NSIS
Internet-Draft
Expires: August 5, 2006

A. Pashalidis
H. Tschofenig
Siemens

February 2006

GIST NAT Traversal
draft-pashalidis-nsis-gimps-nattraversal-02.txt

Status of this Memo

Copyright Notice

Abstract

This document describes a number of mechanisms for the implementation of the General Internet Signalling Transport (GIST) protocol [1] on different types of Network Address Translator (NAT). The focus of these mechanisms is the interaction of GIST with the address translation function of the NAT, and their purpose is to enable GIST hosts that are located on either side of the NAT to correctly interpret signalling messages with respect to the data traffic they refer to. The purpose of this document is to provide guidance to

people that implement GIST and NSLPs on both NAT and non-NAT nodes.

Table of Contents

1. Introduction

Network Address Translators (NATs) modify certain fields in the IP and transport layer 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 data traffic that traverses these nodes.

This document describes mechanisms that can be used in order for GIST signalling messages to 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 subsequently proposed mechanisms are based. The mechanisms are described in Section 5 and Section 6. Finally, Section 7 presents some security issues that arise in conjunction with the mechanisms described in this document.

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 document 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. The term is synonymous to NSIS-unaware.

- 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 object, 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 the data traffic.

A NAT may either be GIST-unaware or GIST-aware. We refer to 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 involved NATs. Consider a GaNAT that supports both the NATFW NSLP, and the NAT traversal mechanism that is described in this document (which operates at the GIST layer). 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-layer NAT traversal mechanism (described 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 a GaNAT should implement a policy according to which one method is used in preference to the other. Note that, however, if the GaNAT prefers GIST-layer NAT traversal, then it may happen, if no on-path GaNATs exist that prefer the NATFW NSLP, that no downstream NATFW NSLP peers are discovered. This may make the entire NATFW session obsolete. It is therefore anticipated that the NATFW NSLP will be the preferred NAT traversal mechanism in most circumstances.

However, in certain cicumstances it may be desirable for GIST signalling messages to traverse a NAT, and not desirable or possible to use the NATFW NSLP for this purpose. Examples of such circumstances are the following.

- GaNATs that do not implement the NATFW NSLP are on the path taken by GIST signalling messages. This situation may arise during incremental deployment of the signalling protocols that are developed by the NSIS working group.

- GaNATs that implement the NATFW NSLP are on the path taken by GIST signalling messages that refer to a given data flow. However, the NSLP that is being signalled is *not* the NATFW NSLP and there

exists no NATFW signalling session for the data flow in question.

Describing NAT traversal for GIST signalling messages in the above circumstances is the subject matter of this document.

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 infrastructure 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 GIST-layer NAT traversal 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. 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 peer is invalid (or the IP address is unreachable) in the address space of the upstream 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 the preserves consistency

after address translation. This processing deviates from the processing of NSLP-aware non-NAT nodes. The following sections describe how to overcome the limitation of two adjacent NSLP peers not being able to execute the NSLP in the presence of in-between NAT(s).

A number of different variations are possible, depending on the level of NSIS support by the in-between NAT(s). The following combinations of NATs that are located between two adjacent NSLP peers are considered.

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

- all NAT(s) are NSLP-aware

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, it cannot indicate that an association exists between the two NAT bindings. Therefore, in the presence of such a NAT, non-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 outside the scope of this version of this document.

Traversal of GaNAT(s) is comparatively 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 above-mentioned information into signalling messages. In this document the following two ways are considered.

1. Non-transparent approach: 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. The responder (which is assumed to be located on the "other side" of the NAT) uses 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 for 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 [1]). Unfortunately though, the transparent approach does not work if the signalling traffic is to be cryptographically protected between the two GIST peers that are located either side of the GaNAT, and the GaNAT is NSLP-unaware. 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 desired 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 proposal in the GIST QUERY and RESPONSE messages; inclusion of IPsec or TLS proposals amounts to cryptographic protection being required.

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. If this is not the case, then the NSLP has to provide additional mechanisms for the traversal of (Ga)NATs. These mechanisms must be compatible the mechanisms described in this document. Note that the NATFW NSLP is an exception to this rule in that it does not need to be compatible with the mechanisms described in this document. This is because the GIST-layer NAT traversal mechanisms described in this document and the NATFW NSLP are mutually exclusive (i.e. it is not permissible that a given (Ga)NAT applies both GIST-layer NAT traversal and NATFW NSLP processing to the messages that belong to the same signalling session).

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 a 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 trigger 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.

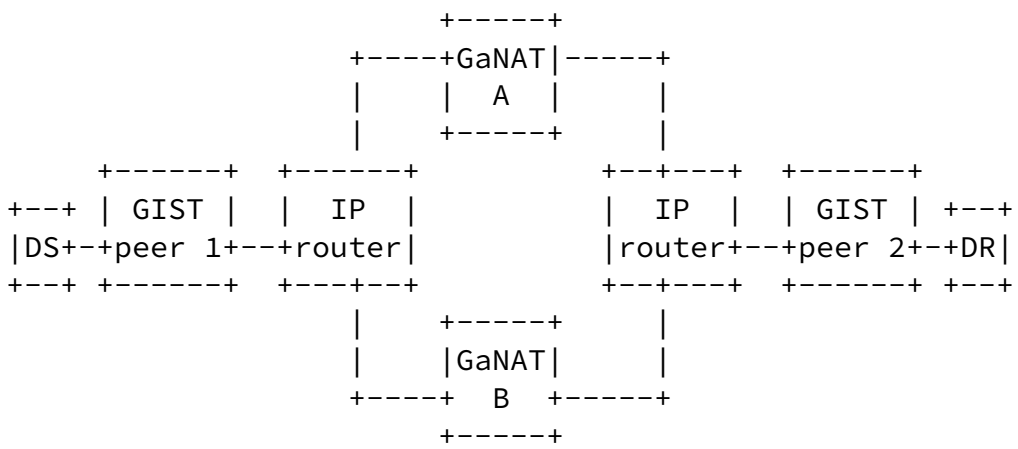

Figure 1: Network with more than one NAT at an addressing boundary

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 necessarily 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.

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. An NSLP-aware GaNAT MUST follow this approach, as described in Section 5.3. An NSLP-unaware GaNAT MAY follow this approach, as described in Section 5.1 and Section 5.2, only if no cryptographic protection of signalling data is requested by the two NSLP peers.

Note that two types of NSLP-unaware GaNAT have to be dealt with, 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 of the downstream GIST peer 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, and if P is identified as a GIST QUERY, the GaNAT performs the following.

    1. We denote P by GQ. The GaNAT looks at the stack proposal in GQ. If it includes a proposal with cryptographic protection, the mechanism that is applied is the one described Section 6.1.

    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.

        + 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. We denote the source IP address of translated data packets by IPds, and their [Transport layer

header].SourcePort by SPDTds.

        + 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, denoted by IPds and SPDTds respectively. This address and port could be dynamic or static, and will be used (among other things) for the installation of a NAT binding for the data traffic in the

future.

4. The GaNAT aquires a source port number for the version of the GIST QUERY that will be forwarded over the downstream link. We denote this port by SPSTds. (There is no requirement that SPSTds must be different from GQ.[UDP Header].SourcePort.)

   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 the downstream direction. Of course, a GaNAT does not need to actually change the source port of signalling traffic; it can always use SPSTds the same port as in the incoming packet. Such a GaNAT may use the GIST session ID in order to demultiplex (i.e. forward to the correct internal address) the traffic that arrives from the downstream direction. It is unclear which of the two approaches is preferable.

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

   1. GQ' <- GQ

   2. GQ'.MRI.SourceIPAddress <- IPds

   3. GQ'.MRI.SourcePortNumber <- SPDTds

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

   5. GQ'.[UDP header].SourcePort <- SPSTds

   6. GQ'.NLI.IA <- IPds

   7. GQ'.S <- true

6. 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.)

7. 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 by GR).

   1. It searches for a matching GQ' in its buffer. A GQ' is said to match a GR 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 fail, GR is discarded. Otherwise, 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 <- IPus, where IPus is an IP address that is bound to the upstream link.

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

      5. GR'.[UDP header].DestinationPort <- GQ.[UDP header].SourcePort

      6. GR'.NLI.IA <- IPus

      7. GR'.S <- true

      8. The GaNAT inspects the Stack-Configuration-Data object in GR' and the corresponding GQ' in order to check whether or not the upstream NSLP peer can select one of multiple transport layer protocol/destination port number combinations for the establishment of a messaging association. If multiple choices exist, the GaNAT invalidates as many transport layer protocol/port number

combination proposals from GR' as necessary, until the upstream NSLP peer can only initiate the establishment of a messaging association with the downstream NSLP peer using a single transport layer protocol/destination port number combination. This invalidation is done by setting the D-flag in those MA-Protocol-Options fields that carry the port number proposals that are to be invalidated. Note that, by setting the D-flag in a particular MA-Protocol-Option field, the GaNAT may also invalidate the associated transport layer protocol and security (e.g. TLS) proposal. The actions of the GaNAT MUST NOT result in the strongest, in terms of security, proposal to be invalidated. In the end, the NAT will expect the upstream NSLP peer to use a particular combination of transport layer protocol and destination port (and possibly other details that are associated with the valid proposal) for the establishment of the messaging association. We call this combination the "stack proposal expected by the NAT" and denote it by ST. The GaNAT remembers this ST, its association with GQ, GQ', GR, GR', and the upstream and downstream links. By doing so, the GaNAT is said to "install" 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.[Transport layer header].PortNumbers=GQ.MRI.PortNumbers

   should be forwarded on the downstream link, with [IP header].SourceIPAddress = IPds and [Transport layer header].SourcePort=SPDTds".

   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. Moreover, security issues may 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 destination port number indicated by some stack 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 checks whether or not the bits where the GIST header is expected, constitute a valid GIST header. If they do not, P is silently discarded. If all is in order, the GaNAT constructs an outgoing packet P' as follows (the variables used below refer to those stored in association with ST).

    1. P' <- P

    2. If P has arrived on the upstream link, then

        1. P'.[IP header].SourceIPAddress <- IPds

2. P'.[IP header].DestinationIPAddress <- GR.NLI.IA

3. P'.MRI <- GQ'.MRI

4. P'.NLI.IA <- IPds

5. The GaNAT forwards P' on the downstream link.

3. else (if P has arrived on the downstream link)

   1. P'.[IP header].SourceIPAddress <- IPus

   2. P'.[IP header].DestinationIPAddress <- GQ.NLI.IA

   3. P'.MRI <- GQ.MRI

   4. P'.NLI.IA <- IPus

   5. 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 header 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, the GaNAT MUST buffer that 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 subjected to normal NAT processing. That is, P is either silently discarded or it causes the installation of a (data) NAT binding.

Brief discussion of the algorithm: The fact that the GaNAT replaces the NSLP peers' NLI.IA with its own IP address (in both directions), causes the GIST peers to send subsequent signalling messages to the GaNAT, in the belief that they talk to their adjacent NSLP 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 outgoing GIST messages is always the same as the size of corresponding incoming GIST messages. Also 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 sees indicates the IPds 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 next 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 and denote it by IPNext. The GaNAT may also have a destination port associated with IPNext. If IPNext is derived from an existing data traffic NAT binding, then this port is typically the destination port after translation from that binding. This port, if known, is denoted PortNext. How IPNext and PortNext are 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.

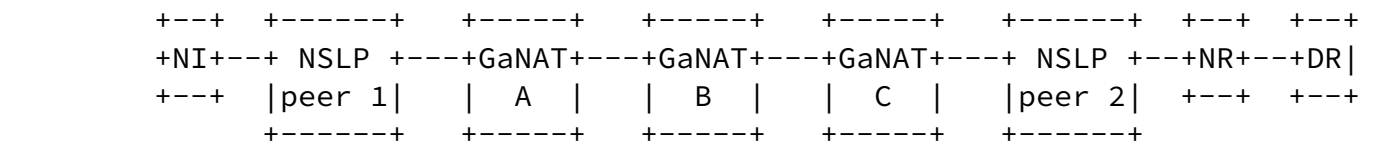

Figure 2: Network with NR-side GaNATs (the public Internet is assumed to be between NI and NSLP peer 1)

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, and if it is identified as a GIST QUERY, the GaNAT does the following.

    1. We denote P by GQ. The GaNAT looks at the stack proposal 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 (and possibly PortNext) 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 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. something like 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 IP and transport layer headers of the data flow are translated according to this NAT binding. Note that the [IP header].DestinationIPAddress and [Transport layer header].DestinationPort of this NAT binding should be equal to IPNext and PortNext respectively. If they are not, this should be handled as an auditive error

condition.

   + 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.

5. 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 a UDP source port number for the version of the GIST QUERY that will be

forwarded to the downstream direction. We denote the acquired IP address and source port number by IPds SPSTds respectively. The GaNAT then constructs a new GIST QUERY packet GQ', as follows.

1. GQ' <- GQ

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

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

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

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

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

7. GQ'.[UDP header].SourcePort <- SPSTds.

8. 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.

The remaining steps 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 MRITS operates at the GIST layer. The motivation behind this is 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 an incoming message (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 data 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 implements. For non-GIST packets, normal NAT behaviour applies. Although the MRITS is part of GIST processing, in order to clarify the exposition, we view it as a somewhat separate processing step (i.e. like a subroutine) that is executed in addition to GIST, as this is specified in [1]. 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

o 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.

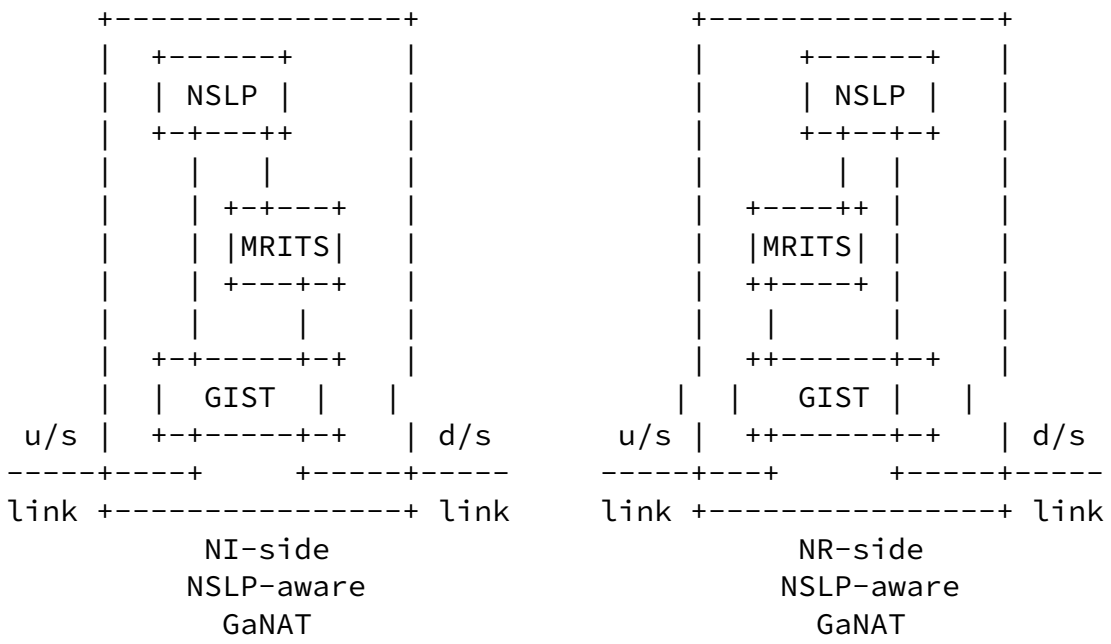

Figure 3: Operation of the MRI Translation Service

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 (IPds and SPDTds are obtained similarly to the case of an NSLP-unaware GaNAT).

    1. MRI.SourceIPAddress <- IPds

    2. MRI.SourcePort <- SPDTds

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

    1. [IP header].SourceIPAddress <- IPds

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

3. NLI.IA <- IPds

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 <- SPDTus, where IPflow is the IP address of the DS (as seen by the GaNAT) and SPDTus 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 <- IPus

   2. [IP header].DestinationIPAddress <- IPpeer

   3. NLI.IA <- IPus

   4. S <- true, where IPus 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 (as seen by the GaNAT). The GaNAT is assumed to determine the correct IPus and IPpeer from previous communications and in cooperation with GIST. [Issue: how exactly should IPus, 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 <- IPds

2. P.MRI.DestinationIPAddress <- IPNext, where IPds 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 <- IPds

    2. [IP header].DestinationIPAddress <- IPNext

    3. NLI.IA <- IPds

    4. S <- true, where IPds 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 peer, the MRITS does the following.

    1. MRI.SourceIPAddress <- IPflow

    2. MRI.Destination_IP_Address <- IPus

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

    1. [IP header].SourceIPAddress <- IPus

    2. [IP header].DestinationIPAddress <- IPpeer

    3. NLI.IA <- IPus

    4. S <- true, where IPus 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 IPus and IPpeer fields from previous communications and in cooperation with GIST. [question: how exactly should IPus 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 for GIST

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 RESPONSE messages, which are never cryptographically protected in their entirety.

In this approach, the GaNAT embeds a "NAT Traversal Object" (NTO) payload type into the GIST QUERY. The NTO encodes the aforementioned

information and is an optional payload in the GIST header of a GIST QUERY. It is added, and processed, by the GaNAT(s) through which the QUERY traverses. The information in the NTO enables 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, and if it is identified as a GIST QUERY, the GaNAT does the following.

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

    2. The GaNAT remembers GQ along with the link on which it arrived. We call this link 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. We denote the source IP address of translated

data packets by IPds, and their [Transport layer header].SourcePort by SPDTds.

+ 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, denoted by IPds and SPDTds respectively. This address and port could be dynamic or static, and will be used (among other things) for the installation of a NAT binding for the data traffic in the future.

4. The GaNAT aquires a source port number for the version of the GIST QUERY that will be forwarded over the downstream link. We denote this port by SPSTds. (There is no requirement that SPSTds must be different from GQ.[UDP Header].SourcePort.)

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

    1. GQ' <- GQ

    2. GQ'.MRI.SourceIPAddress <- IPds

    3. GQ'.MRI.SourcePortNumber <- SPDTds

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

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

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

    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'. Note that the MRI field of the NTO is taken from GQ.

- o NTO.[NAT Count] <- 1.

- o NTO.MRI <- GQ.MRI.

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

- o NTO.opaque information replaced by NAT 1 <- GQ.NLI.IA, GQ.[UDP header].SourcePort, LinkID, where LinkID represents the 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'. Note that the MRI field of the NTO is not modified.

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

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

  - o NTO.opaque information replaced by NAT i <- GQ.NLI.IA, GQ.[UDP header].SourcePort, LinkID, where LinkID represents the upstream link.

9. It remembers GQ, GQ', the fact that they are associated, and the associated upstream and downstream links.

10. 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 with an NSLP ID that is not supported, the GaNAT does the following (P is denoted by GR).

   1. If P does not contain an NTO, the GaNAT discards it without further processing. Otherwise, it searches for a matching GQ' in its buffer. A GQ' is said to be matching if it carries the same cookie value. If none is found, GR is discarded. Otherwise, the GaNAT should also make sure that the session ID in GR is the same as in GQ', that the NSLP IDs

match, and that GR arrived on the downstream link. If these consistency checks fail, GR should be discarded. Otherwise, the GaNAT constructs a new GIST RESPONSE GR', as follows (note that no changes are made to the MRI).

1. GR' <- GR

2. The GaNAT selects the information that it encoded in the [opaque information replaced by NAT i] field of the embedded NTO, denoted by IPAddressToSend, PortAddressToSend and LinkID, where i is the current value of [NAT Count] as indicated in the NTO.

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

4. GR'.[UDP header].DestinationPort=PortAddressToSend.

5. GR'.NTO.[NAT Count] <- reduce by one.

6. GR'.S <- true.

2. The GaNAT inspects the Stack-Configuration-Data object in GR and the corresponding GQ' in order to check whether or not the upstream NSLP peer can select one of multiple transport layer protocol/destination port number combinations for the establishment of a messaging association. If multiple choices exist, the GaNAT invalidates as many transport layer protocol/port number combination proposals from GR' as necessary, until the upstream NSLP peer can only initiate the establishment of a messaging association with the downstream NSLP peer using a single transport layer protocol/destination port number combination. This invalidation is done by setting the D-flag in those MA-Protocol-Options fields that carry the port number proposals that are to be invalidated. Note that, by setting the D-flag in a particular MA-Protocol-Option field, the GaNAT may also invalidate the associated transport layer and security protocol (e.g. TCP/TLS) proposal. The actions of the GaNAT MUST NOT result in the strongest, in terms of security, proposal to be invalidated. In the end, the NAT will expect the upstream NSLP peer to use a particular combination of transport layer protocol and destination port (and possibly other details that are associated with the valid proposal) for the establishment of the messaging association. We call this combination the "stack proposal expected by the NAT" and denote it by ST. The GaNAT remembers this ST, its association with GQ, GQ', GR, GR', and the upstream and downstream links. By doing so, the GaNAT is said to "install" ST.

3. It forwards GR' on the link identified by LinkID.

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

   + K.[IP header].DestinationIPAddress=GR.NLI.IA,,

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

   + K.[Transport layer header].DestinationPort=ST.DestinationPort

   should be forwarded on the downstream link, with [IP header].SourceIPAddress = IPds and [UDP/TCP header].DestinationPort=SIGPort, where SIGPort is a port that the GaNAT allocates for use as a source port for signalling traffic.

5. The GaNAT now installs a NAT binding for the UDP-encapsulated signalling traffic which says that "a packet M that arrives on the upstream link and for which it holds that

   + M.[IP header].DestinationIPAddress=GR.NLI.IA,

   + M.[IP header].Protocol=UDP, and

   + M.[UDP header].DestinationPort=GIST well-known port

   should be forwarded on the downstream link, with [IP header].SourceIPAddress = IPds. Note that this is a special type of NAT binding, in that the source port in M may vary from one incoming message to another. This is why each packet M may be mapped by the GaNAT to a different source port. Translation in the upstream direction must be applied consistently, and timeouts must also be selected appropriately. That is, the overall binding must be timed out together with the GIST state that is associated with this session. However, each incoming packet M that matches this binding causes the installation of a "sub"-binding (in the sense that a new port mapping may occur) that will typically time out faster.

6. 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 L that arrives on the upstream link and for which it holds that

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

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

+ L.[Transport layer header].PortNumbers=GQ.MRI.PortNumbers

should be forwarded on the downstream link, with [IP header].SourceIPAddress = IPds and [UDP/TCP header].SourcePort=SPDTds.

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 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 subjected to normal NAT processing. That is, P is either silently discarded or it causes the installation of a (data) NAT binding.

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 optionally destination port number, as described in Section 5.2. This IP address and destination port number are denoted by 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 in place from where this information can be derived.

For every incoming IP packet P, an NSLP-unaware, NR-side GaNAT

executes the following algorithm.

1. If P carries an [IP header].DestinationIPAddress that belongs to the GaNAT, if it has a RAO followed by the GIST header with an unsupported NSLPID, and if it is identified as a GIST QUERY, the GaNAT does the following.

    1. We denote P by GQ. The GaNAT looks at the stack proposal in GQ. If it does not include any proposal with cryptographic protection, the GaNAT MAY choose to follow the "transparent" approach as described in Section 5.2 above.

    2. If GQ.[IP header].DestinationIPAddress, denoted by IPus in the sequel, is not bound to the link on which GQ arrived, the GaNAT silently discards the packet. Otherwise, it 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 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 destination port number on which this traffic is forwarded. If no pending IPNext is found, then GQ 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, GQ is discarded. Otherwise, the GaNAT performs the following.

4. It searches its table of existing NAT bindings against entries that match 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) are translated according to this NAT binding. Note that the [IP header].DestinationIPAddress and DestinationPort in this NAT binding should be equal to IPNext and PortNext respectively. If they are 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.

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

   1. GQ' <- GQ

   2. GQ'.MRI.DestinationIPAddress <- IPnext

   3. GQ'.MRI.DestinationPortNumber <- PortNext

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

   5. GQ'.[UDP header].DestinationPort <- GIST well-known port (TBD)

   6. 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'. Note that the MRI field of the NTO is taken from GQ.

- NTO.[NAT Count] <- 1.
- NTO.MRI <- GQ.MRI.
- NTO.opaque information for NAT 1 <- LinkID 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'. Note that the MRI field of the NTO is not modified.

  - NTO.[NAT Count] <- i, where i is the current [NAT count] value increased by one.
  - NTO.opaque information replaced by NAT i <- LinkID of upstream link.

7. It remembers GQ, GQ', the fact that they are associated, and the associated upstream and downstream links.

8. It forwards GQ' on the downstream link.

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

   1. It searches for a matching GQ' in its buffer. A GQ' is said to be matching if it carries the same cookie value. If none is found, GR is discarded. Otherwise, the GaNAT should also make sure that the session ID in GR is the same as in GQ', that the NSLP IDs match, and that GR arrived on the downstream link. If these consistency checks fail, GR should be discarded. Otherwise, the GaNAT constructs a new GIST RESPONSE GR', as follows.

   2. If P does not contain an NTO, the GaNAT discards it without further processing. Otherwise, the GaNAT constructs a new

GIST RESPONSE GR', as follows (note that no changes are made to the MRI).

1. GR' <- GR.

2. The GaNAT selects the information that it encoded in the [opaque information replaced by NAT i] field of the embedded NTO, denoted by LinkID, where i is the current value of [NAT Count] as indicated in the NTO.

3. GR'.NLI.IA <- IPus

4. GR'.NTO.[List of translated objects by NAT i] <- [type of NLI], where i is the current value of [NAT Count] as indicated in the NTO.

5. GR'.NTO.[NAT Count] <- reduce by one.

6. GR'.[IP header].SourceIPAddress <- IPus (this is the IP address that is bound to the link identified by LinkID and must be equal to GQ.[IP header].DestinationIPAddress, where GQ is the GIST QUERY associated with GQ').

7. GR'.[UDP header].DestinationPort <- GQ.[UDP header].SourcePort, where GQ is the GIST QUERY associated with GQ'.

8. GR'.S <- true.

3. The GaNAT inspects the Stack-Configuration-Data object in GR and the corresponding GQ' in order to check whether or not the upstream NSLP peer can select one of multiple transport layer protocol/destination port number combinations for the

establishment of a messaging association. If multiple choices exist, the GaNAT invalidates as many transport layer protocol/port number combination proposals from GR' as necessary, until the upstream NSLP peer can only initiate the establishment of a messaging association with the downstream NSLP peer using a single transport layer protocol/destination port number combination. This invalidation is done by setting the D-flag in those MA-Protocol-Options fields that

carry the port number proposals that are to be invalidated. Note that, by setting the D-flag in a particular MA-Protocol-Option field, the GaNAT may also invalidate the associated transport layer and security protocol (e.g. TCP/TLS) proposal. The actions of the GaNAT MUST NOT result in the strongest, in terms of security, proposal to be invalidated. In the end, the NAT will expect the upstream NSLP peer to use a particular combination of transport layer protocol and destination port (and possibly other details that are associated with the valid proposal) for the establishment of the messaging association. We call this combination the "stack proposal expected by the NAT" and denote it by ST. The GaNAT remembers this ST, its association with GQ, GQ', GR, GR', and the upstream and downstream links. By doing so, the GaNAT is said to "install" ST. If ST.DestinationPort is already used by the GaNAT as a destination port in order to demultiplex an existing flow, the GaNAT reserves a destination port SIGPORT and modifies the valid port proposal in GR' such that SIGPORT will be used by the upstream GIST peer. Otherwise it sets SIGPORT=ST.DestinationPort.

4. It forwards GR' on the link identified by LinkID (i.e. the upstream link).

5. The GaNAT now installs a NAT binding for the signalling traffic that is exchanged over a messaging association which says that "a packet K that arrives on the upstream link and for which it holds that

   + K.[IP header].DestinationIPAddress=IPus (which is equal to GQ.MRI.DestinationIPAddress and GQ.[IP header].DestinationIPAddress),

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

   + K.[Transport layer header].DestinationPort=SIGPORT

   should be forwarded on the downstream link, with [IP header].DestinationIPAddress = GR.NLI.IA and [Transport layer header].DestinationPort=ST.DestinationPort.

6. The GaNAT now installs a NAT binding for the UDP-encapsulated

signalling traffic which says that "a packet M that arrives on the upstream link and for which it holds that

+ M.[IP header].DestinationIPAddress=IPus,

+ M.[IP header].Protocol=UDP, and

+ M.[UDP header].DestinationPort=GIST well-known port

should be forwarded on the downstream link, with [IP header].SourceIPAddress = GR.NLI.IA". Note that this is a special type of NAT binding, in that the source port in M may vary from one incoming message to another. This is why each packet M may be mapped by the GaNAT to a different source port. Translation in the upstream direction must be applied consistently, and timeouts must also be selected appropriately. That is, the overall binding must be timed out together with the GIST state that is associated with this session. However, each incoming packet M that matches this binding causes the installation of a "sub"-binding (in the sense that a new port mapping may occur) that will typically time out faster.

7. 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 L that arrives on the upstream link and for which it holds that

   + L.[IP header].DestinationIPAddress=IPus (which is equal to GQ.MRI.DestinationIPAddress and GQ.[IP header].DestinationIPAddress),

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

   + L.[Transport layer header].PortNumbers=GQ.MRI.PortNumbers

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

   Note: If the GaNAT also allows data traffic to traverse in the other direction (i.e. in the upstream direction), then the IP packets of this data traffic MUST have SourceIPAddress=IPus, SourcePort=GQ.MRI.DestinationPort, DestinationPort=GQ.MRI.SourcePort, and must be forwarded on the upstream link. (This applies anyway for GaNATs with only two links and where each link is bound to a single IP

address. However, for other types of GaNAT care has to be taken that this restriction is enforced.)

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 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 subjected to normal NAT processing. That is, P is either silently discarded or it causes the installation of a (data) NAT binding.

The remaining steps of the algorithm are analogous to the algorithm of NSLP-unaware, NR-side GaNATs, which was described in the previous section.

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 next. Note that this processing is in addition to the processing described in [1]. Also note that the processing described in this section applies only to non-NAT nodes.

A GIST peer that receives a GIST QUERY that carries an NSLP ID for a supported NSLP and an NTO, constructs a GIST RESPONSE according to [1]. This response is sent to the public address of the last in-between GaNAT. This address appeared as NLI.NI in the GIST QUERY

(and also as the source address in the IP header).

If local policy allows the installation of state without the reception of a GIST CONFIRM message, then the responder stores the NTO carried with the QUERY together with the routing state information about the querying GIST peer. In particular, the MRI field of the NTO must be saved in order for the peer to be able to map subsequently received signalling messages to this signalling session.

Note that it is not sufficient for the NSLP to exclusively rely on the NTO.MRI for this purpose. In order to see this, consider two private addressing domains, A and B, each with a GaNAT at its border, and a node N in the public internet. In domain A, node N1 has a communication session with N, and in domain B, node N2 also has a communication session with N. Suppose that the (private) IP addresses of N1 and N2 are equal (e.g. 192.168.0.3), and that they both communicate with N using the same source and destination ports. If they both have an NSIS signalling session for this data traffic, the NTO.MRI field in the GIST QUERY of their respective signalling sessions are identical. If these signalling sessions meet at an NSLP node that is located "after" the GaNATs, then this NSLP node sees the same MRI in signalling messages that are received over a messaging association. In this case, the node must use other information in the signalling messages (e.g. session ID, source IP address) in order to map subsequently received signalling messages to existing sessions.

If local policy demands that no session-specific state is installed before the reception of a GIST CONFIRM message, then the responder must encode the information in NTO.MRI and NLI.IA from the GIST QUERY (and possibly other values such as NSLP ID and an identifier of the link on which the GIST QUERY arrived) in the responder cookie. Since this cookie is echoed in the GIST CONFIRM message, the responder can then delay the installation of the relevant state until it receives the GIST CONFIRM. The construction of the responder cookie is implementation-specific, in the sense that it does not raise interoperability issues. Nevertheless, the cookie must be generated in a way that meets the requirements listed in section 8.5 of [1], and in a way that does not introduce additional attacks against the system.

Two responder cookie construction mechanisms are described in the sequel. These methods are in addition to those described in section 8.5 of [1], and meet the requirements listed in that section. Additionally, they enable the responder to authenticate the contents of the cookie, i.e. to ensure that the cookie was not tampered with while in transit. This feature is not provided by the cookie construction mechanisms described in [1].

Responder cookie generation mechanism 1: Responder cookie = (gennum || cookie-left || cookie-right), where || denotes concatenation, cookie-left is computed as ENC (Q-Node NLI, MRI, NSLPID, reception interface, [timestamp]), and cookie-right is computed as MAC (cookie-left). ENC denotes a semantically secure symmetric encryption scheme, and MAC denotes an unforgeable message authentication code scheme. The responder must use a key with ENC that has been selected independently from the one used with MAC. Whenever these keys are refreshed, they MUST be refreshed together. Gennum is the generation number of the ENC and MAC keys. The timestamp is an optional field. Policy dictates whether or not it is included in the construction of the cookie. For example, responders that have a fast enough key rollover may omit the timestamp. Example algorithms for ENC and MAC are AES-128 in CBC mode [3], and HMAC-SHA1 [4].

Responder cookie generation mechanism 2: Responder cookie = (Gennum || AUTHENC (Q-Node NLI, MRI, NSLPID, reception interface, [timestamp])) AUTHENC denotes a symmetric authenticated encryption scheme. Gennum is the generation number of the key used with AUTHENC. The timestamp is an optional element for the same reason as above. Example AUTHENC algorithms include the one specified in RFC3610.

The version of the MRI that the NSLP peers pass to the NSLP is the one in the header of the GIST QUERY (not the one in the NTO, if one is present). Whether or not this is a translated MRI depends on the location of the peer with respect to the in-between GaNAT(s). Note that the same MRI is used by the responder in signalling messages that are sent towards the downstream direction.

7. Security Considerations

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

7.1. 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 variety 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 acquires 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 RESPONSE 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.

7.2. 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 to 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 might not have in the absence of the mechanisms described in this document.

The administrator of an NSLP-unaware GaNAT should therefore make security-conscious 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.

8. IAB Considerations

None.

9. 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.

10. Normative References

[1] Schulzrinne, H. and R. Hancock, "GIST: General Internet Signalling Transport", draft-ietf-nsis-ntlp-09 (work in progress), February 2006.

[2] Stiemerling, M., Tschofenig, H., and C. Aoun, "NAT/Firewall NSIS Signaling Layer Protocol (NSLP)", draft-ietf-nsis-nslp-natfw-09 (work in progress), February 2006.

[3] "Advanced Encryption Standard (AES)", FIPS PUB 197, November 2001.

[4] Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, February 1997.

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

WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Copyright Statement

Acknowledgment

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