting without address overloading requires at least one global address per host, which maps one-to-one onto its corresponding global address. To conserve global addresses, it is necessary to have multiple hosts share one global address. This can be achieved by combining address rewriting with address overloading.

Given that the functional component of address overloading is optional, two types of address translation can be distinguished:

- One-to-one address translation, which consists of address rewriting without address overloading, and which achieves provider independence and topology concealment.

- Many-to-one address translation, which combines address rewriting with address overloading, and which achieves global address conservation in addition to provider independence and topology concealment.

In the following, the architectural impact of address translation will be evaluated separately for its two types.

4. Implications of Address Translation

Since address translation changes the way hosts are addressed and packets are forwarded, it has a signifiant impact on Internet architecture. The analysis below examines the harmfulness of this impact. It shows potential problems that result from address translation, and analyzes the feasibility and cost of mitigating those problems. One-to-one address translation and many-to-one address translation are thereby considered separately, since the functional components of address rewriting and address overloading each have their own architectural impact.

4.1. One-to-One Address Translation

Since address translation renders hosts reachable at different addresses depending on the location of a given peer, peers must be enabled to discover and use one of the host's addresses they can reach. A peer's location hence governs which of a host's addresses can be used in the IP headers or, for referrals, in the payloads of packets exchanged with the peer.

Since address translators rewrite only IP headers, addresses referred to in packet payloads may have to be global end to end. This calls for support in hosts with translated addresses as well as their authoritative DNS servers. Authoritative DNS servers must refer peers to the global addresses of a host if the intended communication will go via an address translator. Hosts must use their global addresses for address referrals in packets they send to peers via an address translator, and they must recognize their global addresses in packets received via an address translator. An example of a protocol that may require hosts to refer to their global address is the Session Initiation Protocol [REF], a protocol used to bootstrap multimedia applications. An example of a protocol that may require hosts to recognize their global address in received packets is ICMP [REF]. ICMP error and notification messages may include a copy of

the packet by which they were triggered. The triggering packet, in turn, may include translated, global addresses, which are not reverse-translated when included in the payload of an ICMP message.

These properties imply that, for one-to-one address translation to function properly, two components are needed:

- Obtainability of global addresses: A host with translated addresses and its authoritative DNS server must be able to obtain the global addresses of the host. They must either be configured with the global addresses or have means to dynamically discover them.

- Peer-dependent address selection: Addresses referred to in packet payloads, including responses from authoritative DNS servers, must be reachable from the peer's location.

The following shows that both components can be realized in a simple and non-disruptive manner based on suitable host support. Host support is critical, though, since in its absence one-to-one address translation breaks applications that use address referrals.

Obtainability of global addresses may in the simplest case be realized by configuring applications and authoritative DNS servers with the global addresses of hosts. In the case of authoritative DNS servers, this approach corresponds to common practice. It is also tractable because, in one-to-one address translation, global addresses are static, hence the configuration usually does not have to be modified. While pre-configuration would in principle also suffice to make applications aware of global addresses, auto-discovery of global addressses is typically preferred to reduce administrative complexity. Standard methods exists for this [REF]. They discover a global address by inquiring of infrastructure what a packet's source address looks like after translation. The methods were broadly introduced to handle many-to-one address translation in the IP version 4 Internet. Still, since neither the methods nor the infrastructure they leverage can be expected to always be available, there may be situations in which applications will fail in the presence of address translation.

It is noteworthy that, for one-to-one address translation, the discovery of global addresses by hosts could be simplified compared to existing methods. Existing discovery methods were designed not only to discover global addresses, but also to initialize disambiguation state in address translators. Since one-to-one address translation is stateless, the latter functionality can be eliminated for the benefit of simplicity. For example, a simplified method for hosts to discover their global addresses is for access

routers to announce address mapping rules, based on which hosts derive their global addresses given their local addresses. In the case where addresses are translated by swapping their prefix, a mapping rule could be as simple as a pair of local and global address prefixes. This discovery method would be similar to the existing practice of auto-configuring addresses based on on-link address prefixes announced by access routers.

For hosts and authoritative DNS servers to refer peers to addresses they can reach, three approaches are possible:

- Pre-selection: Address referrals include either only local addresses or only global addresses of a host, depending on the location of the peer. This approach is also known as "split horizon".

- Post-selection: Address referrals always include both global and local addresses, independent of the location of the peer. It is left to destination address selection mechanisms in peers to find an address that is reachable from a peer's location.

- Fixed: Address referrals always include only global addresses, independent of the location of the peer.

The suitability of each of these approaches depends on the deployment scenario: In deployments where topology concealment is desired, address referrals can only be pre-selected or fixed, because referrals with local addresses could reveal information about the topology to be concealed. Address referrals must also be pre-selected or fixed where peers may be unable to select the right address among the addresses they have been referred to. While suitable destination address selection mechanisms are standard [REF] in IP version 6, they cannot be expected in IP version 4.

An advantageous of pre-selected address referrals over fixed address referrals is that the former always provide an address that the peer can reach via a shortest path, while the latter may cause a connection to traverse an address translator unnecessarily. On the other hand, a disadvantage of pre-selected address referrals is that they require knowledge of a peer's location. If this is unavailable, address referrals must be fixed where topology concealment is desired, or where peers may not support destination address selection appropriately.

Post-selected address referrals are the most robust approach in IP version 6 when topology concealment is not necessary. They provide peers with complete address information that remains valid even when being recursively referred to between peers, or when being carried by

a mobile peer between the scopes of the local and global addresses. For the same reason, post-selected address referrals are used in upcoming standards for multi-homing [REF] in IP version 6. From the perspective of a peer, it is invisible whether one of the addresses is the translation of the other, or whether the purpose of the addresses is to locate different interfaces on a host.

4.2. Many-to-One Address Translation

Given the foregoing analysis of the Internet-architectural impacts of one-to-one address translation, and given that many-to-one address translation differs from one-to-one address translation only in the extra use of address overloading, the impact of many-to-one address translation on Internet architecture can be evaluated based on solely the implications of address overloading. Those are twofold:

o Ambiguous addresses: Overloading renders a global address ambiguous with respect to the host that is reachable through it because it maps a single global address onto the local addresses of multiple hosts.

o Connection-specific forwarding: Address overloading requires forwarding that is connection-specific so that the recipient host of packets destined to an overloaded address can be disambiguated.

These implications are in conflict with fundamental Internet-architectural principles [REF], which mandate that global address are both unambiguous, and sufficient for connection-agnostic forwarding. The conflict leads to the following two problems of many-to-one address translation, with may have a harmful impact on Internet architecture as the subsequent analysis shows.

o No support for certain connection types: Many-to-one address translation relies on the availability and modifiability of port numbers to identify the connection of a packet. Packets without port numbers are therefore dropped, and so are packets with port numbers that are not modifiable due to encryption and authentication. Furthermore, new connections must be initiated in a way that permits the establishment of disambiguation state. Other connection initiation procedures are not possible, such as the immediate transmission of packets to a global address.

o Reduced network robustness: Address translators constitute a single point of failure for two reasons: First, since they maintain disambiguation state that connections depend upon, they limit a network's ability to reroute traffic in the event of failures. Second, address translators may deliberately dispose of disambiguation state after temporary absence of packets in a

connection, which may make it impossible to resume the connection afterwards.

Methods to mitigate these problems are either not always applicable, or they are complex and hence a source of capital and operational cost. Limited applicability is a shortfall of methods that attempt to enable support for a wider set of connection types in many-to-one address translation. One method [REF] enables many-to-one address translation for connections without accessible port numbers by tunneling packets in an extra layer of UDP. The additional UDP header in packets then provides the port numbers that address translators need to identify the connection of the packets. Unfortunately, UDP tunneling in general cannot enable all connections without accessible port numbers because it requires support at both ends of a connection: Both the host which address is being translated, and the peer are involved, independent of whether the address of the peer is translated as well. Where bilateral support is not provided, connections without accessible port numbers cannot pass through many-to-one address translation.

Another method [REF] to increase the applicability of many-to-one address translation enables hosts with overloaded addresses to receive incoming connections initiated by a peer. Without such method, new connections must be initiated by the host which address is overloaded. Only then does the first packet include the local address to be translated, which can be memorized by the address translator -- in conjunction with the port numbers from the packet and a mapped global address -- to enable subsequent disambiguation of packets received at the mapped global address. To enable connections initiated by a peer, the DNS is used as a point of rendezvous between a host and its peer [REF]. A DNS query by the peer for the host's DNS name is then interpreted as a desire to communicate with the host, and it triggers the establishment of disambiguation state in an address translator. Unfortunately, also this method is of limited applicability, as it takes affect only in those cases where the peer actually performs a DNS lookup prior to connection establishment. This may not always be the case, such as when the peer was previously referred to, or pre-configured with, the address of the host that it wants to reach.

A common method to increase the robustness of networks with address translators is to set up the address translators redundantly. This enables failover of connections from one path to another. While redundant provisioning is straightforward for one-to-one address translation, it is complex, and therefore expensive to deploy and maintain, for many-to-one address translation. Address translators that do not overload addresses function without disambiguation state and can hence substitute for each other without prior

synchronization. The cost of redundancy is then directly proportional to the number of address translators deployed. However, address translators that perform address overloading do maintain disambiguation state, and must hence be continuously synchronized with backup address translators. This increases the cost of redundancy to being proportional to the square of the number of address translators deployed. Increasing the robustness of networks with many-to-one address translators is therefore expensive, and the complexity involved constitutes a potential failure source on its own.

5. Conclusion

This document has shown that the harmful implications of address translation are formost due to the overloading of multiple local addresses onto a single global address. Such many-to-one address translation, which is pursued in the IP version 4 Internet to conserve global addresses, is problematic: It fails to support all connection types, it reduces network robustness because it constitutes a single point of failure, and it requires extra host functionality to support address referrals in applications. One-to-one address translation, which does not overload addresses, would be less problematic: It works for all connection types, and it does not constitute a single point of failure. Furthermore, although one-to-one address translation continues to require extra host functionality to support address referrals, the host functionality can be simplified compared to what is necessary in many-to-one address translation.

The superiority of one-to-one address translation over many-to-one address translation naturally leads to the question whether the former can satisfy the demand for address translation, as it has become apparent through the wide deployment of address translators in today's Internet. Clearly, this depends on the availability of global addresses. One-to-one address translation, which requires one global address per local address, is unsuitable for IP version 4, where the shortage of global addresses necessitates the use of address overloading. For IP version 6, however, one-to-one address translation is suitable, as the sufficient number of global addresses here makes address overloading dispensable. Address translation in IP version 6 could hence, if designed without address overloading, be considerably less harmful than address translation in IP version 4.

Appendix A. Acknowledgment

The author would like to thank James Kempf, Tony Li, David Sinicrope,

Suresh Krishnan, and Zoltan Turanyi for their reviews of this document and their valuable comments.

This document was generated using the xml2rfc tool.

Author's Address

Christian Vogt
Ericsson Research
200 Holger Way
San Jose, CA 95134
United States

Email: christian.vogt@ericsson.com