NSIS
Internet-Draft
Expires: April 27, 2006

A. Pashalidis
H. Tschofenig
Siemens

October 24, 2005

NAT Traversal for GIST
draft-pashalidis-nsis-gimps-nattraversal-01.txt

Abstract

This document describes how different types of Network Address Translator (NAT) interact with the General Internet Signalling Transport (GIST) protocol. The purpose of this interaction is for signalling traffic to traverse the NATs in a way that preserves its semantics with respect to the data flows it corresponds to.

Table of Contents

1. Introduction

Network Address Translators (NATs) modify certain fields in the IP header of the packets that traverse them. In the context of signalling as specified by the General Internet Signalling Transport (GIST) protocol [1], this behaviour may lead to the installation of state at network nodes that may be inconsistent and meaningless with respect to the actual traffic that traverses these nodes.

This document describes how GIST signalling messages traverse NATs in a way that preserves the consistency of state that is installed in the network with respect to the data flows to which the signalling messages refer. As the mechanisms that are described in this document exclusively operate at the GIST layer, they are transparent to signalling applications. The document is organised as follows. The next section introduces the terminology that is used throughout this document. Section 3 provides a detailed discussion of the NAT traversal problem and highlights certain design decisions that have to be taken when addressing the problem. Section 4 lists the assumptions on which the proposed mechanisms are based. Section 5 presents the proposed mechanisms for "transparent" NAT traversal, and ??? presents the proposed mechanisms where such protection is required.

2. Terminology

The terminology, abbreviations and notational conventions that are used throughout the document are as follows.

- DR: Data Responder, as defined in [1]
- DS: Data Sender, as defined in [1]
- GaNAT: GIST-aware NAT - a GaNAT MAY implement a number of NSLPs.
- GIST: General Internet Messaging Protocol for Signalling [1]
- NAT: Network Address Translator
- NI: NSIS Initiator, as defined in [1]
- NR: NSIS Responder, as defined in [1]
- NSIS: Next Steps in Signalling: The name of the IETF working group that specified the family of signalling protocols of which this specification is also a member. The term NSIS is also used to refer to this family of signalling protocols as a whole.
- GIST-aware: Implements GIST and MAY also implement a number of NSLPs.
- GIST-unaware: GIST-unaware, does not implement any NSLP.

- NSLP: NSIS Signalling Layer Protocol, as defined in [1]
- downstream: as defined in [1]
- upstream: as defined in [1]
- MRI: Message Routing Information, as defined in [1]
- NLI.IA: Interface Address field of the Network Layer Information header field, as defined in [1]
- <- : Assignment operator. The quantity to the right of the operator is assigned to the variable to its left.
- A.B: Element B of structure A. Example: [IP header].SourceIPAddress denotes the source IP address of an IP header.

- [data item]: This notation indicates that "data item" is a single identifier of a data structure. (Square brackets do not denote optional arguments in this document.)

3. Problem Statement

According to [1], all GIST messages between two peers carry IP addresses in order to define the data flow to which the signalling refers. Moreover, certain GIST messages also carry the IP address of the sending peer, in order to enable the receiving peer to address subsequent traffic to the sender. Packets that cross an addressing boundary, say from addressing space S1 to S2, have the IP addresses in the IP header translated from space S1 to S2 by the NAT; if GIST payloads are not translated in a consistent manner, the MRI in a GIST packet that crosses the boundary, e.g. from address space S1 to S2, refers to a flow that does not exist in S2. In fact, the flow may be invalid in S2 because at the IP address that belongs to S1 may not be routable or invalid in S2. Moreover, the IP address of the sending peer may also be not routable or invalid in the addressing space of

the receiving peer. The purpose of this document is to describe a way for GIST messages to be translated in a way that is consistent with the translation that NATs apply to the IP headers of both signalling and data traffic.

A NAT may either be GIST-unaware or GIST-aware. We denote a GIST-aware NAT as a GaNAT in the sequel. A GaNAT MAY also support at least one NSLP. Note that there exists an NSLP, namely the NATFW NSLP [2], that specifically addresses NAT traversal for data flows. Inevitably, the NATFW NSLP also provides the necessary mechanisms for the related signalling to traverse the NATs involved. Consider a GaNAT that supports both the NATFW NSLP, and the NAT traversal extension to GIST that is specified in this document. Suppose now that a GIST QUERY message arrives at this GaNAT that contains the NSLP identifier (NSLPID) of the NATFW NSLP. A question that arises is whether the GaNAT should use the GIST NAT traversal extension defined in this document, or the NATFW NSLP mechanism, in order to provide "NAT traversal" for both the signalling message and the data flow to which it refers. The answer to this question is that such a GaNAT MUST implement a policy according to which method is used in preference to the other. Note, however, that, should the GaNAT prefer the GIST NAT traversal to NATFW NSLP, then it will happen, if no on-path GaNATs exist that prefer the NATFW NSLP, that the downsteam NATFW peer will be unable to discover any downstream NATFW NSLP peers. This may make the entire NATFW session obsolete.

Clearly, if a GaNAT does not implement the NATFW NSLP, then the only way for the signalling and the data flow to traverse that GaNAT, is for the necessary mechanisms to operate on the GIST layer. The same holds when the NSLP that is being signalled is an NSLP other than the NATFW NSLP. Enabling NAT traversal in presicely these circumstances is the motivation of this specification.

In general, a given data flow between a data sender (DS) and a data receiver (DR) may have to traverse a number of NATs, some of which may be GIST-and-NATFW-aware, some may be GIST-aware, and some may be GIST-unaware. Additionally, NSLP signalling for such a data flow may be required to traverse through a subset of those NATs. Whether or not the routing inftrastructure and state of the network causes the signalling for such a data flow to traverse the same NATs as the flow depends, among other things, on which NSLP is being signalled. While

signalling of the QoS NSLP, for example, might not traverse any of the NATs that are traversed by the data flow, the signalling of the NATFW NSLP traverses at least those NATs that implement the NATFW NSLP (otherwise the signalling path would no longer be coupled to the data path, as this coupling is defined by the GIST QUERY/RESPONSE discovery mechanism for the "path coupled" Message Routing Method). It is desirable that the NAT traversal extension for GIST provides NAT traversal for every possible combination of NATs, either on the data or the signalling path, in a secure manner.

Due to the GIST QUERY/RESPONSE discovery mechanism (according to which QUERY messages are simply forwarded if the current node does not support the required NSLP), two GIST nodes typically identify themselves as NSLP peers only if they both implement the same NSLP. This means that, if one or more NATs that are unaware of this NSLP are between them, then the two NSLP peers are not able to discover each other at all. This is because, even in the unlikely event that the NAT bindings that are necessary for the GIST traffic to traverse the in-between NAT(s) exist, the NLI.IA field included in the RESPONSE message sent by the downstream GIST peer is invalid (or the IP address is unreachable) in the address space of the upstream GIST peer. In order to overcome this limitation, either the two peers need to cope with the in-between NAT(s), or, if the NAT(s) are GaNATs, they (the GaNATs) need to apply additional processing in order to transparently create and maintain consistency between the information in the header of GIST signalling messages and the information in the IP header of the data traffic. Additionally, if NSLP-aware NATs are on the data path, then these NATs should process NSLP traffic in a way that preserves consistency after address translation. This processing deviates from the processing of NSLP-aware non-NAT nodes. In the following sections we specify mechanisms that aim to overcome the limitation of two adjacent GIST peers not being able to execute an NSLP in the presence of in-between NAT(s).

Note that a number of different situations has to be dealt with, depending on the level of NSIS support by a NAT. The following combinations of NATs that are located between two adjacent GIST peers are considered.

- all NAT(s) are GIST-unaware

- all NAT(s) are (NSLP-unaware) GaNAT(s)

- all NAT(s) are NSLP-aware

- a combination of the above NAT types.

The approach taken in this document is to propose separate mechanisms for the traversal of each of the above type of NAT. If NATs that belong to multiple types exist on the path between two adjacent NSLP peers, the proposed mechanisms should work in combination. Thus, traversal of multiple NATs of different types should not require further specification from a functional perspective. However, security issues that arise due to the combination of NAT types may have to be considered.

A GIST-unaware NAT cannot tell data and signalling traffic apart. The installation of the NAT binding for the signalling traffic in such a NAT occurs typically independently from the installation of the NAT binding for the data traffic. Furthermore, as the NAT cannot associate the signalling and the data traffic in any way, it cannot indicate that an association exists between the two NAT bindings. Therefore, in the presence of such a NAT, GIST nodes that are located on either side of the NAT have to cope with the NAT without assistance from the NAT. This would typically require initially discovering the NAT and subsequently establishing an association between between the MRI in the signalling messages and the translated IP header in the data traffic. Due to the variety of behaviours that a GIST-unaware NAT may exhibit, establishing this association is a non-trivial task. Therefore, traversal of such (i.e. GIST-unaware) NATs is considered a special case and is further discussed in Section 7.

Traversal of GaNAT(s) is comperatively more straightforward. This is because, based on the MRI in a given incoming GIST message, a GaNAT can identify the data flow to which the message refers. It can then check its NAT binding cache and determine the translation that is (or, if no NAT binding for the flow exists yet, will be) applied to the IP header of the data flow. The GaNAT can then include sufficient information about this translation into the signalling message, such that its receiver (i.e. the GIST peer that receives the data traffic after network address translation has been applied) can map the signalling message to the data flow.

There exist a variety of ways for a GaNAT to encode the abovementioned information into signalling messages. In this document the following two ways are considered.

1. Non-transparent apprach: The GaNAT includes an additional "NAT Traversal" payload (see section A.3.8 of [1]) into the GIST header of the GIST QUERY message. This "NAT Traversal" payload is echoed by the GIST responder on the other side of the NAT. Both peers use the information in this payload in order to map subsequent signalling messages to the data flows they refer to.

2. Transparent approach: The GaNAT replaces GIST header fields in a way that is consistent with the translation it applies to the data traffic, as necessary. The GaNAT does this GIST QUERY and RESPONSE messages, for D-mode as well as for C-mode messages throughout the duration of the signalling session.

The second approach being "transparent" means that a GaNAT that follows this approach remains completely transparent to the GIST peers that are located either side of it. Thus, this approach works even if these GIST peers do not support the NAT traversal object for GIST (as described in section 7.2 of [1]). Unfortunately though, the transparent approach does not work if the GaNAT is NSLP-unaware and if signalling traffic is to be cryptographically protected between the two GIST peers that are located either side of the GaNAT. If, however, the GaNAT is NSLP-aware, then cryptographic protection is terminated at the GaNAT (i.e. the GaNAT is a GIST peer itself). In this scenario, it is clearly preferable for the GaNAT to follow the transparent approach, rather than to include a NAT Traversal object. Thus, if a GaNAT acts as a GIST peer for a signalling session, it MUST follow the transparent approach, as described in Section 5.3. However, due to the fact that the transparent approach does not work if signalling is to be cryptographically protected, a GaNAT MUST also implement the non-transparent approach (for the case where an NSLP is signalled that the GaNAT does not support), unless the GaNAT is going to be used only in deployments where cryptographic protection of signalling traffic is not a requirement.

Note that a GaNAT MAY implement both approaches. If such a GaNAT is NSLP-unaware, it can then adopt the correct behaviour, based on whether or not cryptographic protection is required for the signalling traffic between two GIST peers. If such protection is required, the GaNAT MUST adopt the mechanisms that follow the non-transparent approach; if it is not, it MAY follow the mechanisms implementing the transparent approach. The GaNAT can tell whether or not cryptographic protection is required from the stack proposals that are exchanged in the GIST QUERY and RESPONSE messages; inclusion of IPsec or TLS proposals amounts to cryptographic protection being required.

Regardless of which approach is taken, after a GIST response passes through an NSLP-unaware GaNAT, the GaNAT should expect a messaging

association to be set up between the two involved GIST peers. For this to happen, the GaNAT has to allow traffic to traverse it. From a security perspective, the GaNAT should allow the minimum necessary traffic types to traverse. Additionally, the amount of per signalling session state that is allocated at a GaNAT should be minimised for reasons of efficiency and resistance to service denial attacks. For these reasons, it makes sense for the GaNAT to be able to predict with certainty which of the GIST responder's stack proposal will be used by the initiator for the establishement of a messaging association. This can be accomplished by having the GaNAT modify the stack configuration object in the GIST RESPONSE as it passes though it such that the initiator is forced to use a particular stack proposal and, if applicable, a particular transport layer destination port.

The reason why it is preferable for the GaNAT to modify only the stack configuration data object (as opposed to the stack proposal object) is that the GIST responder expects its original stack proposal to be included in the GIST CONFIRM message. The initiator would not be able to echo that object as it would not have seen it and, if signalling messages are cryptographically protected, then the GaNAT cannot replace the stack proposal in the GIST CONFIRM message with the one the responder expects, without causing cryptographic checks to fail at the responder. Thus, the approach taken in this document is for the GaNAT to render invalid all but one stack configuration data objects in the GIST RESPONSE message. How this is done depends on the format of this object. If, for example, it contains transport layer port numbers, it is rendered invalid by setting these numbers to zero.

A question arises due to the fact that the stack proposal carried by a GIST QUERY may include multiple proposals of which some provide cryptographic protection for signalling traffic and some do not. Which behaviour should a GaNAT that supports both approaches assume in this case? In order to provide maximise the probability that cryptographic protection is going to used, the GaNAT MUST follow the non-transparent approach in this case.

4. Assumptions

The discussion in this document is based on the following assumptions.

1. No IP addresses and port numbers are carried in the payloads of an NSLP.

2. The path taken by the signalling traffic between those GIST peers that have GaNATs in between is such that the responses to packets that a GaNAT sends on given interface arrive on the same interface (if such responses are sent at all).

3. The path taken by signalling traffic remains fixed between the two GIST peers, as far as the in-between GaNATs are concerned. That is, we assume that signalling traffic traverses the same GaNAT(s) until at least one of the following conditions is met.

   * The NSIS state that is installed at the two GIST peers expires.

   * The NSIS state that is installed at the two GIST peers is refreshed using a GIST QUERY.

   * A new GIST QUERY/RESPONSE exchange takes place due to other reasons, e.g. a detected route change.

   Note that this assumption is not necessarily met by "normal" data path coupled signalling. This is because, under "normal" data path coupled signalling, the signalling traffic is "coupled" to the data traffic at nodes that decide to act as GIST peers. Thus, under "normal" path coupled signalling, it is not an error

condition (e.g. a reason to trigeer a "route change"), for example, if the set of on-path nodes, which do not act as GIST peers, changes, as long as adjacent GIST peers remain the same.

4. The data flow traverses the same set of GaNATs as the signalling traffic. By assumption 3, this set of GaNATs is fixed until the next GIST QUERY/RESPONSE procedure is executed.

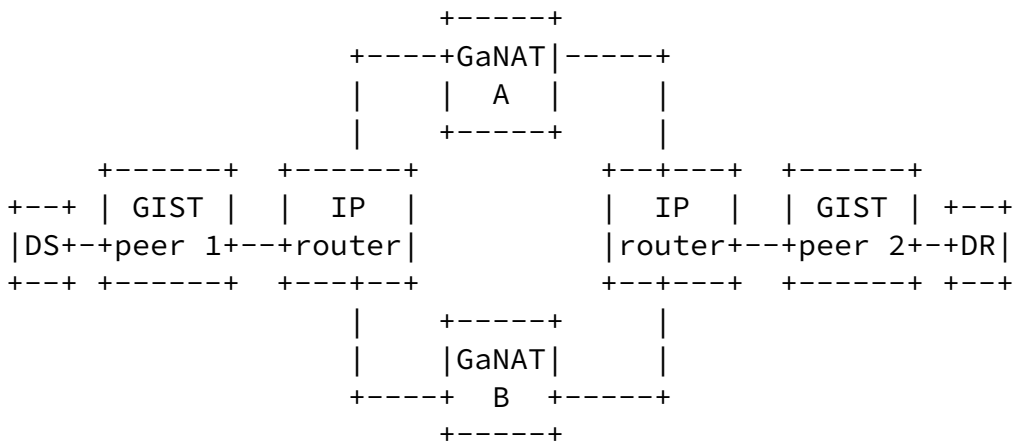

Figure 1: Network with more than one NAT at an addressing boundary

Figure 1 illustrates the importance of assumptions (3) and (4). With regard to that figure, suppose that a (D-mode) signalling session has been setup between the two adjacent GIST peers 1 and 2 and that both signalling and data traffic follows the path GIST peer 1 -> IP router -> GaNAT A -> IP router -> GIST peer 2. Suppose now that, after some time, GIST peer 1 decides to set up a C-mode connection with peer 2. Suppose moreover that the left IP router decides to forward the C-mode signalling traffic on the link towards GaNAT B. Thus, signalling traffic now follows the alternative path GIST peer 1 -> IP router -> GaNAT B -> IP router -> GIST peer 2. Note that this change

in forwarding between the two adjacent GIST peers does not trigger a "route change" at the GIST layer because (a) it does not necessarily destroy the adjacency of peer 1 and 2 and (b) it does not necassarily destroy the coupling of the path taken by signalling traffic to that taken by data traffic (at GIST nodes). Nevertheless, assumptions (3) and (4) mandate that this situation does not occur. However, even if such a situation occurs, the mechanisms described in this document may still work as state expires after a certain timeout period.

If assumption (1) does not hold, the NSLP has to provide additional mechanisms for the traversal of (Ga)NATs. These mechanisms must be compatible with the mechanisms employed by GIST. Assumptions (2), (3) and (4) hold if, at an addressing boundary, only one NAT exists. Due to security and management reasons, this is likely to be the case in many settings.

5. Transparent NAT traversal for GIST

This section describes the operation of GaNATs that implement the transparent approach listed in Section 3. Note that the GaNAT may follow this approach only if it is unaware of the NSLP that is being signalled, and when no cryptographic protection of signalling data is requested by the two NSLP peers.

Note that we have to deal with two types of GaNATs, namely those that are located at the NSIS initiator (NI-side), and those that are located at the NSIS responder (NR-side). This distinction arises due to the fact that NI-side and NR-side GaNATs obtain the destination IP address for forwarded packets in different ways.

5.1. NI-side NSLP-unaware GaNATs

This section describes the "transparent" operation of an NI-side, NSLP-unaware GaNAT.

For every arriving IP packet P, an NSLP-unaware, NI-side GaNAT executes the following algorithm.

1. If P has a RAO followed by the GIST header with an NSLP ID that is not supported, it is identified as a GIST QUERY. In this case the GaNAT performs the following.

    1. We denote P as GQ. It looks at the stack proposal ST in GQ. If it indicates that cryptographic protection is required, the mechanism that is applies is the one described in ???.

    2. The GaNAT remembers GQ along with the interface on which it arrived. We call this interface the "upstream link".

    3. It searches its table of existing NAT bindings against entries that match the GQ.MRI. A matching entry means that the data flow, to which the signalling refers, already exists.

        1. If a matching entry is found, the GaNAT looks at which link the packets of the data flow are forwarded; we call this link the "downstream" link. Further, the GaNAT checks how the headers of the data flow (IP addresses, port numbers, etc) are translated according to this NAT binding. We denote [IP header].SourceIPAddress used on the downstream link as IPGaNATds, and the source port number used to forward the data traffic as SPNDTGaNATds. The NAT may also use a different source port number when forwarding signalling traffic. This port number is

denoted as SPNSTGaNATds.

2. If no matching entry is found, the GaNAT determines, based on its routing table, the link on which packets that match GQ.MRI (excluding GQ.MRI.SourceIPAddress) would be forwarded. We call this link the "downstream link". Then, the GaNAT acquires an IP address for itself on the downstream link. (This address could be dynamic or static.) This address will be used to forward both signalling and data traffic on the downstream link. If it also performs port translation, the GaNAT also

acquires a source port number for the data traffic on the downstream link. This will be used with the NAT binding, if such a binding will be established for the data traffic at a later stage, and is denoted as SPNDTGaNATds. The signalling traffic packets may also be forwarded using the a different source port number as the incoming packets. We denote the acquired IP address as IPGaNATds and the source port number for the signalling traffic as SPNSTGaNATds.

Issues: The reason why the GaNAT may also assign a different source port number to the signalling traffic, is to enable the GaNAT to demultiplex (i.e. forward to the correct internal address) the signalling responses that arrive from downstream. Of course, a GaNAT does not need to actually change the source port of signalling traffic; it can always use SPNSTGaNATds the same port as in the incoming packet. Such a GaNAT may use the GIST session ID in order to demultiplex the traffic that arrives from the downstream direction. It is unclear which of the two approaches is preferable.

4. It creates a new GIST QUERY packet GQ', as follows.

    1. GQ' <- GQ

    2. GQ'.MRI.SourceIPAddress <- IPGaNATds

    3. GQ'.MRI.SourcePortNumber <- SPNDTGaNATds

    4. GQ'.[IP header].SourceIPAddress <- IPGaNATds

    5. GQ'.[UDP HEADER].SourcePort <- SPNSTGaNATds

    6. GQ'.NLI.IA <- IPGaNATds

    7. GQ'.S <- true

5. It remembers GQ and GQ', the fact that they are associated, and the associated upstream and downstream links. (Note: The

GaNAT does not have to remember the entire packets; for simplicity of exposition, however, we assume it does. An implementation SHOULD discard at this point all information that is not used later.)

6. It forwards GQ' on the downstream link.

2. Otherwise, if P carries an [IP header].DestinationIPAddress that belongs to the GaNAT, and if it is identified as a GIST response in D-mode with an NSLP ID that is not supported, the GaNAT does the following (P is denoted as GR).

   1. It searches for a matching GQ' in its buffer. A GR is said to match a GQ' if they carry the same cookie value. If none is found, GR is discarded. Otherwise, the GaNAT may also perform further consistency checks on a matching GR/GQ' pair, such as checking that they contain the same session IDs, MRIs, NSLP IDs. If consistency checks succeed, the GaNAT constructs a new GIST response GR', as follows.

      1. GR' <- GR

      2. GR'.MRI <- GQ.MRI, where GQ is the packet associated with GQ'(as remembered previously), and GQ' is the packet that matches the received GR.

      3. GR'.[IP header].SourceIPAddress <- IPGaNATus, where IPGaNATus = GQ.[IP header].DestinationIPAddress.

      4. GR'.[IP header].DestinationIPAddress <- GQ.NLI.IA

      5. GP'.S <- true.

      6. It inspects the stack proposals in GR' and the corresponding GQ' to see if the upstream GIST peer has a choice of more than one possible stack. If such a choice exists, the GaNAT removes as many stack proposals from GR' as necessary, until only one stack can be chosen by the upstream peer for the messaging association. We denote this stack as ST. The GaNAT remembers this ST and its association with GQ, GQ', GR, GR'. We say that, in this case, the GaNAT "installs" the ST.

2. It forwards GR' on the upstream link.

3. If no NAT binding for the data traffic was found in step 1.3.2, the GaNAT now installs a NAT binding (for the unidirectional data traffic) which says that "a packet K that arrives on the upstream link and for which it holds that

   + K.[IP header].DestinationIPAddress=GQ.MRI.DestinationIPAddress,

   + K.[IP header].Protocol=GQ.MRI.Protocol, and

   + K.[TCP/UDP header].SourcePort=GQ.MRI.SourcePort

   should be forwarded on the downstream link, with [IP header].SourceIPAddress = IPGaNATds.

   Issues: there is a question of whether this NAT binding should also enable data traffic in the opposite direction to traverse the NAT; in order to be able to demultiplex upstream traffic that carries data that belongs to different flows, the GaNAT should keep the necessary per-flow state. From a signalling point of view, however, upstream data traffic that corresponds (on the application level) to the downstream flow to which this GIST session refers, is a separate flow for which, depending on the application, there may or there may not exist a signalling session. If such a signalling session exists, then the GaNAT acts as an NR-side GaNAT for this session. Thus, during the processing of this signalling care has to be taken not to establish a NAT binding for a flow for which a NAT binding already exists. Finally, security issues arise when traffic, for which no signalling exists, is allowed to traverse a GaNAT.

   Another issue is about refreshing the NAT binding. A NAT binding that was established as a result of GIST signalling should remain in place for as long as the associated GIST state in the GaNAT remains valid. If GIST signalling refers to a NAT binding that already exists, then the timeout of the NAT binding should occur according to the NAT policy, in a manner independent from GIST processing. (If signalling persists after the deletion of a NAT binding, then the NAT binding may be re-installed and then timed out together with GIST state).

3. Otherwise, if P.[IP header].DestinationIPAddress belongs to the GaNAT, and if P carries the transport protocol and port numbers

indicated by some stack proposal ST that has previously been

installed by the GaNAT, and if P has arrived on either the upstream or the downstream interface that is associated with ST, then P is said to "match" ST. For such a packet, the GaNAT does the following. If P is expected to contain a GIST header, then the GaNAT check whether or not the bits where the GIST header is expected, consitute a valid GIST header. If not, P is silently discarded. Otherwise, the GaNAT constructs an outgoing packet P' as follows (the variables used below refer to those stored together with ST).

1. P' <- P
2. If P has arrived on the upstream link, then
    1. P'.[IP header].SourceIPAddress <- IPGaNATds
    2. P'.MRI <- GQ'.MRI
    3. P'.NLI.IA <- IPGaNATus
    4. The GaNAT forwards P' on the downstream link.
3. else (if P has arrived on the downstream link)
    1. P'.[IP header].SourceIPAddress <- IPGaNATus
    2. P'.MRI <- GQ.MRI
    3. P'.NLI.IA <- IPGaNATus
    4. The GaNAT forwards P' on the upstream link.

Note that the GaNAT can determine the location in a packet where a GIST header is expected. If, for example, the packet is a UDP packet, then the GIST header should follow immediately after the UDP header. If the packet is a TCP packet, then the GaNAT can determine the location where the GIST heaer should start by counting the number of NSLP payload bits that followed the end of the previous GIST

header. The start of the next GIST header is expected at the position where the previous GIST message, including NSLP payload, ends. The GaNAT can tell where this message ends from the LENGTH field inside the previous GIST header. It should be noted here that, in order to correctly count the bits, the GaNAT may have to keep track of TCP sequence numbers, and thereby be aware of the correct ordering of packets. However, the GaNAT only has to keep buffers that

are as long as the LENGTH field inside the previous GIST header (and possibly up to one MTU size more than that).

Also note that some TCP packets P may not be expected to contain any GIST header (this happens when the NSLP payload from a previous packet stretches over several packets). For those packets, the GaNAT only applies the transformation in the IP header. Finally, note that a GIST header may start a packet but finish in another. If such a packet is received by the GaNAT, it MUST buffer the entire packet, until the packet is received where the GIST header completes. It can then apply the required processing and forward both packets.

4. Otherwise, if P matches a (data) NAT binding, the GaNAT applies normal NAT processing and forwards the packet on the corresponding link.

5. Otherwise, P is silently discarded.

Brief discussion of the algorithm: The fact that the GaNAT replaces the NSLP peer's NLI.IA with its own IP address (in both directions), causes the two GIST peers to send subsequent signalling messages to the GaNAT, in the belief that they talk to the their adjacent peer. The GaNAT transparently forwards the signalling traffic and appropriately translates the fields in the GIST header, in a way that is consistent with the translation it applies to the data traffic.

Note that, according to this mechanism, the size of an outgoing GIST message is always the same as the size of its corresponding incoming GIST message. Finally note that the MRI that the NR sees indicates as destination address the IP address of the DR (as expected), but as source address it indicates the is IPGaNATds of the GaNAT that is closest to the NR.

5.2. NR-side NSLP-unaware GaNATs

The case of NR-side GaNATs is more subtle, since, in this setting, the DS does not learn the IP address of the DR (which is assumed to be on the same side of the GaNATs as the NR) and the NI does not learn the address of the NR. In this setting we assume that each NR-side GaNAT that is in between two GIST peers, a priori knows a routable IP address of the downstream GaNAT. The last GaNAT of this chain is assumed to know the IP address of the DR. In order to clarify this assumption, see, for example, Figure 2. In this figure, GaNAT A is assumed to know the IP address of GaNAT B, GaNAT B is assumed to know the IP address of GaNAT C, and GaNAT C is assumed to know the IP address of the DR. A given GaNAT that knows such an address, in effect anticipates to receive a signalling message from

the upstream direction that refers to a data flow that terminates in a downstream node. In other words, such a GaNAT may typically have already a NAT binding in place for the data traffic. We call the IP address of the next downstream GaNAT (or, if the GaNAT is the last in the chain, the address of the DR) the "pending" IP address. In the following description it is denoted by IPNext. How IPNext is made known to each GaNAT (e.g. how the NAT binding for the data traffic is installed in the GaNAT) is outside the scope of this document.

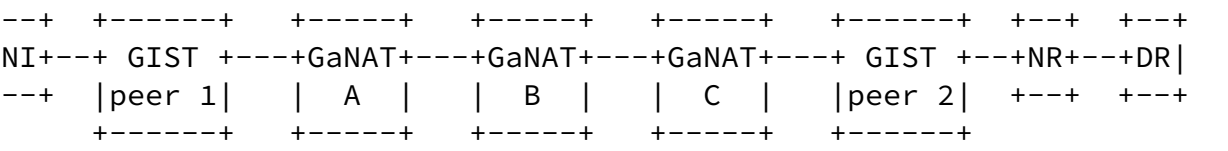

Figure 2: Network with NR-side GaNATs (the public Internet is assumed to be between NI and GIST peer 1)

For every arriving IP packet P, an NSLP-unaware, NR-side GaNAT executes the following algorithm.

1. If P has a RAO followed by the GIST header with the NSLP ID indicates an unsupported NSLP, it is identified as a GIST QUERY. In this case the GaNAT does the following.

1. We denote P as GQ. The GaNAT looks at the stack proposal ST in GQ. If it indicates that cryptographic protection is required, the algorithm that is executed is the one described in section Section 6 below.

2. The GaNAT remembers GQ along with the link on which it arrived. We call this link the "upstream" link.

3. The GaNAT determines whether or not this GIST QUERY is anticipated, i.e. if a pending IPNext exists that matches this GIST QUERY. A pending IPNext is said to "match" a GIST QUERY, if [this condition is an open issue!] If no pending IPNext is also matching, P is discarded (it is a question whether or not an error message should be sent). Otherwise, additional checks may be performed (e.g. a DSInfo object may have to be checked against the GQ). If these checks fail, P is discarded. Otherwise, the GaNAT performs the following.

4. It searches its table of existing NAT bindings against entries that match the GQ.MRI. A matching entry means that the data flow, to which the signalling refers, already exists.

+ If a matching entry is found, the GaNAT looks at which link the packets of the data flow are forwarded; we call this link the "downstream" link. Further, the GaNAT checks how the headers of the data flow (IP addresses, port numbers, etc) are translated according to this NAT binding. We denote [IP header].SourceIPAddress used on the downstream link as IPGaNATds, and the source port number as SPNDTGaNATds. Note that the [IP header].DestinationIPAddress of this NAT binding should be equal to IPNext. If it is not, this should be handled as an auditive error condition. The GaNAT may also assign a new source port number to signalling traffic, which is denoted as SPNSTGaNATds.

+ If no matching entry is found, the GaNAT determines, based on its routing table, the link on which packets that match GQ.MRI (excluding GQ.MRI.SourceIPAddress and where GQ.MRI.DestinationIPAddress is replaced with IPNext) would be forwarded. We call this link the "downstream" link.

Then, the GaNAT acquires an IP address for itself on the downstream link. (This address could be dynamic or static.) Depending on its type, the GaNAT may also acquire source port numbers for the translation of data traffic. We denote the acquired IP address as IPGaNATds and the source port numbers for data and signalling traffic as SPNDTGaNATds and SPNSTGaNATds respectively.

5. It creates a new GIST QUERY packet GQ', as follows.

    1. GQ' <- GQ

    2. GQ'.MRI.SourceIPAddress <- IPGaNATds

    3. GQ'.MRI.DestinationIPAddress <- IPNext.

    4. GQ'.MRI.SourcePort <- SPNDTGaNATds.

    5. GQ'.[IP header].SourceIPAddress <- IPGaNATds

    6. GQ'.[UDP HEADER].SourcePort <- SPNSTGaNATds

    7. GQ'.[IP header].Destination_IP_Address <- IPNext

    8. GQ'.NLI.IA <- IPGaNATds.

    9. GQ'.S <- true

6. It remembers GQ, GQ' the fact that they are associated, and the associated upstream and downstream links (interfaces).

7. It forwards GQ' on the downstream link.

Steps 2,3, 4 and 5 of the algorithm are analogous to the corresponding steps of the algorithm executed by NSLP-unaware, NI-side GaNATs, which was described in Section 5.1.

5.3. NSLP-aware GaNATs

The difference of NSLP-aware GaNATs and NSLP-unaware GaNATs is that

the former perform NSLP processing in addition to the processing of the NSLP-unaware GaNATs. Another way to see this is by observing that NSLP-aware GaNATs should provide an "MRI translation service" (MRITS) in addition to normal GIST and NSLP processing. The motivation behind the MRITS is for GIST to hide from the NSLP that signalling messages traverse an addressing boundary. In other words, the purpose of the MRITS is to make the NSLP believe that it is operating in a single IP addressing space. When and how the MRITS is invoked for a particular packet depends on (i) the direction of the packet (i.e. downstream or upstream) and (ii) the location of the GaNAT (i.e. NI-side or NR-side). It should also be noted that certain NSLP layer tasks must be carried out in consistency with the placement of the MRITS. This is to prevent events triggered by the NSLP to cause installation of inconsistent state. In order to clarify this, consider the scenario of the QoS NSLP running in a GaNAT that operates according to the mechanisms described in this section. Since the GaNAT only presents a single addressing space to the NSLP (say, the internal addressing space), the packet classifier of the GaNAT's QoS provisioning subsystem should classify packets based on internal addresses only (i.e. it should first translate packets that carry external addresses and then classify them). Whether the MRITS presents internal-only or external-only addresses to the NSLP is not significant, as long as NSLP layer operations are carried out consistently. In the remainder of this section we present the case where internal addresses are presented to the NSLP.

The MRITS is obviously invoked only on GIST packets that carry an NSLP identifier that corresponds to an NSLP that the GaNAT actually supports. Also, for non-GIST packets, normal NAT behaviour applies. Although the MRITS is part of GIST processing, in order to clarify our discussion, we view it as a somewhat separate processing step (i.e. like a subroutine). For NI-side, NSLP-aware GaNATs, it holds that

- for a GIST/NSLP packet that is to be forwarded on the downstream link of an NI-side GaNAT, the MRITS is invoked after the packet has been processed by the NSLP and before it is given to GIST, and
- for a GIST/NSLP packet that is received on the downstream link,

the MRITS is invoked after GIST processing and before the packet is given to the NSLP.

The converse holds for NR-side NSLP-aware GaNATs. In particular,

- for a GIST/NSLP packet that is to be forwarded on the upstream link of an NI-side GaNAT, the MRITS is invoked after the packet has been processed by the NSLP and before it is given to GIST, and
- for a GIST/NSLP packet that is received on the upstream link, the MRITS is invoked after GIST processing and before NSLP processing.

Figure 3 illustrates this idea.

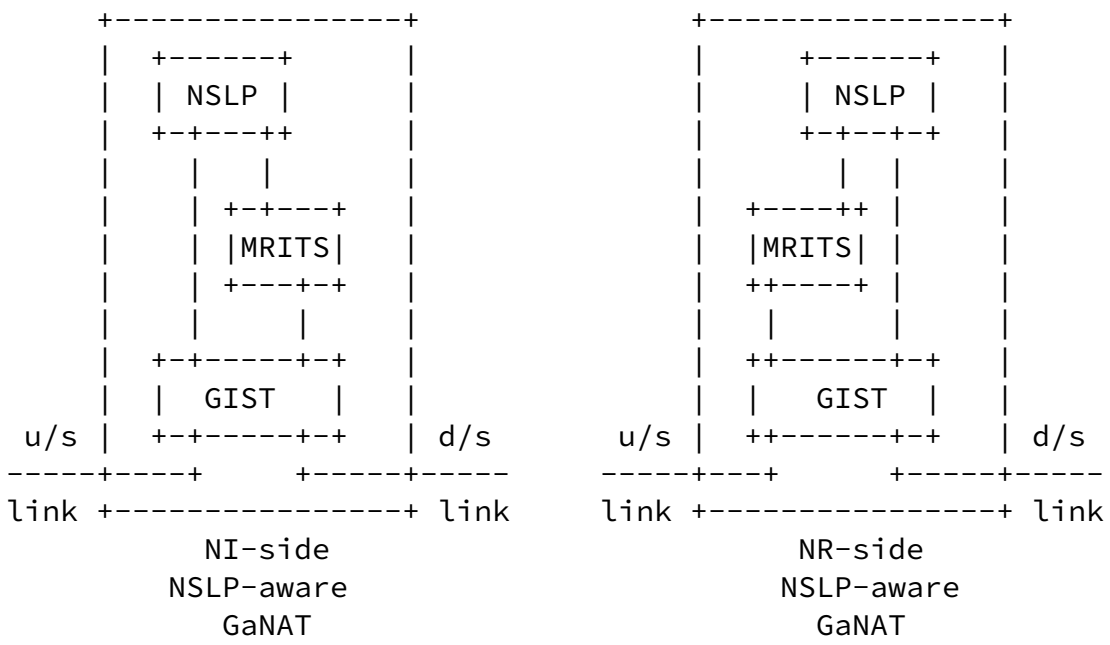

Figure 3: Operation of the MRI Translation Service

The reason for this construction is to give the NSLP the impression that it works only with flows that originate and terminate in the internal address space. We now describe the operation of the MRITS and GIST in NSLP-aware GaNATs. An NI-side NSLP-aware GaNAT operates according to the following rules.

1. When the NSLP asks for a message to be sent towards the downstream GIST peer, the MRITS does the following (IPGaNATds and SPNDTGaNATds are obtained similarly to the case of an NSLP-unaware GaNAT).

    1. MRI.SourceIPAddress <- IPGaNATds

    2. MRI.SourcePort <- SPNDTGaNATds

2. Additionally, GIST performs the following on the resulting packet before it is forwarded on the downstream link (SPNSTGaNATds is obtained similarly to the case of an NSLP-unaware GaNAT).

    1. [IP header].SourceIPAddress <- IPGaNATds

    2. [UDP/TCP header].SourcePort <- SPNSTGaNATds

    3. NLI.IA <- IPGaNATds

    4. S <- true

3. If a message is received on the downstream link, the MRITS does the following before the NSLP is invoked.

    1. MRI.SourceIPAddress <- IPflow

    2. MRI.SourcePort <- SPNDTGaNATus, where IPflow is the IP address of the DS (as seen by the GaNAT) and SPNDTGaNATus is the destination port number used in the original MRI.

4. If, after NSLP processing, a message is to be forwarded on the upstream link, GIST performs the following processing (note that no MRITS processing takes place in this case).

    1. [IP header].SourceIPAddress <- IPGaNATus

    2. [IP header].DestinationIPAddress <- IPpeer

    3. NLI.IA <- IPGaNATus

    4. S <- true, where IPGaNATus is the GaNATs IP address for the upstream link, IPpeer is the IPaddress of the NI (or the next GaNAT in the upstream direction), and IPflow is the IP address of the DS (as seen by the GaNAT). The GaNAT is assumed to determine the correct IPGaNATus and IPpeer from previous communications and in cooperation with GIST. [Issue: how exactly should IPGaNATus, IPpeer and IPflow be resolved; i.e. what exactly should the GaNAT remember?]

An NR-side NSLP-aware GaNAT operates according to the following rules.

1. If the packet is received on the upstream link, the MRITS does the following, before the NSLP is notified.

    1. P.MRI.SourceIPAddress <- IPGaNATds

    2. P.MRI.DestinationIPAddress <- IPNext, where IPGaNATds is the GaNAT's IP address for the downstream link and IPNext is the address of the DR. IPNext is obtained in a way similar to the case of an NSLP-unaware GaNAT.

2. If, after NSLP processing, a message is to be forwarded on the downstream link, GIST performs the following processing (note that no MRITS processing takes place in this case).

    1. [IP header].SourceIPAddress <- IPGaNATds

    2. [IP header].DestinationIPAddress <- IPNext

    3. NLI.IA <- IPGaNATds

    4. S <- true, where IPGaNATds is the GaNATs IP address for the downstream link, IPNext is the IP address of the DR (or the next GaNAT in the downstream direction). The GaNAT is assumed to determine the correct IPNext in a way similar to the case of an NSLP-unaware GaNAT.

3. When the NSLP asks for a message to be sent towards the upstream GIST peer, the MRITS does the following.

    1. MRI.SourceIPAddress <- IPflow

    2. MRI.Destination_IP_Address <- IPGaNATus

4. Additionally, GIST performs the following on the resulting packet before it is forwarded on the downstream link.

    1. [IP header].SourceIPAddress <- IPGaNATus

    2. [IP header].DestinationIPAddress <- IPpeer

3. NLI.IA <- IPGaNATus

4. S <- true, where IPGaNATus is the GaNATs IP address for the upstream link, IPpeer is the IP address of the NI (or the next GaNAT in the upstream direction), and IPflow is the IP

address of the DS. The GaNAT is assumed to determine the correct IPGaNATus and IPpeer fields from previous communications and in cooperation with GIST. [question: how exactly should IPGaNATus and IPpeer be resolved; i.e. what exactly should the GaNAT remember]?

5.4. Combination of NSLP-aware and NSLP-unaware GaNATs

In the absence of an adversary, a combination of NSLP-aware and NSLP-unaware GaNATs should work without further specification. However, in the presence of an adversary, additional security issues may arise from the combination. These issues may introduce opportunities for attack that do not exist in setting where the on-path GaNATs are either all NSLP-aware or all NSLP-unaware.

6. Non-transparent NAT traversal

This section discusses the "non-transparent" operation for GaNAT traversal at the GIST layer, i.e. the first approach listed in Section 3. For this approach, the behaviour of both the GaNAT and the GIST peers is defined. As with the transparent approach, the case of the in-between GaNAT(s) being located at the NI-side is different from that of NR-side GaNATs. Note that the mechanisms in this section apply only to NSLP-unware GaNATs.

The GaNAT informs the NSLP peers about its presence during the GIST discovery process. This information enables the NSLP peers to map the translated data flow to the signalling messages, and to consistently translate the MRI, so that the NSLP only "sees" the correct MRI. Cryptographic protection of signalling messages can be supported with this approach because the GaNAT only modifies the GIST QUERY and REPONSE messages, which are never cryptographically protected in their entirety.

In this approach, the GaNAT embeds a new GIST payload type into the GIST QUERY. This payload encodes the aforementioned information, and we call this payload type the "NAT Traversal Object" (NTO). The NTO is an optional payload in the GIST header of a GIST QUERY, and is added, and processed, by the GaNAT(s) through which the QUERY traverses. The information in the NTO must enable the two NSLP peers to locally translate the MRI in the same way as if it were consistently and transparently translated by the in-between GaNAT(s). Note that there may be more than one GaNAT between the two NSLP

peers. The format of the NTO follows the format of the object in the GIST common header. In particular, the NTO is preceded by a TLV common header, as defined in [1]. The A and B flags are both set to 0 in this header, indicating that support for the NTO is mandatory. The type value is TBD. The NTO is defined as in section A.3.8 of [1].

6.1. NI-side NSLP-unaware GaNATs

For every arriving IP packet P, an NSLP-unaware, NI-side GaNAT executes an algorithm that is equivalent to the following.

1. If P has a RAO followed by the GIST header with an NSLP ID that is not supported, it is identified as a GIST QUERY. In this case the GaNAT does the following.

    1. We denote P as GQ. The GaNAT looks at the stack proposal ST in GQ. If it does not include a proposal with cryptographic protection, the GaNAT MAY choose to follow the approach described in Section 5.1 above.

    2. We call the link on which GQ arrived the "upstream" link.

    3. The GaNAT searches its table of existing NAT bindings against entries that match the GQ.MRI. A matching entry means that the data flow, to which the signalling refers, already exists.

        + If a matching entry is found, the GaNAT looks at which link the packets of the data flow are forwarded; we call this link the "downstream" link. Further, the GaNAT checks how the headers of the data flow (IP addresses and port numbers) are translated according to this NAT binding.

        + If no matching entry is found, the GaNAT determines, based on its routing table, the link on which packets that match GQ.MRI (excluding GQ.MRI.SourceIPAddress) would be forwarded. We call this link the "downstream" link. Then, the GaNAT acquires an IP address and source port for itself on the downstream link. (This address and port could be dynamic or static.)

4. We denote [IP header].SourceIPAddress used on the downstream link as IPGaNATds, and the source port number for the data and signalling traffic as SPNDTGaNATds and SPNSTGaNATds respectively.

5. It creates a new GIST QUERY packet GQ', as follows.

    1. GQ' <- GQ

    2. GQ'.MRI.SourceAddress <- IPGaNATds.

    3. GQ'.MRI.SourcePort <- SPNDTGaNATds.

    4. GQ'.NLI.IA.<- IPGaNATds.

    5. GQ'.[IP header].SourceIPAddress <- IPGaNATds.

    6. GQ'.[UDP header].SourcePort <- SPNSTGaNATds.

    7. GQ'.S <- true.

    8. It checks whether or not an NTO was included in GQ.

        - If none was included, it creates a new NTO as follows and adds it to GQ'.

            o NTO.[NAT Count] <- 1.

            o NTO.MRI <- GQ.MRI.

            o NTO.[List of translated objects] <- [type of MRI], [type of NLI], [type of SCD]

            o NTO.opaque information for NAT 1 <- GQ.[IP header].SourceAddress, [link identifier of upstream link].

        - If one was included, it replaces certain fields and appends new fields into the NTO, as follows, and adds the resulting object to GQ'.

- NTO.[NAT Count] <- i, where i is the current [NAT count] value increased by one.

- NTO.[List of translated objects] <- [type of MRI], [type of NLI], [type of STD]

- NTO.[opaque information replaced by NAT i] <- GQ.[IP header].SourceAddress, [LinkID of upstream link].

9. It forwards GQ' on the downstream link. Note: The encoding of the information that the GaNAT encodes into the NTO.[opaque information replaced by NAT i] field is a local implementation issue.

2. Otherwise, if P carries an [IP header].DestinationIPAddress that belongs to the GaNAT, and if it is identified as a GIST RESPONSE packet in datagram mode with an NSLP ID that is not supported, the GaNAT does the following (P is denoted as GR).

   1. If P does not contain an NTO, the GaNAT forwards it as usual without further processing. Otherwise, the GaNAT selects the information encoded by it in the [opaque information replaced by NAT] field of the embedded NTO, denoted by IPAddressToSend and LinkID. If multiple [opaque information replaced by NAT] fields are present in the NTO, the GaNAT uses the last one in the list. It then constructs a new GIST response GR', as follows (note that no changes are made to the MRI).

      1. GR' <- GR

      2. GR'.[IP header].DestinationIPAddress <- IPAddressToSend.

      3. GR'.NTO.[NAT Count] <- current value minus one.

      4. Remove the last [opaque information replaces by NAT i] field from GR'.NTO

      5. GR'.S <- true.

2. It forwards GR' on the upstream link, i.e. the link identified by LinkID.

3. The GaNAT SHOULD now invalidate all but one stack configuration objects in the stack proposal in GR'. This is done so that the quering node can only chose that one proposal, and that therefore only one NAT binding must be installed for the signalling traffic to traverse the GaNAT. The GaNAT SHOULD keep valid the strongest, in terms of security, stack proposal. We denote this proposal as PR.

4. The GaNAT now installs a NAT binding for the signalling traffic which says that "a packet K that arrives on the upstream link and for which it holds that

   + K.[IP header].SourceIPAddress=IPAddressToSend,

   + K.[IP header].Protocol=PR.Protocol, and

   + K.[TCP/UDP header].DestinationPort=PR.[Destination Port]

   should be forwarded on the downstream link (i.e. on the link on which GR arrived), with [IP header].SourceIPAddress = IPGaNATds.

5. If no NAT binding for the data traffic was found in step 1.3.2, the GaNAT now installs a NAT binding (for the unidirectional data traffic) which says that "a packet K that arrives on the upstream link and for which it holds that

   + K.[IP header].DestinationIPAddress=GQ'.NTO.MRI.Destination IPAddress,

   + K.[IP header].Protocol=GQ'.NTO.MRI.Protocol, and

   + K.[TCP/UDP header].PortNumbers=GQ'.NTO.MRI.PortNumbers

   should be forwarded on the downstream link (i.e. the link on which GR arrived), with [IP header].SourceIPAddress = IPGaNATds and [UDP/TCP header].SourcePort=SPDTGaNATds.

Issues: there is a question of whether this NAT binding should also enable data traffic in the opposite direction to traverse the NAT; in order to be able to demultiplex upstream traffic that carries data that belongs to different flows, the GaNAT should keep the necessary per-flow state. From a signalling point of view, however, upstream data traffic that corresponds (on the application level) to the downstream flow to which this GIST session refers, is a separate flow for which, dependent on the application, there may or there may not exist a signalling session. If such a signalling session exists, then the GaNAT acts as an NR-side GaNAT for this session. Thus, during the processing of this signalling, care has to be taken not to establish a NAT binding for a flow for which a NAT binding already exists. Finally, security issues may arise when traffic, for which no signalling exists, is allowed to traverse a GaNAT.

3. Otherwise, if P matches an existing NAT binding, normal NAT processing is applied.

4. Otherwise, P is silently discarded.

6.2. NR-side NSLP-unaware GaNATs

As is the case with NR-side NSLP-unaware GaNATs that follow the "transparent" approach, an NR-side NSLP-unaware GaNAT that follows the "non-transparent" approach must know a "pending" IP address and, depending on the scenario, also a destination port number, as described in Section 5.2. This IP address and destination port number are denoted as IPNext and PortNext respectively. How they are made known to the GaNAT is outside the scope of this document. Note, however, that a typical scenario would be that the GaNAT has an existing NAT binding for the data flow in place from where this information can be derived.

For every arriving IP packet P, an NSLP-unaware, NR-side GaNAT executes the following algorithm.

1. If P has a RAO followed by a GIST header with an unsupported NSLPID, and is identified as a GIST QUERY, the GaNAT does the following.

    1. We denote P as GQ. The GaNAT looks at the stack proposal ST in GQ. If it indicates that no cryptographic protection is required, the GaNAT MAY choose to follow the "transparent" approach as described in Section 5.2 above.

2. If GQ.[IP header].[Destination Address] is not bound to the link on which GQ arrived, the GaNAT silently discards the packet.

3. The GaNAT determines whether or not this GIST QUERY is anticipated, i.e. if a pending IPNext and PortNext exists. One way of determining whether or not a pending IPNext and PortNext exists is checking whether or not a NAT binding for the data traffic, as this is defined by the MRI in the GIST QUERY, exists in the NAT binding cache. If one exists, then IPNext and PortNext is the address and port number to which the data traffic is sent after translation. If no pending IPNext is found, GQ is discarded (it is an open issue whether or not an error message should be sent). Otherwise, additional checks may be performed (e.g. a DSInfo object may have to be checked against the GQ). If these checks fail, GQ is discarded. Otherwise, the GaNAT performs the following.

4. We call the link on which GQ arrived the "upstream" link. The GaNAT aquires an IP address for itself for the upstream link (this could be a static or a dynamic IP address). This address is denoted IPGaNATus. The GaNAT will use this address as a source IP address in order to send subsequent signalling messages to the upstream direction. If the GaNAT is an edge NAT, IPGaNATus will typically coincide with the destination IP address of the (original) MRI in the GIST QUERY.

5. The GaNAT searches its table of existing NAT bindings against entries that match the GQ.MRI. A matching entry means that the data flow, to which the signalling refers, already exists.

   + If a matching entry is found, the GaNAT determines the link on which the packets of the data flow are forwarded; we call this link the "downstream" link. Further, the GaNAT checks how the headers of the data flow (IP addresses, port numbers) are translated according to this NAT binding. Note that the [IP header].DestinationIPAddress of this NAT binding should be equal to IPNext. If it is not, this should be handled as an auditive error condition. (This check is done as a consistency check.)

+ If no matching entry is found, the GaNAT determines, based on its routing table, the link on which packets that match GQ.MRI (where GQ.MRI.DestinationIPAddress is replaced with IPNext) would be forwarded. We call this link the

"downstream" link.

6. It creates a new GIST QUERY packet GQ', as follows.

   1. GQ' <- GQ

   2. GQ'.MRI.DestinationAddress <- IPNext.

   3. GQ'.MRI.DestinationPort <- PortNext.

   4. GQ'.[IP header].DestinationIPAddress <- IPNext.

   5. It checks whether or not an NTO was included in GQ.

      - If none was included, it creates a new NTO as follows and adds it to GQ'.

        o NTO.[NAT Count] <- 1.

        o NTO.MRI <- GQ.MRI.

        o NTO.[List of translated objects] <- [type of MRI], [type of NLI], [type of SCD]

        o NTO.opaque information for NAT 1 <- IPGaNATus, [link identifier of upstream link].

      - If one was included, it replaces certain fields and appends new fields into the NTO, as follows, and adds the resulting object to GQ'.

        o NTO.[NAT Count] <- i, where i is the current [NAT count] value increased by one.

        o NTO.[List of translated objects] <- [type of MRI], [type of NLI], [type of SCD]

- o NTO.[opaque information replaced by NAT i] <- IPGaNATus, [LinkID of upstream link].

6. It forwards GQ' on the downstream link. Note: The encoding of the information that the GaNAT encodes into the NTO.[opaque information replaced by NAT i] field is a local implementation issue.

2. Otherwise, if P is identified as a GIST RESPONSE packet in datagram mode with an NSLP ID that is not supported, the GaNAT does the following (P is denoted as GR).

1. If P does not contain an NTO, the GaNAT forwards it as usual without further processing. Otherwise, the GaNAT selects the information encoded by it in the [opaque information replaced by NAT] field of the embedded NTO, denoted by IPGaNATus and LinkID. If multiple [opaque information replaced by NAT] fields are present in the NTO, the GaNAT uses the last one in the list. The GaNAT then constructs a new GIST response GR', as follows (note that no changes are made to the MRI).

   1. GR' <- GR

   2. GR'.[IP header].SourceIPAddress <- IPGaNATus.

   3. GR'.MRI.NLI.IA <- IPGaNATus.

   4. Remove the last [opaque information replaced by NAT i] field from GR'.NTO.

   5. GR'.S <- true.

2. The GaNAT SHOULD now invalidate all but one stack configuration objects in the stack proposal in GR'. This is done so that the quering node can only chose that one proposal, and that therefore only one NAT binding must be installed for the signalling traffic to traverse the GaNAT. The GaNAT SHOULD keep valid the strongest, in terms of security, stack proposal. We denote this proposal as PR. If PR.[Destination Port] is already used by the GaNAT as a port in order to demultiplex an existing signalling flow, the

GaNAT reserves a port SIGPORT (that it will use as a source port for UDP/TCP signalling traffic that it will send on the upstream link) and replaces PR.[Destination Port] with SIGPORT. Otherwise it sets SIGPORT=PR.[Destination Port]. It then sets GR'.[UDP header].SourcePort <- SIGPORT.

3. It forwards GR' on the upstream link, i.e. the link identified by LinkID.

4. The GaNAT now installs a NAT binding for the signalling traffic which says that "a packet K that arrives on the upstream link and for which it holds that

   + K.[IP header].DestinationIPAddress=IPGaNATus,

   + K.[IP header].Protocol=PR.Protocol, and

   + K.[TCP/UDP header].DestinationPort=SIGPORT

should be forwarded on the downstream link (i.e. on the link on which GR arrived), with [IP header].DestinationIPAddress = GR.MRI.NLI.IA and [UDP/TCP header].DestinationPort=PR.[Destination Port].

5. If no NAT binding for the data traffic was found in step 1.3.2, the GaNAT may now install a NAT binding (for the unidirectional data traffic) which says that "a packet L that arrives on the upstream link and for which it holds that

   + L.[IP header].DestinationIPAddress=GR'.NTO.MRI.Destination IPAddress,

   + L.[IP header].Protocol=GR'.NTO.MRI.Protocol, and

   + L.[TCP/UDP header].DestinationPortNumbers=GR'.NTO.MRI.DestinationPort

   should be forwarded on the downstream link, with [IP header].DestinationIPAddress = IPNext and [UDP/TCP header].DestinationPort=PortNext.

Issues: there is a question of whether this NAT binding should also enable data traffic in the opposite direction to traverse the NAT; in order to be able to multiplex upstream traffic that carries data that belongs to different flows, the GaNAT should keep the necessary per-flow state. From a signalling point of view, however, upstream data traffic that corresponds (on the application level) to the downstream flow to which this GIST session refers, is a separate flow for which, dependent on the application, there may or there may not exist a signalling session. If such a signalling session exists, then the GaNAT acts as an NI-side GaNAT for this session. Thus, during the processing of this signalling care has to be taken not to establish a NAT binding for a flow for which a NAT binding already exists. Finally, security issues may arise when traffic, for which no signalling exists, is allowed to traverse a GaNAT.

3. Otherwise, if P matches an existing NAT binding, normal NAT processing is applied.

4. Otherwise, P is silently discarded.

6.3. GIST peer processing

In the presence of GaNATs on the signalling path between two NSLP peers, and if the GaNATs follow the "non-transparent" approach (which

they have to follow in the context of cryptographically protected signalling), the consistent translation of the GIST header fields must be carried out by the NSLP peers. The GIST processing that performs this task, is described in section 7.2 of [1].

7. GIST-unaware NATs

The following may serve as indications for the existence of an GIST-unaware NAT between two GIST peers. These indications can only be detected by the receiver of a GIST message. The first occasion these indications may be detected is with the reception of a GIST QUERY, typically by the downstream peer. (Note that != denotes inequality).

- The MRI.SourceIPAddress does not belong to the addressing space of the receiving peer.

- The MRI.DestinationIPAddress does not belong to the addressing space of the receiving peer.

- The IP address in the NLI.IA field does not belong to the addressing space of the receiving peer.

- The D flag of a received GIST packet denotes downstream direction and the S flag is not set and [IP header].SourceIPAddress != MRI.SourceIPAddress.

- The D flag of a received GIST packet denotes upstream direction and the S flag is not set and [IP header].SourceIPAddress != MRI.DestinationIPAddress.

- This is a GIST QUERY and [IP header].DestinationIPAddress != MRI.DestinationIPAddress.

Note that these are only indications. In the presence of an adversary, a GIST peer may be tricked into believing that an GIST-unaware NAT exists between itself and one of its neighbouring peers, while in reality this may not be the case.

When a downstream GIST peer detects such an indication, it may notify the upstream peer about the error. It may include additional information that enables the upstream peer to construct a GIST packet in such a way that, after it traverses the GIST-unaware NAT, the IP addresses in the MRI field and the NLI.IA field are consistent with those in the IP header (which match the addressing space of the receiving peer). However, this requires the specification of new data structures and formats, processing rules, and requires the peers to maintain additional state.

Unfortunately, this approach is likely to fail in many circumstances. In order to see this, consider the behaviour of an GIST-unaware NAT when it receives an IP packet. The packet either

1. matches an existing NAT binding in which case its IP header is

translated and the packet it is forwarded on another link, or

2. matches an existing policy rule which causes a new binding to be established and then (1) happens, or

3. is discarded because neither (1) nor (2) applies.

With GIST-unaware NATs it is a matter of local policy (i.e. the rules that exist in case (2) above) whether or not traffic will be allowed to traverse the NAT. This obviously applies to both signalling and data traffic, as an GIST-unaware NAT is unable to distinguish the two types of traffic. It may be the case that GIST node A is unable to contact GIST node B which is "behind" a NAT, even if communication in from B to A may be possible because such communication would match a policy rule; typically, in a scenarios where A is towards the NI and B is towards the NR, the NAT would have this behaviour.

Another approach to deal with GIST-unaware NATs is similar to the NAT traversal approach taken by IKEv2, i.e. by encapsulating GIST messages into UDP datagrams, rather than directly into IP datagrams. This technique requires the inclusion of additional fields into a GIST QUERY, as follows. The sender adds (a hash of) its own IP address and the IP address of what it believes to be the DR into the GIST payload. The receiver of this GIST messages compares these addresses to the [IP header].SourceIPAddress and the [IP header].DestinationIPAddress respectively. If at least one of them is unequal, the receiver deduces that a NAT is between sender and receiver. After the detection of a NAT, the remainder of the communication is encapsulated into UDP datagrams that are addressed to a specified port.

Unfortunately, the IKEv2 NAT traversal mechanism cannot be used "as is" for NAT traversal in GIST. This is because of a number of reasons, including the following.

- The NAT may use an IP address for the forwarding of data traffic that is different from the IP address it uses to forward GIST traffic. Since the NAT is GIST-unaware it cannot update the MRI in the GIST messages such that it matches the translation applies to the data traffic. Moreover, neither the GIST sending, nor the GIST receiving peer can perform this update; the sending peer cannot predict the translation that the NAT will apply, and the receiving peer does not have enough information to associate data flows to signalling messages.

- It is unclear whether or not the IKEv2 NAT traversal mechanism supports cascades of NATs.

- It seems to be inappropriate to use UDP encapsulation for certain C-mode scenarios. For example, using UDP encapsulation for TCP C-mode would result in GIST to appear in TCP over UDP over IP.

8. Security Considerations

The mechanisms proposed in this document give rise to a number of threats that must be considered. In the following, a subset of these threats is mentioned.

8.1. Service Denial Attacks

As described above, NSLP-unaware GaNATs create some state whenever they receive a GIST QUERY message. This state is necessary in order for the GaNAT to be able to map a GIST RESPONSE that arrives from the downstream direction to the corresponding GIST QUERY and thereby to perform the required translation.

The threat here is an attacker flooding the GaNAT with maliciously constructed GIST QUERIES with the aim of exhausting the GaNAT's memory. The attacker might use a variety of methods to construct such GIST QUERIES, including the following.

1. Use as [IP header].SourceIPAddress the address of some other node or an unallocated IP address. This method is also known as IP spoofing.

2. Use an invalid NSLPID, in order to make sure that all on-path GaNAT(s) will behave like NSLP-unaware GaNATs.

3. For each packet, use a different value for the cookie field.

4. For each packet, use a different value for the session ID field.

5. Combinations of the above.

How vulnerable a GaNAT is to the above service denial attack depends on a variaty of factors, including the following.

- The amount of state allocated at the receipt of a GIST QUERY. This amount may vary depending on whether or not the data flow to which the signalling refers, already exists (i.e. whether or not the GaNAT already maintains a NAT binding for it).

- The mechanism that the GaNAT uses to map RESPONSEs to QUERIEs.

- Whether or not the GaNAT acriques dynamic IP addresses and ports for the downstream link.

In order to decrease the exposure of a GaNAT to service denial attacks, the following recommendations are made.

- The GaNAT should perform ingress filtering. This limits the amount of locations from which an attacker can perform IP spoofing without being detected.

- The GaNAT should allocate the minimum amount of state required at the reception of a GIST QUERY.

- All state allocated by the GaNAT should timeout according to a local policy. If the GaNAT detects heavy loads (which may indicate a service denial attack in progress), the GaNAT should timeout the state allocated as a result of a received GIST QUERY quicker, proportionally to the experienced load.

- The installation of a NAT binding for the data traffic (if such a binding does not exist prior to signalling) should be postponed until the correct GIST REPONSE traverses the NAT.

The service denial threats mentioned in this section do not apply to an NSLP-aware GaNAT, as such a GaNAT is required, in accordance with its local policy, to verify the validity of the cookie(s) before allocating any state, including the state required by the mechanisms in this document.

8.2. Network Intrusions

Although the primary goal of a NAT is to perform address translation between two addressing spaces, NATs are sometimes also used to provide a security service similar to the security service provided by firewalls. That is, a NAT can be configured so that it does not forward packets from the external into the internal network, unless it determines that the packets belong to a communication session that was originally initiated from an internal node and are, as such,

solicited.

If an NSLP-unaware GaNAT performs the above security-relevant function in addition to address translation, then the presence of GIST signalling and, in particular the mechanisms described in this document, might allow an adversary cause the installation of NAT bindings in the GaNAT using these mechansisms. These NAT bindings would then enable the adversary to inject unsolicited traffic into the internal network, a capability that it may not have in the absence of the mechanisms described in this document.

The administrator of an NSLP-unaware GaNAT should therefore make security-concious decisions regarding the operation of the GaNAT. An NSLP-aware GaNAT, on the other hand, follows an NSLP policy which indicates the required security mechanisms. This policy should account for the fact that this NSLP-aware node performs also NAT and

the associated packet filtering.

9. IAB Considerations

None.

10. Acknowledgements

The authors would like to thank Cedric Aoun, Christian Dickmann, Robert Hancock, and Martin Stiemerling for their insightful comments. Furthermore, we would like to mention that this document builds on top of a previous document regarding migration scenarios.

11. Normative References

[1] Schulzrinne, H. and R. Hancock, "GIST: General Internet Signalling Transport", draft-ietf-nsis-ntlp-06 (work in

progress), May 2005.

[2] Stiemerling, M., Tschofenig, H., and C. Aoun, "NAT/Firewall NSIS Signaling Layer Protocol (NSLP)", draft-ietf-nsis-nslp-natfw-06 (work in progress), May 2005.

Authors' Addresses

Andreas Pashalidis
Siemens

Otto-Hahn-Ring 6
Munich, Bavaria 81739
Germany

Email: Andreas.Pashalidis@siemens.com

Hannes Tschofenig
Siemens
Otto-Hahn-Ring 6
Munich, Bavaria 81739
Germany

Email: Hannes.Tschofenig@siemens.com

Intellectual Property Statement

Disclaimer of Validity

Copyright Statement

Acknowledgment

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