NSIS Working Group
Internet-Draft
Expires: August 16, 2004

M. Stiemerling
NEC
H. Tschofenig
Siemens
M. Martin
NEC
C. Aoun
Nortel Networks

February 16, 2004

NAT/Firewall NSIS Signaling Layer Protocol (NSLP)
draft-ietf-nsis-nslp-natfw-01

Status of this Memo

Copyright Notice

Abstract

This memo defines the NSIS Signaling Layer Protocol (NSLP) for Network Address Translators and Firewalls. The network scenarios, problems and solutions for path-coupled Network Address Translator and Firewall signaling are described. The overall architecture is given by the framework and requirements defined by Next Steps in Signaling (NSIS) working group. This is one of two NSIS Signaling Layer Protocols (NSLPs) the working group will address during its

work.

Table of Contents

1. Introduction

Firewalls and Network Address Translators (NAT) have been both used throughout the Internet for many years and they will be present in future. Using firewalls brings security to networks and in times of IPv4 address depletion NATs virtually extend IP address space. In general, both types are obstacles to many applications, since they only allow specific applications to traverse them (i.e. HTTP traffic). Other applications, as for instance IP telephony, with more dynamic properties suffer from firewalls and NATs so that they don't work at all. Therefore, many applications cannot traverse any kind of firewall or NAT.

Several solutions to enable any application to traverse those boxes have been proposed and are currently used. Typically, application level gateways (ALG) have been integrated and so configuring firewalls and NATs dynamically. Another approach is middlebox communication (MIDCOM, currently under standardization at the IETF). In this approach firewall and NAT external ALGs configure them via the MIDCOM protocol [6]. Several other work around solutions are available as well. Anyway, all of these approaches introduce other problems that are hard to solve; one of them is dependency on topology issues.

NAT and firewall (NATFW) signaling share a property with Quality of Service (QoS) signaling, i.e. in both cases it is needed to reach any device on the data path that is involved in QoS or NATFW treatment of data packets. Currently, RSVP [13] is used for QoS signaling, but the conception of a new IP signaling protocol is under work in the Next Step of Signaling (NSIS) working group. This new signaling protocol is path-coupled, like RSVP is, and its primary use is QoS signaling, but NATFW signaling is considered as well.

This memo defines this NATFW path-coupled protocol. The NATFW signaling protocol is carried over the NSIS Network Transport Layer Protocol (NTLP, [3]) as NATFW NSIS Signaling Layer Protocol (NSLP). This NATFW NSLP is used to open pin-holes in firewalls and create NAT address mappings along the data path, so that subsequent data packets can traverse those devices.

Traversal of non NATFW NSLPs or the NTLP is out of scope of this document. Furthermore, only firewalls and NATs are considered in this document, any other device, for instance IPSec security gateway, is out of scope.

Section 2 describes the network environment for NATFW NSLP signaling and highlights the trust relationship/ authorization. Problems and challenges are listed in section 3, whereas a NSIS NAT handling

solution is described in section 4. Section 5 describes the protocol itself and section 6 gives some usage examples.

Readers are recommended to read the NSIS framework [1] and requirements documents beforehand [2]).

1.1 Terminology and Abbreviations

This document uses terms defined in [2]. Furthermore, these following terms are used:

- NSIS NAT Forwarding State: The term "NSIS NAT Forwarding State" in this context refers to a state used to forward the NSIS signaling message beyond the targeted destination address; that state is typically used when the NSIS Responder address is not known

- Sender-/Receiver Initiated Signaling

  Sender-initiated: NAT bindings and firewall rules are created immediately when the "path" message hits the NSIS nodes. With "path" message we refer to the signaling message traveling from the data sender towards the data receiver.

  Receiver-initiated: NAT bindings and firewall rules are created when the "reserve" message returns from the other end. With "reserve" message we refer to a signaling message on the reverse path, this means from the receiver to the sender (i.e. backwards routed).

  Note that these definitions have nothing to do with number of roundtrips, who performs authorization etc.

- Policy rule: In general, a policy rule is "a basic building block of a policy-based system. It is the binding of a set of actions to a set of conditions - where the conditions are evaluated to determine whether the actions are performed." [RFC3198]. In the context of NSIS NATFW NSLP the condition is a specification of a set of packets to which rules are applied. The set of actions always contains just a single element per rule, and is limited to either action "reserved" or action "enable".

- Firewall: A packet filtering device that matches packet against a set of policy rules and applies the actions. In the context of NSIS NATFW NSLP we refer to this device as firewall.

- Network Address Translator: Network Address Translation is a method by which IP addresses are mapped from one realm to another, in an attempt to provide transparent routing to hosts (see [8]).

Network Address Translator are devices that perform this method.

- Middlebox: from [11]: "A middlebox is defined as any intermediate device performing functions other than the normal, standard functions of an IP router on the datagram path between a source host and a destination host". The term middlebox in context of this document and in NSIS refers to firewalls and NATs only. Other types of middlebox are currently outside the scope.

- Security Gateway: IPsec based gateways.

- NSIS Initiator (NI): the signaling entity which makes the resource request, usually as a result of user application request.

- NSIS Responder (NR): the signaling entity that acts as the final destination for the signaling and can optionally interact with applications as well.

- NSIS Forwarder (NF): the signaling entity between an NI and NR which propagates NSIS signaling further through the network.

- Receiver (DR or R): the node in the network which is receiving the data packets of a flow.

- Sender (DS or S): the node in the network which is sending the data packets of a flow.

- NATFW NSLP session: Application layer flow of information for which some network control state information is to be manipulated or monitored (as defined in [1]). The control state for NATFW NSLP is NSLP state and associated policy rules at the middlebox.

- NSIS peer or peer: NSIS node with which a NSIS adjacency has been created as defined in [3].

- Edge NAT: By edge NAT we refer to the NAT device which is reachable from outside and has a globally routable IP address.

1.2 Middleboxes

The term middlebox raises different expectations about functionality provided by such a device. Middleboxes in the scope of this memo are firewalls that filter data packets against their set of filter rules and NATs that translate addresses from one address realm to another address realm. Other types of middleboxes, for instance QoS traffic shapers and security gateways, are out of scope.

The term NAT used in this document is placeholder for a range of different NAT flavors. We consider those types of NATs:

- traditional NAT (basic NAT and NAPT)
- Bi-directional NAT
- Twice-NAT
- Multihomed NAT

For a detailed discussion about each NAT type please see [7].

Both types of middleboxes use policy rules for decision on data packet treatment. Policy rules consist of a 5-tuple and an associated action; Data packets matching this 5-tuple experience the policy rule action. A 5-tuple consists of:

- Source IP address and port number
- Destination IP address and port number
- Transport protocol

Actions for firewalls are usually:

- Allow: forward data packet
- Deny: block data packet and discard it
- Other actions like logging, diverting, etc

Actions for NATs are (amongst many others):

- Change source IP address and port number to an global routeable IP address and port number.
- Change destination IP address and port number to a private IP address and port number.

The exact implementation of policy rules and mapping to firewall rule sets and NAT bindings or sessions at the middlebox is an implementation issue and thus out of scope of this document.

Some devices entitled as firewalls only accept traffic after cryptographic verification (i.e. IPsec protected data traffic). Particularly for network access scenarios either link layer or network layer data protection is common. Hence we do not address

these types of devices (referred as security gateways) since per-flow signaling is rather uncommon in this environment. For a discussion of network access authentication and associated scenarios the reader is referred to the PANA working group (see [22]).

Discovering security gateways, which was also mentioned as an application for NSIS signaling, for the purpose of executing an IKE to create an IPsec SA, is already solved without requiring NSIS.

In mobility scenarios an often experienced problem is the traversal of a security gateway at the edge of the corporate network. Network administrators often rely on the policy that only authenticated data traffic is allowed to enter the network. A problem statement for the traversal of these security gateways in the context of Mobile IP can be found at [21]).

Other proposals for path-coupled NAT and firewall traversal like RSVP and CASP are described in [23] and [24].

1.3 General Scenario for NATFW Traversal

The purpose of NSIS NATFW signaling is to enable any communication between endpoints across networks even in presence of middleboxes. It is expected that those middleboxes are configured in such a way that NSIS NATFW signaling messages itself are allowed to traverse them. NSIS NATFW NSLP signaling is used to install such policy rules in all middleboxes along the data path. Firewalls are configured to forward data packets matching the policy rule provided by the NSLP signaling. NATs are configured to translate data packets matching the policy rule provided by the NSLP signaling.

The basic high-level picture of NSIS usage is that endhosts are located behind middleboxes (NAT/FW in Figure 1). Applications located at these endhosts try to establish communication between them and use NSIS NATFW NSLP signaling to establish policy rules on a data path, which allows the said data to travel from the endhost to the endpoint unobstructed. The applications can somehow trigger middlebox traversal (e.g. via an API call) at the NSIS agent at the local host.

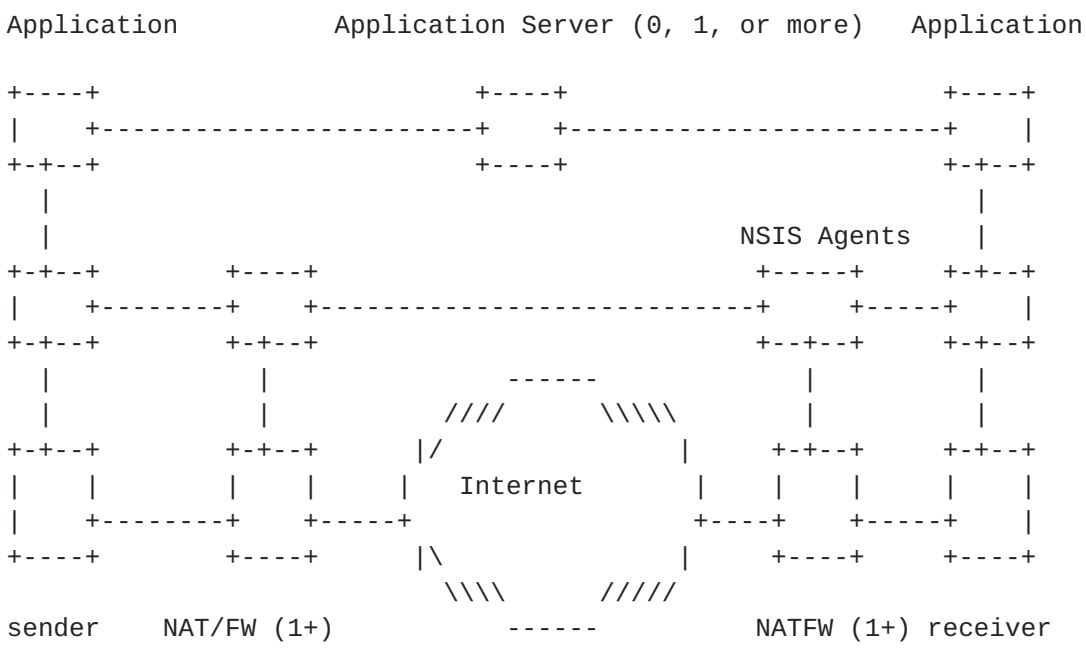


Figure 1: Generic View on NSIS in a NAT / Firewall case

For running NATFW signaling it is necessary that each firewall and each NAT involved in the signaling communication runs an NSIS NATFW agent. There might be several NATs and FWs in various possible combinations on a path between two hosts. The reader is referred to Section 2.1 where different scenarios are presented.

2. Network Environment

2.1 Network Scenarios for Protocol Functionality

This section introduces several scenarios for middleboxes in the Internet. Middleboxes are located in different locations, i.e. at Enterprise network borders, within enterprise networks, mobile phone network gateways, etc. In general, middleboxes are place more towards the edge of networks and less in network cores. Those middleboxes are not only either firewall or NAT, but any type of combination is possible. Thus, combined firewall and NATs are available.

NSIS NATFW NSLP signaling messages are sent by the NSIS initiator (NI) via the regular data path to the NSIS responder (NR). On the data path NATFW NSLP signaling messages reach different NSIS peers that have the NATFW NSLP implemented. Each NATFW NSLP node processes the signaling messages according to Section 5 and installs, if necessary, policy rules for subsequent data packets.

Each following section introduces a different scenario for a different set of middleboxes and their ordering within the topology. It is assumed that each middlebox implements the NSIS NATFW NSLP signaling protocol.

2.1.1 Firewall traversal

This section describes a scenario with firewalls only, no NATs are involved. Both end hosts are behind a firewall that are connected via the public Internet. Figure 2 shows the topology. The part labeled "public" is the Internet connection both firewalls.

FW: Firewall
NI: NSIS Initiator
NR: NSIS Responder

Figure 2: Firewall Traversal Scenario

Each firewall on-path must provide traversal service for NATFW NSLP

in order to permit the NSIS message to reach the other end host. All firewalls process NSIS signaling and establish appropriate policy rules, so that the required data packet flow can traverse them.

The difference between this scenario and the following is that only firewalls are on the path, but no NATs. This has specific implication concerning the used destination address for path-coupled signaling message sent by the NSIS initiator to an NSIS responder hosted behind a NAT.

2.1.2 NAT with two private Networks

Figure 3 shows a scenario with NATs at both ends of the network. Therefore, each application instance, NSIS initiator and NSIS responder are behind NATs. The outermost NAT at each side is connected to the public Internet. The NATs are labeled as MB (for middlebox), since those devices implement at least NAT-only, but can implement firewalling as well.

Only two middleboxes MB are shown in Figure 3 at each side, but in general more than one MB on each side must be considered.

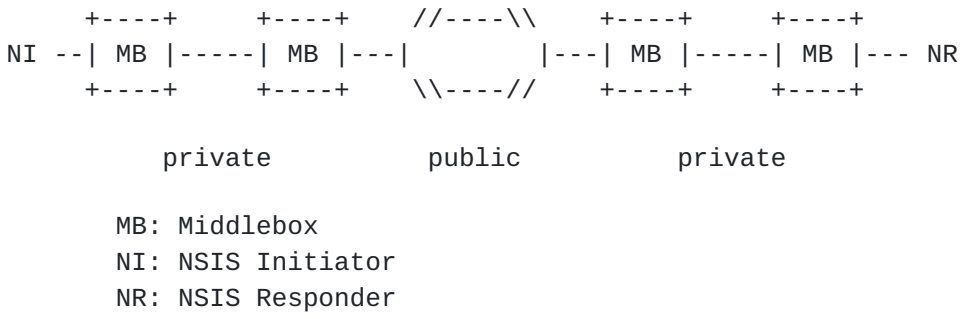


Figure 3: NAT with two private networks Scenario

Signaling traffic from NI to NR has to traverse all four middleboxes on the path and all four middleboxes must be configured properly to allow NSIS signaling to traverse. The NATFW signaling must configure all middleboxes and consider any address translation in further signaling. The sender (NI) has to know the IP address of the receiver (NR) in advance, otherwise he cannot send a single NSIS signaling message towards the responder. Note that this IP address is not the private IP address of the responder. Instead a NAT binding (including a public IP address) has to be obtained from the NAT which subsequently allows packets hitting the NAT to be forwarded to the receiver within the private address realm. This generally requires further support from an application layer protocol for the purpose of

discovering and exchanging information. The receiver might have a number of ways to learn its public IP address and port number and might need to signal this information to the sender using the application level signaling protocol.

2.1.3 NAT with private network on sender side

This scenario shows an application instance at the sending node which is behind one or more NATs (shown as MB). The receiver is located in the public Internet.

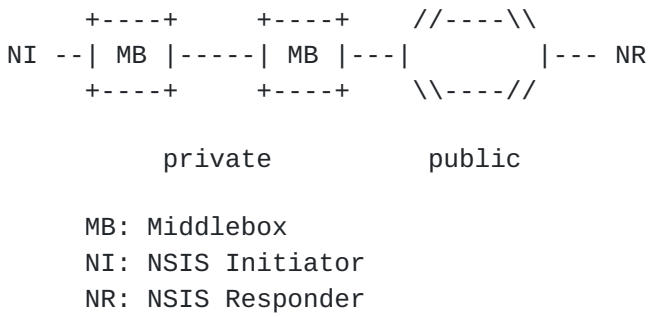


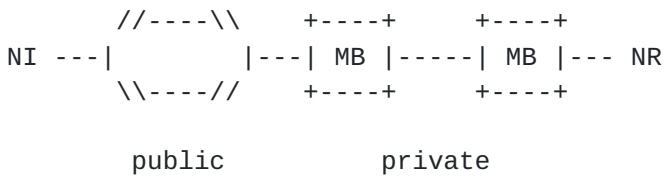
Figure 4: NAT with private network on sender scenario

The traffic from NI to NR has to traverse only middleboxes on the sender's side. The receiver has a public IP address. The NI sends its signaling message directly to the address of the NSIS responder. Middleboxes along the path intercept the signaling messages and configure the policy rules accordingly.

Note that the data sender does not necessarily know whether the receiver is behind a NAT or not, hence, it is the receiving side that has to detect the whether it is behind a NAT or not. As described in Section 4 NSIS can also provide help for this procedure.

2.1.4 NAT with private network on receiver side

The application instance receiving data is behind one or more NATs.



MB: Middlebox
NI: NSIS Initiator
NR: NSIS Responder

Figure 5: NAT with private network on receiver Scenario

First, the sender must determine the public IP address of the receiver. One possibility is that an application level protocol is used. In this case, the receiver must first fix its public IP addresses at the middlebox on its side. This information about IP address and port number could be signaled via an application level protocol to the actual sender directly or indirectly via a third party (e.g. proxy). In the scenario, this means the receiver has first to determine its public IP address (NAT binding) and register this address with a third party.

The NSIS initiator can start NSIS signaling after he has received information about the receiver's public IP address and port number.

2.1.5 Both End Hosts behind twice-NATs

This is a special case, where the main problem is to detect that both nodes are logically within the same address space, also behind a twice-NAT (see [7] for discussion about twice-NAT functionality). This scenario primarily addresses performance aspects.

Sender and receiver are both within a private address realm and potentially have overlapping IP addresses. Figure 6 shows the ordering of NATs. This is a common configuration in several networks, particularly after the merging of companies that have use the same address space, thus having overlapping addresses in many cases.

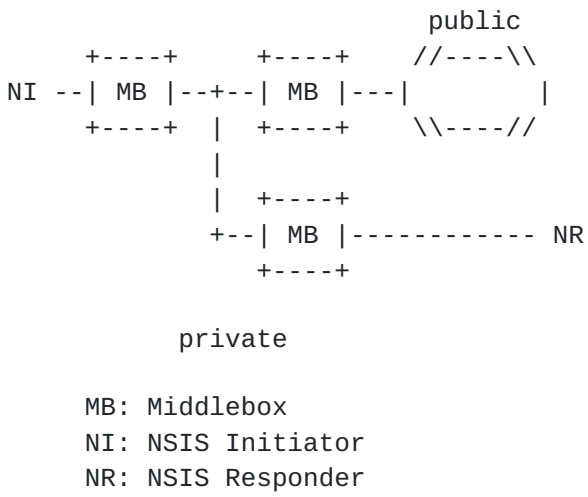


Figure 6: NAT to public, sender and receiver behind twice-NAT Scenario

The middleboxes shown in Figure 6 are twice-NATs, i.e. they map IP addresses and port numbers on both sides, at private and public interfaces.

This scenario requires assistance of application level entities, like DNS server. Those application level gateways must handle request that are based on symbolic names and configure the middleboxes so that data packets are correctly forwarded from NI to NR. The configuration of those middleboxes may require other middlebox communication protocols, like MIDCOM [6]. NSIS signaling is not required in the twice-NAT only case, since the middleboxes of type twice-NAT are configured by other means. Nevertheless, NSIS signaling might by useful when there are firewalls in between. In this case NSIS will not configure any policy rule at twice-NATs, but will configure policy rules at the intermediate firewalls. The NSIS signaling protocol must be at least robust enough to survive this scenario.

2.1.6 Both End Hosts behind same NAT

When NSIS initiator and NSIS responder are behind the same NAT (thus being in the same address realm, see Figure 7) , they are most likely not aware of this fact. As in Section 2.1.4 the NSIS responder must determine its public IP address in advance and transfer it to the NSIS initiator. Afterwards, the NSIS initiator can start sending the signaling messages to the responder's public IP address. During this process, a public IP address will be allocated for the NSIS initiator at the same middlebox as for the responder. Now, the NSIS signaling and the subsequent data packets will traverse the NAT twice: from

initiator to public IP address of responder (first time) and from public IP address of responder to responder (second time). This is the worst case, both sender and receiver obtain a public IP address at the NAT and the communication path is not optimal anymore.

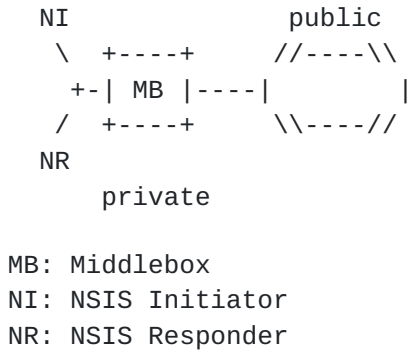


Figure 7: NAT to public, both host behind same NAT

NSIS NATFW signaling protocol should support mechanisms to detect such a scenario. The signaling should directly by exchanged between NI and NR without involving the middlebox.

2.1.7 IPv4/v6 NAT with two private networks

This scenario combines the usage case mentioned in Section 2.1.2 with the IPv4 to IPv6 transition scenario, i.e. using Network Address and Protocol Translators (NAT-PT, [10]).

The difference to the other scenarios is the use of IPv6 to IPv4 (and vice versa) address and protocol translation. Additionally, the base NTLP must take care of this case for its own functionality of forwarding messages between NSIS peers.

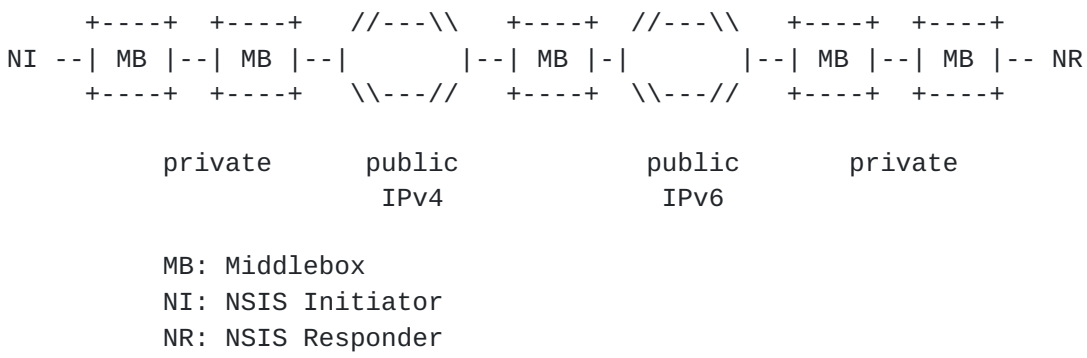


Figure 8: IPv4/v6 NAT with two private networks

This scenario needs the same type of application level support as described in Section 2.1.5 and so those issues of twice-NATs apply here as well.

2.2 Trust Relationship and Authorization

Trust relationships and authorization are very important for the protocol machinery. Trust and authorization are closely related to each other in the sense that a certain degree of trust is required to authorize a particular action. For any action (e.g. "create/delete / prolong policy rules" then authorization is very important due to the nature of middleboxes.

It is particularly not surprising that different degrees of required authorization in a QoS signaling environment and middlebox signaling exist. As elaborated in [19], establishment of a financial relationship is very important for QoS signaling, whereas for middlebox signaling is not directly of interest. For middlebox signaling a stronger or weaker degree of authorization might be needed.

Different trust relationships that appear in middlebox signaling environments are described in the subsequent sections. Peer-to-peer trust relationships are those, which are used in QoS signaling today and seem to be the simplest. However, there are reasons to believe that this is not the only type of trust relationship found in today's networks.

2.2.1 Peer-to-Peer Trust Relationship

Starting with the simplest scenario it is assumed that neighboring nodes trust each other. The required security association to authenticate and to protect a signaling message is either available (manual configuration) or dynamically established with the help of an

authentication and key exchange protocol. If nodes are located closely together it is assumed that security association establishment is easier than establishing it between far distant node. It is, however, difficult to describe this relationship generally due to the different usage scenarios and environments. Authorization heavily depends on the participating entities but for this scenario it is assumed that neighboring entities trust each other (at least for the purpose of policy rule creation, maintenance and deletion). Note that Figure 9 does not illustrate the trust relationship between the end host and the access network.

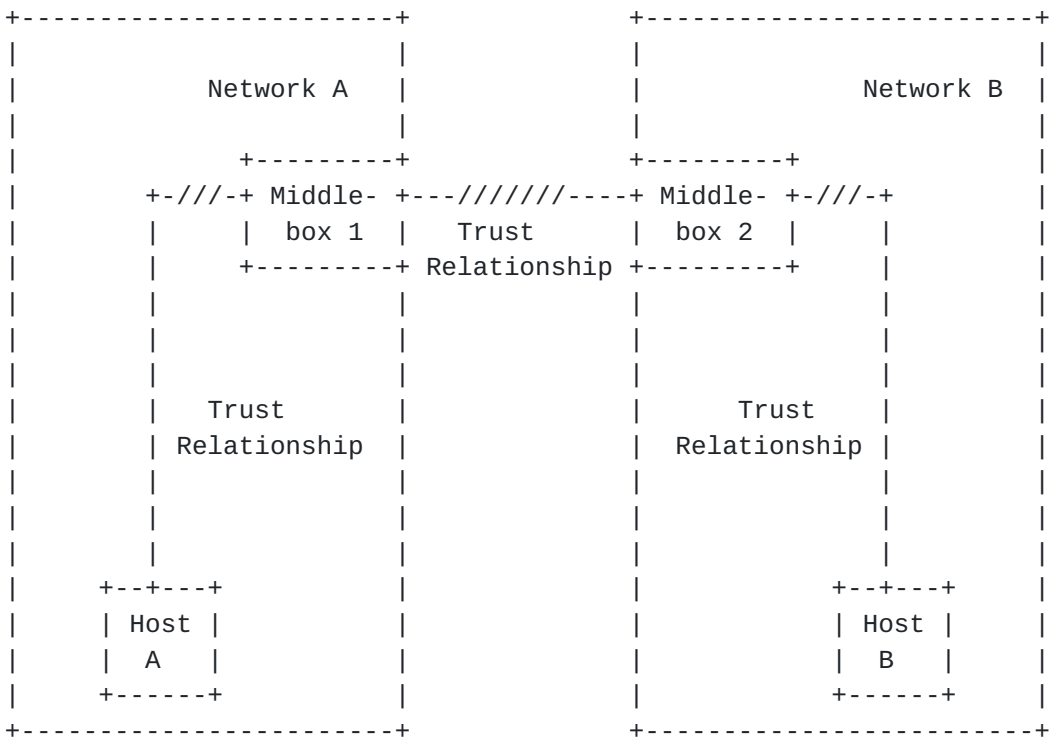


Figure 9: Peer-to-Peer Trust Relationship

2.2.2 Intra-Domain Trust Relationship

In larger corporations often more than one middlebox is used to protect different departments. In many cases the entire enterprise is controlled by a security department, which gives instructions to the department administrators. In such a scenario a peer-to-peer trust-relationship might be prevalent. Sometimes it might be necessary to preserve authentication and authorization information within the network. As a possible solution a centralized approach could be used whereby an interaction between the individual

middleboxes and a central entity (for example a policy decision point - PDP) takes place. As an alternative individual middleboxes could exchange the authorization decision to another middlebox within the same trust domain. Individual middleboxes within an administrative domain should exploit their trust relationship instead of requesting authentication and authorization of the signaling initiator again and again. Thereby complex protocol interaction is avoided. This provides both a performance improvement without a security disadvantage since a single administrative domain can be seen as a single entity. Figure 10 illustrates a network structure which uses a centralized entity.

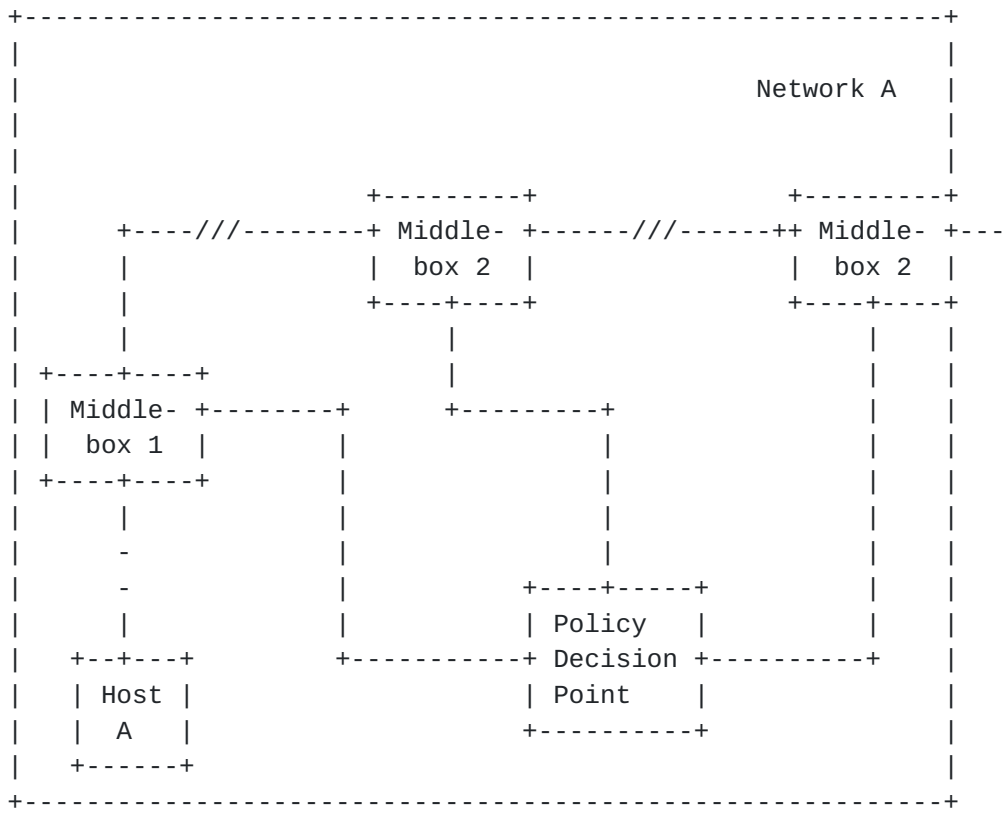


Figure 10: Intra-domain Trust Relationship

2.2.3 End-to-Middle Trust Relationship

In some scenarios a simple peer-to-peer trust relationship between participating nodes is not sufficient. Network B might require additional authorization of the signaling message initiator. If authentication and authorization information is not attached to the initial signaling message then the signaling message arriving at

Middlebox 2 would cause an error message to be created, which indicates the additional authorization requirement. In many cases the signaling message initiator is already aware of the additionally required authorization before the signaling message exchange is executed. Replay protection is a requirement for authentication to the non-neighboring middlebox which might be difficult to accomplish without adding additional roundtrips to the signaling protocol (e.g. by adding a challenge/response type of message exchange).

Figure 11 shows the slightly more complex trust relationships in this scenario.

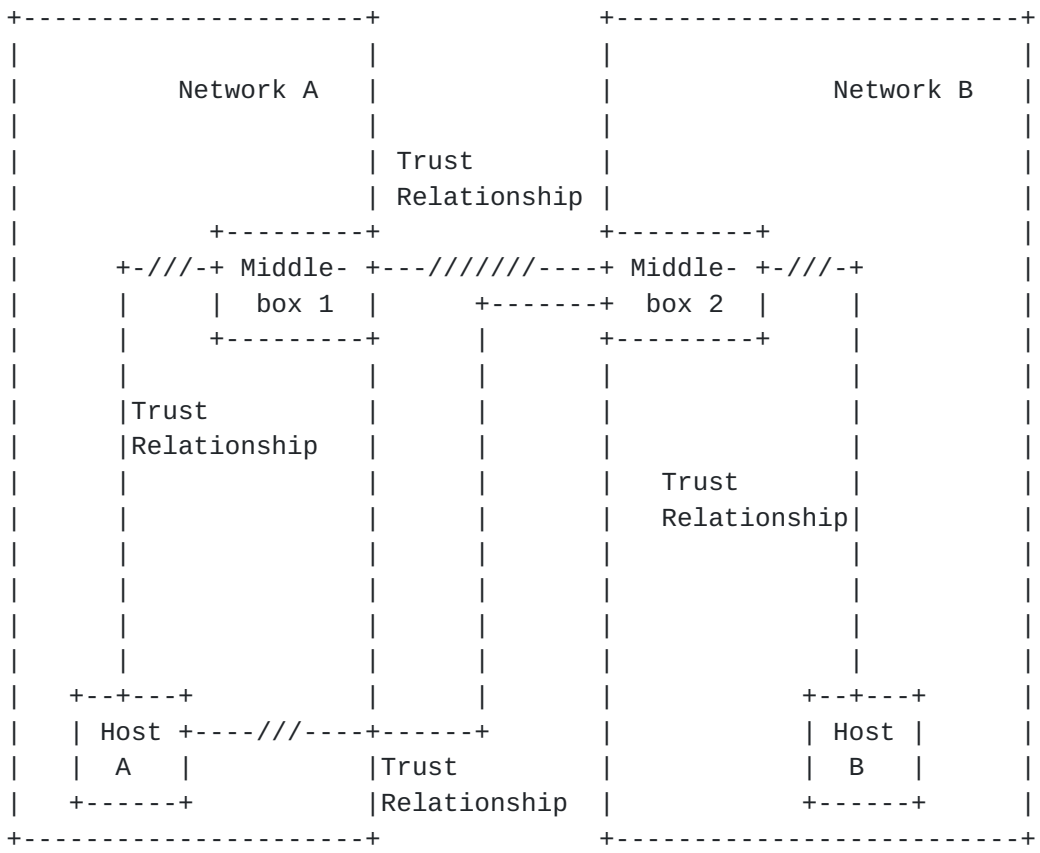


Figure 11: End-to-Middle Trust Relationship

3. Problems and Challenges

This section describes a number of problems which have to be addressed for NSIS NAT/Firewall. These might be also of relevance to other NSLP protocols.

3.1 Missing Network-to-Network Trust Relationship

Peer-to-peer trust relationship, as shown in Figure 9, is a very convenient assumption that allows simplified signaling message processing. However, it might not always be applicable, especially between two arbitrary access networks (over a core network where signaling messages are not interpreted). Possibly peer-to-peer trust relationship does not exist because of the large number of networks and the unwillingness of administrators to have other network operators to create holes in their firewalls without proper authorization. Hence in the following scenario we assume a somewhat different message processing and show three possible approaches to tackle the problem. None of these three approaches is without drawbacks or constraining assumptions.

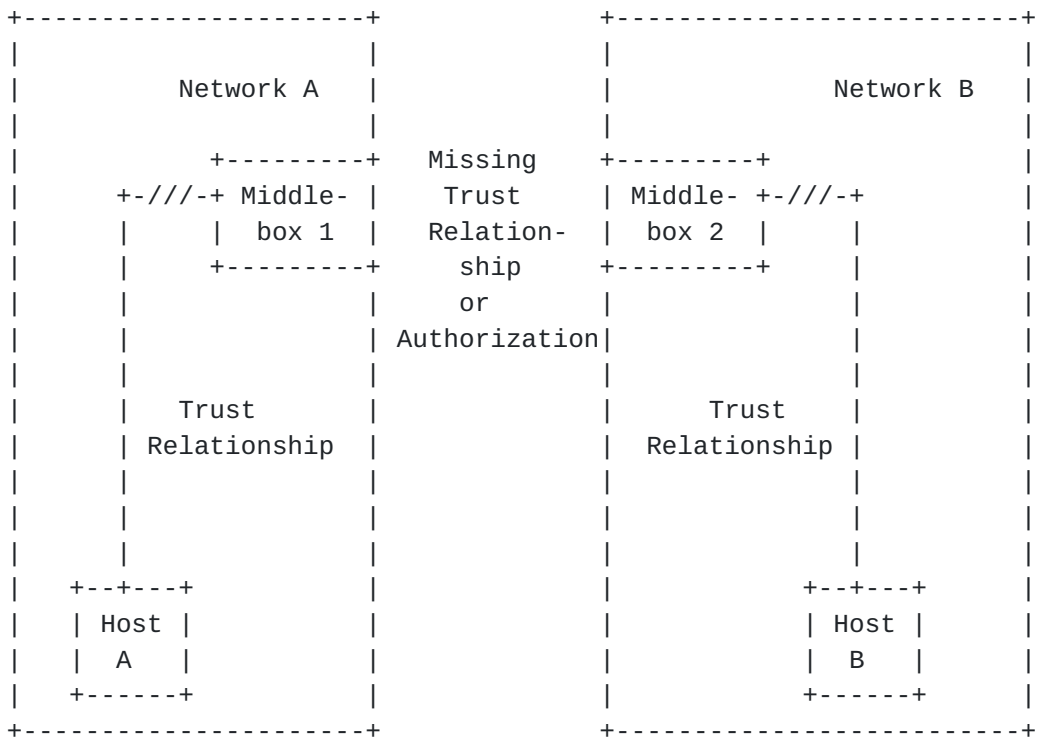


Figure 12: Missing Network-to-Network Trust Relationship

Figure 12 illustrates a problem whereby an external node is not

allowed to manipulate (create, delete, query, etc.) packet filters at a firewall. Opening pinholes is only allowed for internal nodes or with a certain authorization permission. Hence the solution alternatives in Section 4 focus on establishing the necessary trust with cooperation of internal nodes.

3.2 End-to-end significance

In the case of NAT/firewalls traversal, the NSIS signaling messages need to be sent all they way from the DS and DR or vice versa. This is so because a middlebox does not know whether the remaining path to the destination is clear of potentially obstructing middleboxes or not.

3.3 Relationship with routing

The data path is following the "normal" routes. The NAT/FW devices along the data path are those providing the service. In this case the service is something like "open a pinhole" or even more general "allow for connectivity between two communication partners". The benefit of using path-coupled signaling is that the NSIS NATFW NSLP does not need to determine what middleboxes or in what order the data flow will go through.

Creating NAT bindings modifies the path of data packets between two end points. Without NATs involved, packets flow from endhost to endhost following the path given by the routing. With NATs involved, this end-to-end flow is not directly possible, because of separated address realms. Thus, data packets flow towards the external IP address at a NAT (external IP address may be a public IP address). Other NSIS NSLPs, for instance QoS NSLP, which do not interfere with routing - instead they only follow the path of the data packets.

3.4 Dynamic state installation and maintenance

For NAT/Firewall traversal, the lifetime of a NAT binding or a packet filter is maintained through periodic refresh. For short-lived flows, having unpredictable filters, signaling for dynamically policy rules is preferable as opposed to statically configured policy rules requested for long duration in time.

For static state other mechanisms than an NSIS signaling protocol might be preferable; such mechanisms would include a management protocols such as SNMP or command line interfaces.

3.5 Affected Parts of the Network

NATs and Firewalls are usually located at the edge of the network,

whereby other signaling applications affect all nodes along the path. One typical example is QoS signaling where all networks along the path must provide QoS in order to achieve true end-to-end QoS. In the NAT/Firewall case, only some of the domains/nodes are affected (typically access networks), whereas most parts of the networks and nodes are unaffected (e.g. the core network).

This fact raises some questions. Should an NSIS NTLP node intercept every signaling message independently of the upper layer signaling application or should it be possible to make the discovery procedure more intelligent to skip nodes. These questions are also related to the question whether NSIS NAT/FW should be combined with other NSIS signaling applications.

3.6 NSIS backward compatibility with NSIS unaware NAT and Firewalls

Backward compatibility is a key for NSIS deployments, as such the NSIS protocol suite should be sufficiently robust to allow traversal of none NSIS aware routers (QoS gates, Firewalls, NATs, etc).

NSIS NATFW NSLP's backward compatibility issues is different than the NSIS QoS NSLP backward compatibility issues, where an NSIS unaware QoS gate will simply forward the QoS NSLP message. An NSIS unaware firewall rejects NSIS messages, since firewalls typically implement the policy "default to deny".

The NSIS backward compatibility support on none NSIS aware firewall would typically consist of configuring a static policy rule that allows the forwarding of the NSIS protocol messages (either protocol type if raw transport mode is used or transport port number in case a transport protocol is used).

For NATs backward compatibility is more problematic since signaling messages are forwarded (at least in one direction), but with a changed IP address and changed port numbers. The content of the NSIS signaling message is, however, unchanged. This can lead to unexpected results, both due to embedded unchanged local scoped addresses and none NSIS aware firewalls configured with specific policy rules allowing forwarding of the NSIS protocol (case of transport protocols are used for the NTLP). NSIS unaware NATs must be detected to maintain a well known deterministic mode of operation for all the involved NSIS entities. Such a "legacy" NAT detection procedure can be done during the NSIS discover procedure itself.

Based on experience it was discovered that routers unaware of the Router Alert IP option [RFC 2113] discarded packets, this is certainly a problem for NSIS signaling.

3.7 Authentication and Authorization

For both types of middleboxes, firewall and NAT security is a strong requirement. Authentication and authorization means must be provided.

For NATFW signaling applications it is partially not possible to do authentication and authorization based on IP addresses. Since NATs change IP addresses, such a address based authentication and authorization scheme would fail.

3.8 Directional Properties

There two directional properties that need to be addressed by the NATFW NSLP:

- Directionality of the data
- Directionality of NSLP signaling

Both properties are relevant to NATFW NSLP aware NATs and Firewalls.

With regards to NSLP signaling directionality: As stated in the previous sections, the authentication and authorization of NSLP signaling messages received from hosts within the same trust domain (typically from hosts located within the security perimeter delimited by firewalls) is normally simpler than received messages sent by hosts located in different trust domains.

The way NSIS signaling messages enters the NSIS agent of a firewall (see Figure 2) might be important, because different policies might apply for authentication and admission control.

Hosts deployed within the secured network perimeter delimited by a firewall, are protected from hosts deployed outside the secured network perimeter, hence by nature the firewall has more restrictions on flows triggered from hosts deployed outside the security perimeter.

3.9 Routing Asymmetry

Routing asymmetry [20] is a general problem for path-coupled signaling, especially when installed states on NSIS forwarders are related to bi-directional flows.

Path state, on an NSIS forwarder, including the next NSIS hop (for packets sent from the NR to NI), is used to handle routing asymmetry for NSIS messages, but not for data flows (i.e. no route pinning for

data flows).

Similarly to path-coupled QoS signaling, middlebox signaling also has to be aware of the routing asymmetry when bi-directional flows relevant states need to be installed on NSIS aware nodes, although the routing asymmetry might not be significant within the local networks where firewalls are typically located. For signaling NAT bindings this issue comes with a different flavor since an established NAT binding changes the path of the data packets. Hence a data receiver might still be able to send NSIS signaling messages to create a NAT binding, although they travel the previously "wrong" path.

3.10 Addressing

A more general problem of NATs is the addressing of the end-point. NSIS signaling message have to be addressed to the other end host to follow data packets subsequently sent. Therefore, a public IP address of the receiver has to be known prior to sending an NSIS message. When NSIS signaling messages contain IP addresses of the sender and the receiver in the signaling message payloads, then an NSIS agent must modify them. This is one of the cases, where a NSIS aware NATs is also helpful for other types of signaling applications e.g. QoS signaling.

3.11 NTLP/NSLP NAT Support

It must be possible for NSIS NATs along the path to change NTLP and/or NSLP message payloads , which carry IP address and port information. This functionality includes the support of providing mid-session and mid-path modification of these payloads. As a consequence these payloads must not be reordered, integrity protected and/or encrypted in a non peer-to-peer fashion (e.g. end-to-middle, end-to-end protection). Ideally these mutable payloads must be marked (e.g. a protected flag) to assist NATs in their effort of adjusting these payloads.

3.12 Route changes

The effect of route changes are more severe than in other signaling applications since a firewall pinhole and NAT binding is needed before further communication can take place. This is true for both NSIS signaling and for subsequent data traffic. If a route changes and NSIS signaling messages do not configure NSIS NATs and firewalls along the new path then the communication is temporarily interrupted. This is naturally a big problem for networks where routes frequently change e.g. ad-hoc networks or in case of fast mobility. In these cases state refresh messages have to provide a mechanism for fast

reaction.

3.13 Combining Middlebox and QoS signaling

In many cases, middlebox and QoS signaling has to be combined at least logically. Hence, it was suggested to combine them into a single signaling message or to tie them together with the help of some sort of data connection identifier, later on referred as Session ID. This, however, has some disadvantages such as:

- NAT/FW NSLP signaling affects a much small number of NSIS nodes along the path (for example compared to the QoS signaling).

- NAT/FW signaling might show different signaling patterns (e.g. required end-to-middle communication).

- The refresh interval is likely to be different.

- The number of error cases increase as different signaling applications are combined into a single message. The combination of error cases has to be considered.

3.14 Difference between sender- and receiver-initiated signaling

For NAT/FW signaling there seems to be little difference between sender- and receiver-initiated signaling messages. Some other characteristics of QoS signaling protocols are not applicable (e.g. the adspec object) to the NAT/FW context. It seems that a full roundtrip is always required if the protocol aims to be generic enough.

3.15 Inability to know the scenario

In Section 2.1 a number of different scenarios are presented. Data receiver and sender may be located behind zero, one, or more firewalls and NATs. Depending on the scenario, different signaling approaches have to be taken. For instance, data receiver with no NAT and firewall can receive any sort of data and signaling without any further action. Data receivers behind a NAT must first obtain a public IP address before any signaling can happen. The scenario might even change over time with moving networks, ad-hoc networks or with mobility.

NSIS signaling must assume the worst case and cannot put responsibility to the user to know which scenario is currently applicable. As a result, it might be necessary to perform a "discovery" periodically such that the NSIS agent at the end host has enough information to decide which scenario is currently applicable.

This additional messaging, which might not be necessary in all cases, requires additional performance, bandwidth and adds complexity. Additional, information by the user can provide information to assist this "discovery" process but cannot replace it.

4. NSIS NAT Handling Solution

This section describes a mechanism for allowing NSIS signaling messages to travel end-to-end in the presence of NATs at the receiving side. This requires to establish state information at the NSIS-aware NAT device.

Note: The discussed mechanism only creates state relevant for NSIS message handling. It does not create NAT bindings for data traffic.

4.1 Problem Description

NSIS signaling messages follow the data path from the data sender to the data receiver. To provide this property of being path-coupled a discovery process sends signaling messages along the same route as taken by subsequent data packets. The NSIS messages are directed to a particular destination IP address and hence the destination address needs to be known in advance before NSIS signaling can start.

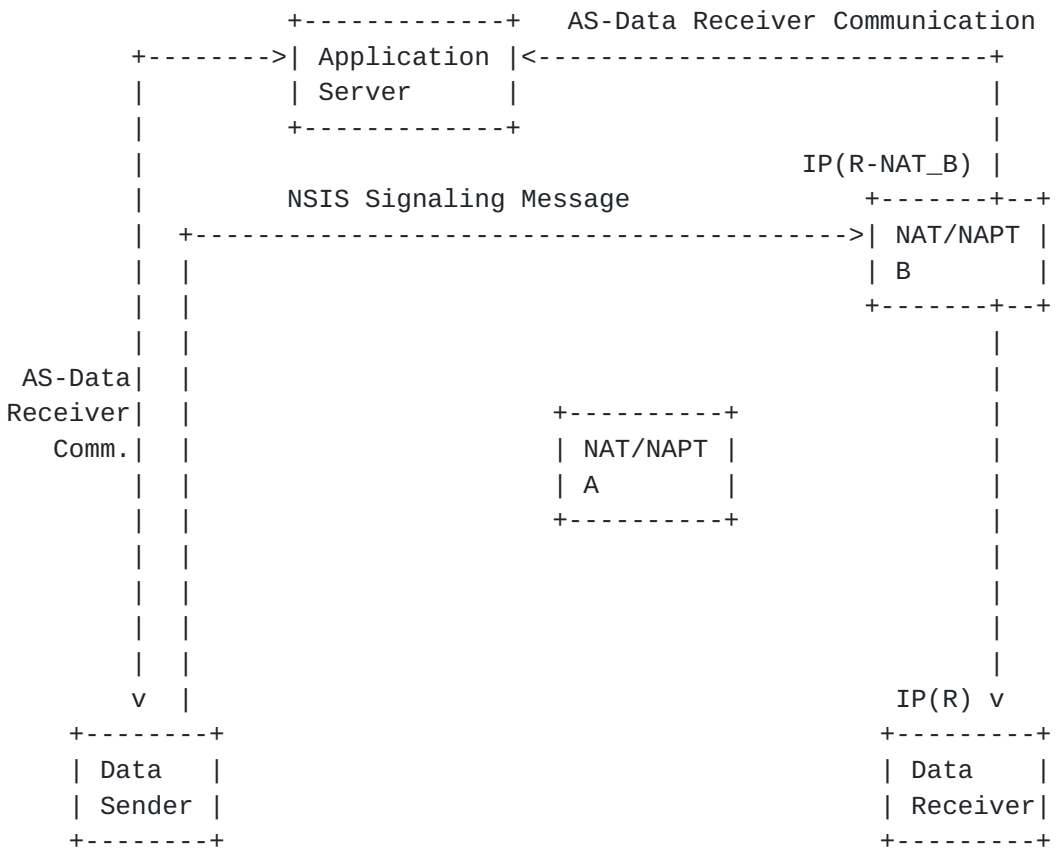


Figure 13: The Data Receiver behind NAT problem

Figure 13 describes a typical message communication in a peer-to-peer networking environment whereby the two end points learn of each others existence with the help of a third party (referred as Application Server). The communication with the application server and the two end points (data sender and data receivers) serves a number of functions. As one of the most important functions it enables the two end hosts to learn the IP address of each other.

Without the proposed mechanism it would not be possible to establish a NAT binding end-to-end in all scenarios.

Some sort of communication between the end hosts and a third party is typically necessary (independently of NSIS). NSIS signaling messages cannot be used to communicate application level relevant end point identifiers (in the generic case at least) as a replacement for the communication with the application server.

If the data receiver is behind a NAT then an NSIS signaling message will be addressed to the IP address allocated at the NAT (if there

was one allocated). If no corresponding NSIS NAT Forwarding State at NAT/NAPT B exists (binding IP(R-NAT B) <-> IP(R)) then the signaling message will terminate at the NAT device (most likely without proper response message). The signaling message transmitted by the data sender cannot install the NAT binding or NSIS NAT Forwarding State "on-the-fly" since this would assume that the data sender knows the topology at the data receiver side (i.e. the number and the arrangement of the NAT and the private IP address(es) of the data receiver). The primary goal of path-coupled middlebox communication was not to force end hosts to have this type of topology knowledge.

A number of solutions exist to allow nodes behind a NAT to establish a NAT binding to allow the receiver to receive IP packets. These solutions can at best be labeled as hacks (see [NATP2P]) and they have their drawbacks:

- They assume a certain behavior of NAT boxes.
- They work in some environments whereas in others they do not properly function.
- They only allow NAT bindings for UDP traffic to be established.
- They often fail.

Some other solutions assume that both nodes are registered in the DNS directory (see [12]).

The requirements for an NSIS solution are two-fold:

1. NSIS signaling messages must be able to travel end-to-end (between data sender and data receiver) - if desired. This is important for a number of NSIS NSLPs
2. NSIS relies on a generic solution which works in all scenarios (see section 5 of [26]).

Since the NSIS signaling messages are intercepted at each NSIS device, the NAT solution depends on the properties of the NTLP. In particular, multiplexing capability is important. Two possible options are feasible:

1. Multiplexing with the help of transport layer information (i.e. port information)
2. Multiplexing at the NSIS application layer (e.g. based on session identifier)

We describe the second approach although we believe that alternatives are possible.

Enough information has to be available to convert IP address information of an incoming signaling message to different IP addresses of an outgoing NSIS message. Finally the signaling message must reach the data receiver.

It seems that the session identifier can be used to associate state information of the two independent signaling exchanges. The two exchanges (as described in Section 4.2) are:

1. Signaling exchange from the data receiver (NR) to the NAT(s)

2. End-to-end NSIS signaling message exchange from the NI to the NR.

If the session identifier is used for this purpose then it is necessary to communicate the session ID from the data receiver (NR) to the NI. Communicating the IP address information instead (as an alternative solution approach) is easier since this functionality is already provided by SIP whereas securely exchanging (e.g. confidentiality protected) the Session Identifier is not available.

4.2 Solution Overview

The data receiver starts to signal an NSIS Create-NAT-Binding message into the "wrong direction". By "wrong" we refer to the usual behavior of path-coupled signaling where the data sender starts signaling in order to tackle with routing asymmetry. The data receiver would typically return signaling messages to the data sender in the reverse direction by utilizing state created at nodes along the path (i.e. to reverse route signaling messages). The concept of path-coupled or path-decoupled signaling is, however, no relevant for this special type of signaling communication. In case of establishing NAT bindings (and NSIS NAT Forwarding State) the direction does not matter since routing is modified. Subsequent NSIS messages (and also data traffic) will travel through the same NAT boxes.

The proposed solution requires two NSIS signaling messages:

1. Reserve External Address Request

2. Reserve External Address Acknowledgment

The semantics of the two messages will be described in detail in this section.

The data receiver sends a Reserve External Address NSIS signaling message into the local network (before the data sender starts NSIS signaling). In Section Section 4.2.1 we will discuss where to address this signaling message (i.e. which destination IP address to use).

The signaling message creates NSIS NAT Forwarding State at intermediate NSIS NAT node(s). Furthermore it has to be ensured that the edge NAT device is discovered as part of this process. The end host cannot be assumed to know this device - instead the NAT box itself is assumed to know that it has such a capability. Forwarding of the Reserve External Address NSIS message beyond this entity is not necessary, and should be prohibited as it provides information on internal hosts capabilities.

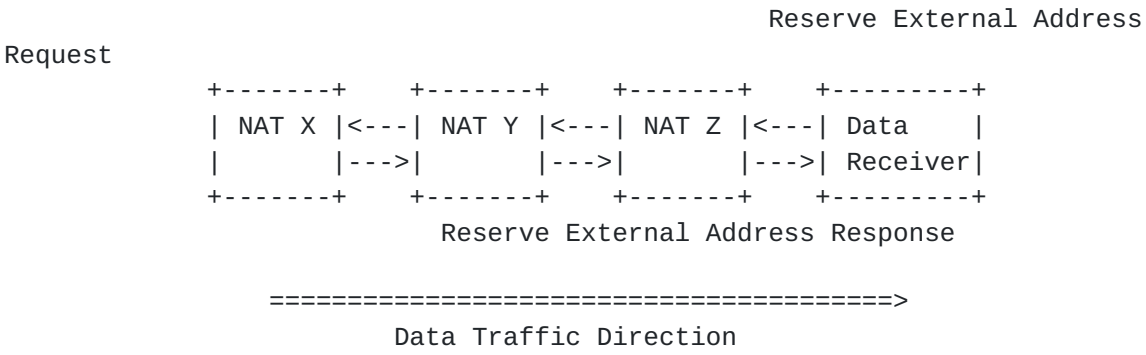


Figure 14: Reserve External Address NSIS Signaling Message

The goal of this signaling message exchange is:

- to create one (or more) NAT binding(s)
- to allow the data receiver to learn its global routable IP address (for communication with NSIS)
- not to require the data receiver to learn topology information.

Figure 14 shows a number of NAT devices at the data receivers network side. NSIS NAT Forwarding State is established at these network elements.

The Reserve External Address Request message triggers the state creation and the discovery. The message carries information where the sender expects incoming NSIS signaling messages.

The Reserve External Address Response message confirms the state

creation and allows to return information about the NATs and the topology to the end host (for informational purposes). As a result the end host will learn the public IP address which can be used by the data sender to address NSIS signaling messages.

4.2.1 Destination IP address Selection

The Reserve External Address Request message has to be addressed to a specific destination IP address. Since there is no natural candidate a few alternatives might be considered. The discussed options refer to entities of Figure 13

Possible options are:

1. Public IP address of the data sender

2. Public IP address of the data receiver (allocated at NAT B)

3. IP address at the Application Server

Actually, there is no "correct" answer to this question and from a theoretical point of view it does not really matter as long as Host A learns an IP address where he has to send the NSIS signaling message. From a performance point of view there is, however, a difference since it would be desirable to create an "optimal" routing path.

1. Public IP address of the data sender:

    * Assumption:

        + The data receiver already learned the IP address of the data sender (e.g. via a third party).

    * Problems:

        + The data sender might also be behind a NAT. In this case the public IP address of the data receiver is the IP address allocated at this NAT.

        + Due to routing asymmetry it might be possible that the routes taken by a) the data sender and the application server b) the data sender and NAT B might be different. As a consequence it might be necessary to advertise a new (and different) external IP address with SIP after using NSIS to establish a NAT binding.

2. Public IP address of the data receiver (allocated at NAT B):

- Assumption:
  - The data receiver already learned his externally visible IP address (e.g. based on the third party communication).
- Problems:
  - Communication with a third party is required.

3. IP address at the Application Server:

- Assumption:
  - An application server (or a different third party) is available.
- Problems:
  - If the NSIS signaling message is not terminated at the NAT of the local network then an NSIS unaware application server might discard the message.
  - Routing might not be optimal since the route between a) the data receiver and the application server b) the data receiver and the data sender might be different.

5. Protocol Description

5.1 Basic protocol overview

The NSIS Signaling Layer Protocol (NLSP) for NAT and FW traversal is carried over the NSIS Transport Layer Protocol (NTLP) defined in [3]. NATFW NSLP messages are initiated by the NSIS initiator, (NI) handled by NSIS forwarders (NF) and finally processed by the NSIS responder (NR). It is required that at least NI and NR implement this NSLP, intermediate NF only implement this NSLP when they provide middlebox functions. Forwarders that do not have any NATFW NSLP functions just forward these messages; those forwarders implement NTLP and one or more other NSLPs.

A Data Sender (DS) that intents to send data to a Data Receiver (DR) must start its NATFW NSLP signaling. So the NI at the data sender (DR) starts NSLP signaling towards the address of data receiver DR (see Figure 15).

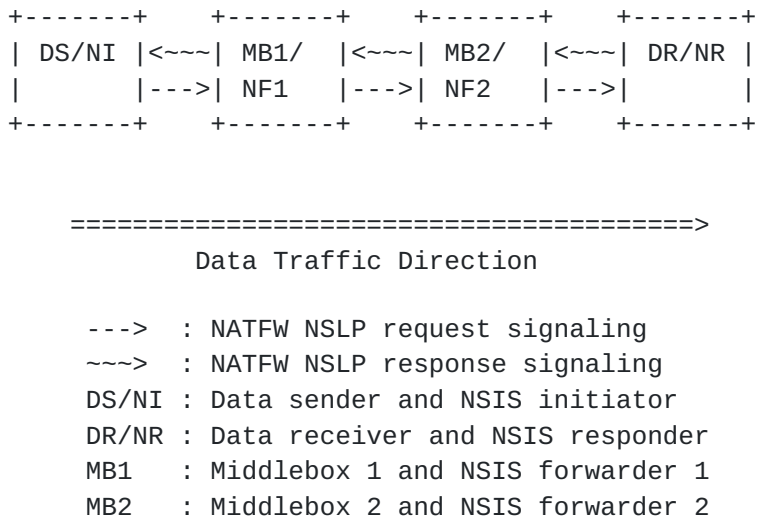


Figure 15: General NSIS signaling

The NSLP request messages are processed each time a NF with NATFW NSLP support is passed. Those nodes process the message, check local policies for authorization and authentication, possibly create policy rules, and forward the signaling message to the next NSIS node. The request message is forwarded until it reaches the NSIS responder. NSIS responders will check received messages and process those if applicable. NSIS responders generate response messages and sent them

back to the NI via the same chain of NFs. The response message is processed at each NI forwarder implementing NATFW NSLP. The Data Sender can start sending its data flow to the Data Receiver, when the signaling was successful, meaning that NI has received a successful response.

In general, NATFW NSLP signaling follows the data path from DS to DR. This enables communication between both hosts for scenarios with only firewalls on the data path or NATs on sender side. For scenarios with NATs on the receiver side certain problems arise.

When Data receiver (DR) and Data Sender (DS) are located in different address realms and DR is behind a NAT, DS cannot signal to DR. DR is not reachable from DS and thus no NATFW signaling can be sent to DR's address. Therefore, DR must first fix a address at a NAT that is reachable for DS, for instance DR must determine its public IP address. Once DR has fixed a public address it forwards this to DS via a separate mechanism, which may be application level signaling like SIP. This application level signaling may involve third parties that assist in exchanging this information. This separate mechanism is out of scope of NATFW NSLP.

NATFW NSLP signaling supports this public address fixing with this mechanism:

- First, DR fixes a public address by signaling on the reverse path (DR towards DS) and thus making itself available to other hosts. This process of fixing public addresses is called reservation. This way DR reserves publicly reachable addresses and ports.

- Second, DS is signaling directly to DR, creating policy rules at middleboxes. Note, that the reservation mode will usually make reservations only, which will be "activated" by the signaling from DS towards DR. The first mode is detailed in the Section 4

The protocol is intended to work on a soft-state basis. This means, that whatever state is installed or reserved on a middlebox, will expire, and thus be de-installed/ forgotten after a certain period of time. To prevent this the involved boxes will have to specifically request a session prolongation. An explicit NATFW NSLP state deletion message is also provided by the protocol.

Middleboxes should report back in case of error, so that appropriate measures and debugging can be performed.

The next sections define the NATFW NSLP message types and formats, protocol operations, and policy rule operations.

5.2 NATFW NSLP Header

The NATFW NSLP header is common to all messages and follows directly the NTLP header. A NSLP node can distinguish based on this header whether a request or response message is passed in the packet. It is followed by a series of objects.

The NSIS NATFW NSLP header contains:

- version: NSIS NATFW NSLP protocol version number.
- header_len: length of the NSLP payload in bytes, including NSLP header
- obj_count: number of objects that follow after the NSIS header.
- message type: The type of the NSLP message, request or response. Sub-types are encoded in this field as well.

Message type indicates whether the NSLP packet is a request or a response. For request messages, four sub-types are defined:

- Create Session
- Prolong Session
- Delete Session
- Reserve Session

For response messages, three sub-types are defined:

- Return External Address
- Path Succeeded
- Error

The next sections define which objects are included in which message type. For each message type the allowed combination of objects is described.

5.3 NATFW NSLP Objects

5.3.1 NATFW NSLP Object Header

NATFW NSLP objects carry the actual information about policy rules, lifetimes and error conditions. All objects share the same object

header. An object header is followed by the object data, whereas the format of the object data depends on the object type. A NATFW NSLP payload may contain several objects.

The object header has the following format:

- obj_len: total length of the object, including object header
- obj_type: type of NATFW NSLP object. Identifies the data that follows.

For the moment four object types are defined in the next sections. Other objects can be defined later on. These four objects each describe a message request type.

5.3.2 NATFW Session ID Object

The NATFW Session ID is the handle to the NATFW session at a particular NSIS node. It is randomly generated by the NSIS initiator.

5.3.3 Lifetime Object

The lifetime object indicates the lifetime of a NATFW NSLP session. The real lifetime at a NSIS peer is the current time plus the lifetime value of this object.

5.3.4 Policy Rule Object

The policy rule objects contains the flow information for the data traffic from DS to DR. The information contained in this object will change as soon as NATs are involved.

The policy rule object has these fields:

- Source address: The IP address where the data will come from. If it is DS sending data to DR, the source address is either DS or the closest NAT in the route from DS to the middlebox that gets the packet; That is, the address where each middlebox will see the packet come from.
- Destination address: The IP address where the data is headed. If it is DS sending data to DR, the destination address is either DR or the public address DR reserved itself.
- Protocol: The protocol carried in the IP data packet. Currently TCP, UDP and IP is defined.

- Source Port: The transport layer port the data will come from
- Destination Port: The transport layer port the data will go to.
- IPv flow label: The IPv6 flow label (Editor's note: needs further in-depth discussion).

Note: you might want to leave the source address or port set to ANY, to accept any source address port. This makes the pinhole not so pin like, but might be necessary at the integration with certain NAT/FW types. Whether this loose pinhole is authorized or not by the middlebox, is a policy decision based on the middlebox configuration.

5.3.5 External Address Object

This object contains the reserved external address and if applicable port number.

The object has these fields:

- External IP address: The reserved external IP address at the NAT.
- External port number: The reserved external port number at the NAT.

5.4 Request Message Formats

This section defines the message types and their format for the NATFW NSLP. Note, that at the moment of writing this document, no final decision has been reached on the details of the NTLP. Thus, message types and formats may change in future revisions of this document.

Currently, the NATFW NSLP header and 4 request messages are defined. Furthermore, three response message types are defined. All those messages are explained in this chapter.

The NATFW payload of a NSIS NTLP packet consists of a NATFW NSLP header that is common to all request, response and error messages. Several NATFW NSLP objects follow the NSLP header, depending on the message type.

NOTE: Any bit-level definition of messages and headers are to be done in future revision of this memo. Furthermore, any order of object fields below is not mandating their order in the actual bit-level definition.

5.4.1 Create Session

The create session request message is used to create policy rules on middleboxes. Middleboxes receiving this message type will, if authenticated and authorized, enable the requested policy rules, so that data packets of the specified data flow can traverse.

The create session messages carries these objects:

- Session ID object: A newly generated session ID
- Policy Rule object: The description of the data flow
- Lifetime Object: A request lifetime for this NSIS NATFW NSLP session

5.4.2 Prolong Session

The prolong session request message is used to extend the lifetime of a NATFW NSLP session. The NSIS initiator requests a certain lifetime extension.

The prolong session message carries these objects:

- Session ID object: Session to be prolonged
- Lifetime Object: Requested new lifetime

5.4.3 Delete Session

The delete session request message is used to delete NATFW NSLP session.

The delete session object carries this object:

- Session ID: The session to be deleted.

5.4.4 Reserve External Address

The reserve external address request message is used in the case that a Data Receiver (DR) is located behind a NAT. The DR needs to received data and so uses this request message to reserve an external IP address at a NAT.

The reserve external address message carries these objects:

- Session ID: The session ID for the reservation. Note that this session ID is only valid for the reservation. Create messages using the reservation will use their own generate session ID.

- Lifetime: The lifetime of the reservation

- Policy Rule: In the reserve external address message the policy rule object must be set accordingly:

  Source address: The source address of the data flow. This is the destination of the NATFW reserve address packet. The way of NSLP signaling is in the reverse way of the data flow.

  Source port: The source port of the data flow.

  Destination address: The internal IP address to where data flow will be destined. This is the source address of the NATFW reserve address packet.

  Destination port: The destination port of the data flow

  Protocol: Expected protocol

The direction of NSIS NATFW NSLP signaling is reverse to the reserved data flow. The source address of the expected data flow is the destination of the signaling. Vice versa, the destination address of the expected data flow is the source of the signaling (see section Section 4).

Note that no state, be it a firewall rule or a NAT binding, is installed as a result of this message. The state is only remembered, and might be later installed by a create message.

5.5 Response Messages

The following messages are responses messages that are generate either by any NF or NR. Currently, three different types of request messages are defined.

5.5.1 Return External Address Response

The return external address response message is sent as a successful reply to a reserve external address request.

Return external address message contains these objects:

- Session ID: The session this packet is replying to.

- External address object: Contains reserved external IP address and port number
- Lifetime: The minimum granted lifetime for this reservation.

5.5.2 Path Succeeded Response

The path succeeded response message is sent as a successful reply to a create session request.

The Path succeeded response message contains these objects:

- Session ID: The session ID for which a path was successfully installed
- Lifetime: The minimum granted lifetime of this session

5.5.3 Error Response Messages

Any NATFW NSLP error occurring at NF or NR is reported via the error response message towards the NI.

The error message contains these objects:

- Session ID: The session id of the object that generated the error
- Error code: The error to report.

Possible error code classes are:

- Policy rule errors
- Authentication and Authorization errors
- NATFW NSLP protocol errors

5.6 Protocol Operations

This section defines the message flow and protocol operation for all message types

5.6.1 Message Handling Overview

When a NSIS NATFW peer receives an NSIS message, it might take an action based on the message type, the nature of the middlebox

function, its configuration and local security policies.

As a summary, here's the behavior of the boxes, depending on message type and configuration parameters:

	NAT	FW	NAT+FW	DS	DR
reserve	5	-	5	+	+
ret_ext_addr	-	-	-	+	8
create	1	2	3	+	4
prolong	6	6	6	+	4
delete	7	7	7	+	4
path_succeed	9	9	9	8	+

ret_ext_addr: Return External Address response message

1: Remember the policy rule, but do not install. Rewriting either the source or destination address depending on whether the packet comes from the external_address or not. Always forward.

2: Remember policy rule, but do not install. Always forward.

3: 1+2. The order depends on whether it comes from the outside address (NAT, then FW) or the inside one (FW, then NAT)

4: If it fits one of its requests, send a path_succeeded packet back. Otherwise, drop the packet.

5: Make a reservation. If middlebox is an edge NAT is set, send back the reserved external address and do not forward the message further. Otherwise, forward and do not send anything back.

6: Prolong the session. Always forward.

7: Terminate the session. Always forward.

8: hand it over to upper layers, and stop processing.

9: If it fits a prior request, enable policy rule that has been remembered only before.

-: ignore and forward.

+: ignore and drop.

Note that policy rule ordering at middlebox is important, when it comes to combined NAT and firewall middleboxes, because the filter rules have to be set up according to the packet they will see. Source NAT is done at the end so it does not disturb routing

decisions, meaning that filter sees the original packets. Destination NAT, on the other hand, is done at the beginning, so it can be routed properly, and so the filter sees the modified packets.

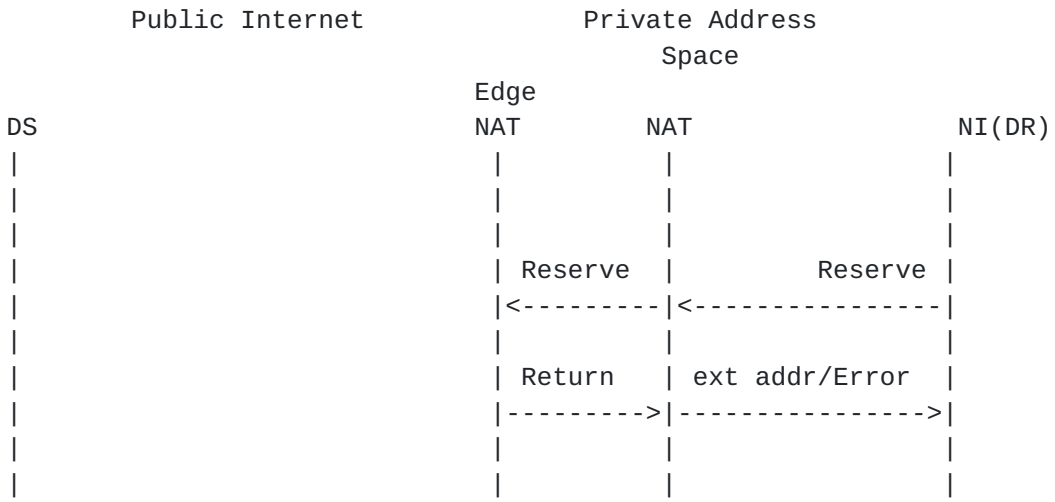
Note also that for each action, the host might demand a certain degree of authorization, and thus refuse to take the action, sending an error message back instead.

The details of protocol operations for each request type is defined in the next sections. Each section describes the exact handling for each type of middlebox.

5.6.1.1 Reserving Addresses

As explained in section Section 4, data receivers located behind a NAT must first reserve an external address and port number (if applicable) before any NSIS message can be send towards them.

With the reserved external address message exchange NSIS peers can obtain this required external address and port number at a NAT. Therefore, NI sets the policy rule object and sends the signaling message to an address chosen on its own (see Section 4.2.1. The reserve message is sent in this way:



Handling of reserve external address messages depends on the middlebox type and NSIS peer:

o NAT Box:

When a NAT box gets a Reserve external address message, it checks

whether it arrived on the public address, or the private one. If it arrived in the public one, an error message of the type: "Requested an external address from the outside" is sent back.

If it arrived on the private side, an entry is made in the internal reservation list with the packet information. If the box is an edge NAT (either by configuring it to true, or just for that connection if it is set to auto), it drops the message, and replies with a return external address message containing the allocated address port pair. If it is not an edge NAT, it forwards the packet on.

- Firewall Box:

  Reserve messages are silently ignored in Firewall boxes. They are simply forwarded on.

- NAT+FW Box:

  When a box that integrates both a NAT and a Firewall gets a reserve message, it will hand it to its NAT part. Its firewall part will simple ignore it.

- Data Sender:

  The message should never get here. It should be ignored and dropped.

- Data Receiver:

  The message should never get here. It should be ignored and dropped.

Response messages are handled differently depending on NSIS peer type:

- NAT Box, Firewall Box and NAT+Firewall Box:

  When one of these boxes gets a Return external address message, it must simply ignore it and let it traverse.

- Data Sender:

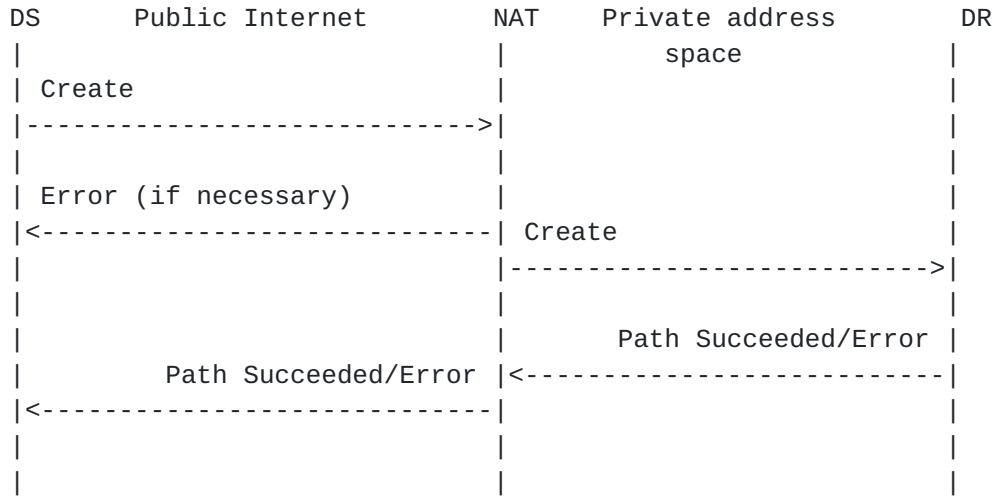
  The message should never get here. It should be ignored and dropped.

- Data Receiver:

A return external address message in the Data receiver, has reached its destination. It must be dropped, and it's information handed to superior layers.

5.6.1.2 Creating Sessions

Creating sessions enables communication between DS and DR. Both are enabled to exchange data packets even with middleboxes on path. DS generates a create session message with a chosen session ID, the policy rule object set to the requested flow, and a requested lifetime. DS sends the create session message towards DR. The message flow is sketched in the next figure.



Create session messages are processed differently at each NSIS peer:

o NAT Box:

When a NAT box gets a create message, it first checks if it arrived on the public address or not.

If it came from the public side, it means an external box will try to send data. It then looks for a reservation in its reservation list, that matches the dst_addr and dst_port of policy rule included in the create message. If it does not find it, it returns an error message of the type "No reservation found". If it finds it, it fills in the reservation with the data from the packet, and remembers the given rule. It then changes the dst_addr and dst_port fields of the create packet and forwards it to the tgt_addr of the reservation.

If it came from the private side, it installs the NAT rule with the information in the packet. It then changes the src_addr and src_port of the create message to its own external address and port.

- Firewall Box:

  When a firewall box gets a create message, it simply remembers the rule specified in the message and forwards the packet.

- NAT+FW Box:

  When a box that integrates both a NAT and a Firewall gets a create message, it first checks whether it arrived on the public address or not.

  If it arrived on the public side, the NAT part of the box takes care of the packet first, as said in the NAT Box case. Afterwards, the modified packet is handed to the firewall part, where it is handled as in the Firewall Box case.

  If it arrived on the private side, the message is handed to the firewall part first, and then to the NAT one.

- Data Sender:

  The message should never get here. It should be ignored and dropped.

- Data Receiver:

  If the data receiver gets a create message, it means all the boxes on the way accepted it, and so the signaling succeeded. All it does is drop the packet, and send back a Path Succeeded message to the IP packet source address.

As described above, DRs return a path succeeded when the create message arrived at DR. The path succeeded message is returned along all NSIS forwarders. Each NSIS forwarder enables the prior remembered policy rules and forwards the message to next NSIS hop.

Forwarding of the path succeeded messages is terminated at the DS.

5.6.1.3 Prolonging Session

NATFW NSLP sessions are maintained on a soft-state base. After a certain timeout they are removed automatically by the middlebox, if they are not refreshed by a prolong session message. DS is sending prolong message towards DR and each NSIS forwarder maintaining state for the given session ID extends the lifetime of the session.

Extending lifetime of a session is calculated as current local time plus lifetime.

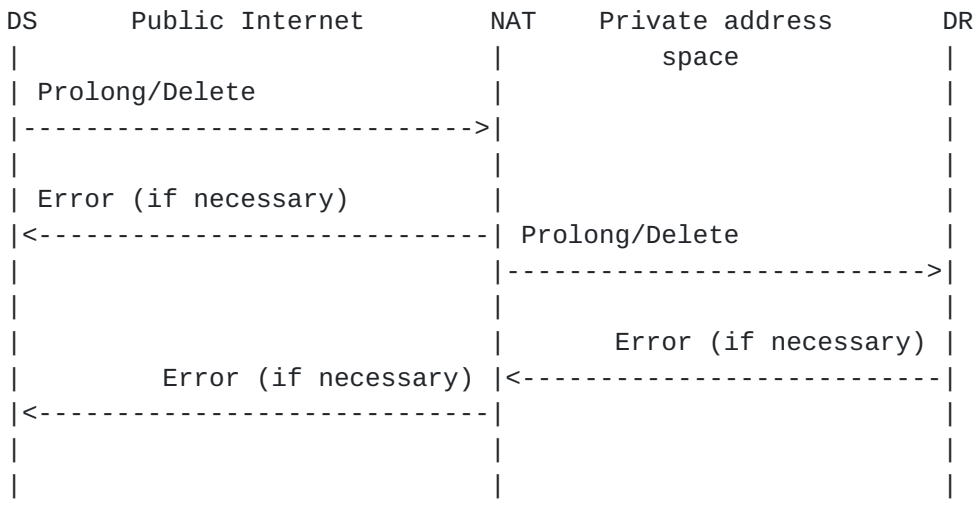


Figure 19: Prolongation message flow

o NAT Box, Firewall Box and NAT+Firewall Box:

When one of these boxes gets a Prolong session message, the expiration time of the session should be changed to the time of reception plus the configured session lifetime.

o Data Sender:

As in the create session message, this packet is sent from the DS to the DR, and should never arrive at the DS. Again, it should be ignored and dropped.

o Data Receiver:

The same behavior as in the case of a Delete session message on the DR should be applied.

5.6.1.4 Deleting Sessions

Deleting sessions is done via the delete session message. DS can request the deletion of a session at any time by sending this message. Processing of these messages at:

o NAT Box, Firewall Box and NAT+Firewall Box:

When one of these boxes gets a Delete session message, it erases the session referred in the message.

- Data Sender:

  This packet should never get to the DS, so it is to be ignored and dropped.

- Data Receiver:

  As in the create session message, this is the final destination of the message. DR erases its session. Message forwarding stops here.

6. Solution examples

6.1 Firewall traversal

DS wants to send data traffic to DR through tight firewalls, as seen in Figure 20. To do that, it will have to signal using NSIS, on the data path.

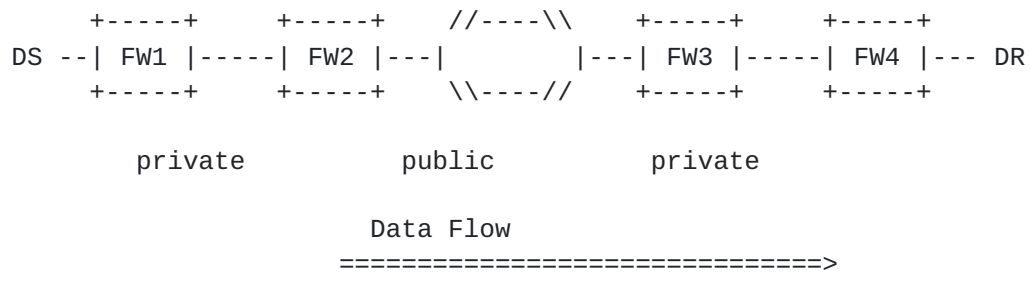


Figure 20: Firewall Traversal Scenario

Therefore, DS initiates signaling to DR by sending a create object to the IP address of DR. Note that DS already knows its source address and port (say, 1111), and the destination address of DR. The destination port (let's say 9999) has been send to DS by DR via application layer messages, possibly, but not necessarily involving a third party. The message looks like:

- dst_addr = DR
- dst_port = 9999
- src_addr = DS
- src_port = 1111

This message is received by FW1, which installs the state that reads: "Any packet coming from DS:1111 headed for DR:9999 will be allowed traversal"

FW2, FW3 and FW4 do exactly the same, and forward the packet to each other, until it finally reaches DR. At this point, the data path is open, and DR sends back a Path succeeded message to DS, which can now start sending traffic.

6.2 NAT with private network on sender side

In the example in Figure 21, DS is in a private network and wants to send data to DR, out in the public internet. To do so, DS will have to initiate NSIS signaling towards DR

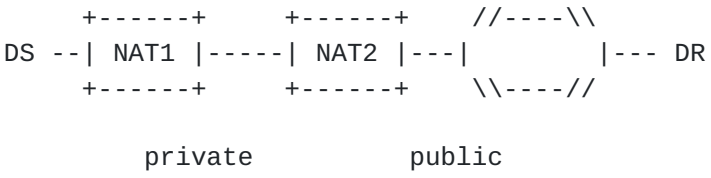


Figure 21: NAT with private network on sender scenario

Apparently, the normal NAT functionality will take care of sending the data from DS out into the public internet, and route back the replies from DR. This is indeed true, but doesn't give NSIS control on what the source address or port is, as it is usually assigned dynamically by the NAT. Moreover, the NSLP would have no information on this hops, and could not install proper pinholes, as it would set DS as the source address, and not that of the last NAT.

DS builds a create packet with the information he has, which is the same as that in Section 6.1. It looks like this:

- dst_addr = DR
- dst_port = 9999
- src_addr = DS
- src_port = 1111

NAT1 is the first to get the packet; It is not coming from its configured "nat external address", and so, it knows it will have to rewrite the information on the source, and not that of the destination. NAT1 then picks a free port (incidentally 1011) and installs a nat rule that reads: "Whatever packet comes from DS:111, heading for DR:9999 will be rewritten so that the source address looks like NAT1:1011".

It then rewrites the packet it received as follows:

- dst_addr = DR

- dst_port = 9999
- src_addr = NAT1
- src_port = 1011

And forwards the packet.

NAT2 gets it now, and does exactly the same. Port 2022 is chosen, and the rule: "Whatever packet comes from NAT1:1011, heading for DR:9999 will be rewritten so that the source address looks like NAT2:2022" is installed. The packet gets modified as follows:

- dst_addr = DR
- dst_port = 9999
- src_addr = NAT2
- src_port = 2022

And is forwarded. It eventually reaches DR, who sends back a path succeeded message. Data flow from DS:1111 to DR:9999 is now possible.

6.3 NAT with private network on receiver side

In this example, DS wants to send data to DR over the network in Figure 22:

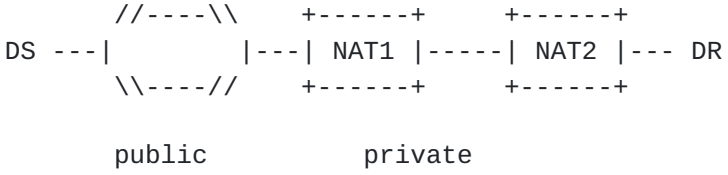


Figure 22: NAT with private network on receiver Scenario

The problem, of course, is that DR is not publicly reachable. Because of that, DR will have to signal on the data path, in the opposite direction (DR->DS) to get itself a public address it can use. This method is described in Section 4

To get an external address, DR sends a packet to DS. It could actually send it to anything in the public internet, as it would force it to traverse what NATs are on its way. In the case of

multihomed environments, though, more than one NAT to the outside is possible, so the better we "aim" the more the chances we go out the right NATs and get more optimal routes.

The said packet is an NSIS reserve_addr object which looks like this:

- tgt_addr = DR
- tgt_port = 9999
- src_addr = 0.0.0.0
- src_port = 0

Notice that this is a really loose pinhole, since any src_addr and port is allowed.

NAT2 gets the packet and looks for a free port (say, 2022, for clarity's sake). It then adds an entry to its reservation list. The entry looks like this:

- src_addr = 0.0.0.0
- src_port = 0
- dst_addr = NAT2
- dst_port = 2022
- tgt_addr = DR
- tgt_port = 9999

This means simply that packets coming from any source, destined to the public address we just reserved, should be targeted to the internal box DR, on port 9999

It then rewrites the packet so that it looks like:

- tgt_addr = NAT2
- tgt_port = 2022
- src_addr = 0.0.0.0
- src_port = 0

Because it is not an edge NAT, it forwards the modified packet and

does not sent a return_external_addr object to DR. Note that no NAT binding is installed so far in NAT2, although the state is now reserved.

NAT1 now gets the packet, picks free port 1011 and adds the following entry to its reservation list:

- src_addr = 0.0.0.0
- src_port = 0
- dst_addr = NAT1
- dst_port = 1011
- tgt_addr = NAT2
- tgt_port = 2022

As it turns out, NAT1 IS an edge_nat, so it doesn't forward the packet. Instead, it replies to DR sending back a return external address packet on the same connection, so it finds its way back through the NATs:

- ext_addr = NAT2
- ext_port = 2022

By using some application layer protocol, and possibly, although not necessarily, using a third party box, DR sends it's freshly allocated external address and port to DS.

DS now knows who to signal, so it sends a create message:

- dst_addr = NAT1
- dst_port = 1011
- src_addr = DS
- src_port = 1111

When it reaches NAT1, it does so through NAT1 external address. It realizes it is being asked to forward the traffic from some outside box towards the inside. It then looks up its reservation list, looking for a session that has the external address and port NAT1:1011 assigned. It finds this:

- src_addr = 0.0.0.0
- src_port = 0
- dst_addr = NAT1
- dst_port = 1011
- tgt_addr = NAT2
- tgt_port = 2022

Using the information in the create object, it then fills in this structure to:

- src_addr = DS
- src_port = 1111
- dst_addr = NAT1
- dst_port = 1011
- tgt_addr = NAT2
- tgt_port = 2022

This IS a tight pinhole. NAT1 installs the rules now, which say: "Whatever packet comes from DS:1111 heading for NAT1:1011, should have its destination address changed to NAT2:2022, and be forwarded". The packet is also rewritten into this:

- src_addr = DS
- src_port = 1111
- dst_addr = NAT2
- dst_port = 2022

And is forwarded to NAT2. Upon arrival, a similar process issues. NAT2 finds its reservation entry:

- src_addr = 0.0.0.0
- src_port = 0
- dst_addr = NAT2

- dst_port = 2022
- tgt_addr = DR
- tgt_port = 9999

Fills it in accordingly:

- src_addr = DS
- src_port = 1111
- dst_addr = NAT2
- dst_port = 2022
- tgt_addr = DR
- tgt_port = 9999

Rewrites the packet:

- src_addr = DS
- src_port = 1111
- dst_addr = DR
- dst_port = 2222

And forwards it to DR. Once there, DR acknowledges it by sending back a path succeeded message in reply, back to DS.

The path is now open and data transmission from DS:1111->DR:9999 can commence.

6.4 Both end hosts are in same private network behind NATs

In this example (see Figure 23), DS, in a private address space, wants to send data to DR, in another private address space. The point marked "%" is yet another private address space. Notice that since NAT1 and NAT3 have addresses in the same address space, NAT3 might want to consider itself an edge NAT. We will consider both situations.

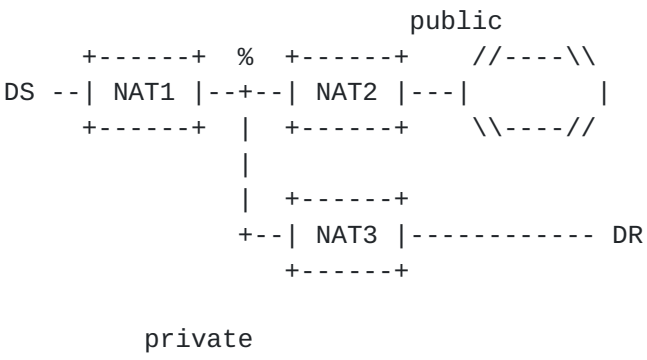


Figure 23: NAT to public, receiver in same private network Scenario

We will first assume that NAT3 has the edge_nat option set to false. In this case, the connection is a combination of Section 6.3 and Section 6.2.

Firstly DR will signal against on the data path, against the data flow, with a reserve external address object. NAT3 will reserve the address and forward the packet on to NAT2, who IS an edge NAT in all cases. NAT2 will reply with the external address, and the connection goes on just as in Section 6.2, except for the fact the topology becomes:

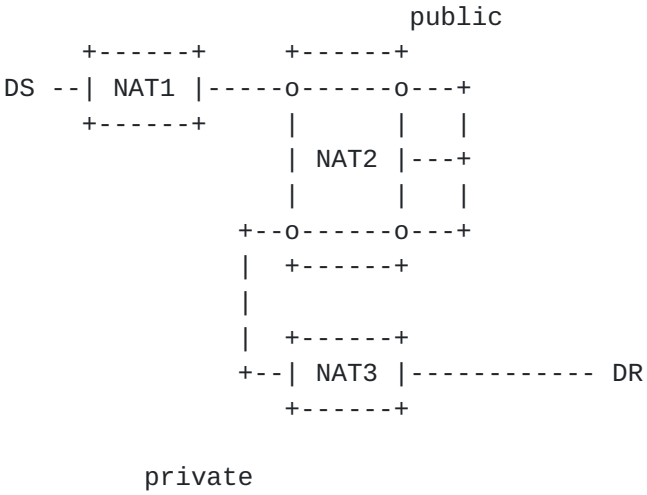


Figure 24: New topology due to the non optimal edge nat parameter decision

This is not optimal, but the connection does succeed, and data flow can commence.

Let us now solve the case in which NAT3 has edge_nat set to auto. Back in Figure 23, NAT3 will decide it IS an edge_nat if the destination we pick up for the reserve address packet is in the address space marked as "%", and will NOT consider itself an edge_nat if we point it anywhere else. This is an optimization issue such as the one pointed out in Section 6.3.

Well so, if it doesn't consider itself an edge NAT, we already saw what the topological equivalent is, and how it proceeds. If it IS an edge NAT, the topological equivalent would be:

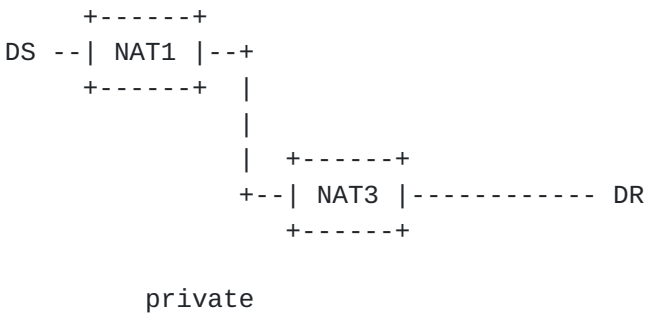


Figure 25: A good edge nat decision brings an optimal route

And we would proceed in the same way, only on a more optimal route.

6.5 IPv4/v6 NAT with two private networks

TBD

6.6 Full example for NAT/FW with two private networks

The NAT's have the nat_capabilities variable set to true. NAT+FW3 and NAT+FW5 have the edge_nat variable set to true. The rest of boxes have it set to false.

Let's now suppose that DR wants to get a data stream from DS in Figure 26. For that, we need some way for B to get messages from A, be it through some third party application server or some publicly reachable proxy, perhaps made public through a NAT binding.

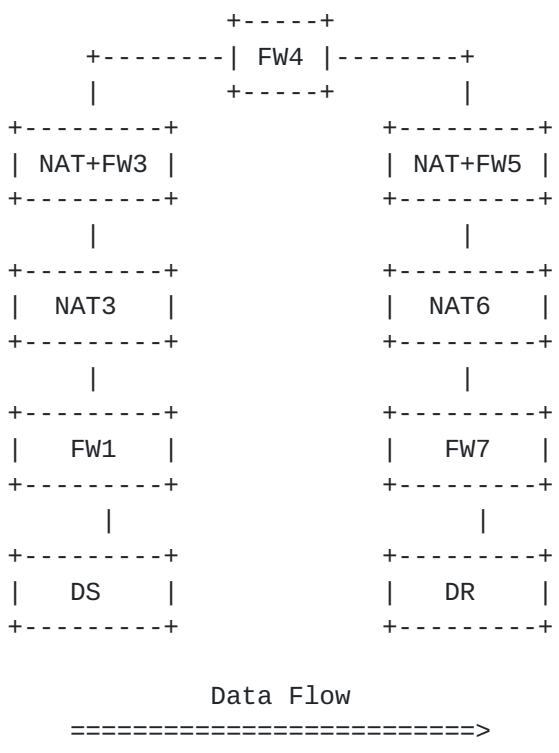


Figure 26: Example network topology

DR wants a data stream from DS, which means that the direction of the data is DS->DR. A will have to make itself publicly reachable by signaling its NATs and firewalls as necessary. This is a step by step guide to the whole process.

In steps 1 to 4, DR makes itself publicly reachable. From 5 and on, DS is signaling on the data path towards DR.

1. DR wants to get data from DS, so it sends a reserve_addr object to a target in the public internet. The closest this target is, the more the chances that the resulting route is optimal, but any will work. The reserve_addr obj looks like this:

o tgt_addr = DR

o tgt_port = 888

o src_addr = 0.0.0.0

o src_port = 0

Notice that this is a really loose pinhole, since any src_addr and port is allowed.

2. FW7 gets the packet, ignores its contents and forwards it. Firewalls always ignore reserve_addr objects.

3. NAT6 gets the packet, and looks for a free port (say, 666, for clarity's sake). It then adds an entry to its reservation list. The entry looks like this:

- src_addr = 0.0.0.0
- src_port = 0
- dst_addr = NAT6
- dst_port = 666
- tgt_addr = DR
- tgt_port = 888

It then rewrites the packet so that it looks like:

- tgt_addr = NAT6
- tgt_port = 666
- src_addr = 0.0.0.0
- src_port = 0

Because it is not an edge NAT (edge_nat=false), it does not sent a return_external_addr object to DR, but rather forwards the modified packet. Note that no NAT binding is installed so far in NAT6, although the state is now reserved.

4. NAT+FW5 receives the packet. The firewall part gets the object, but, being as it is an address reservation only object, it ignores it. The NAT part gets it next. Because it is a NAT, it binds a free port, which is thus reserved. An entry to the reservation list is added:

- src_addr = 0.0.0.0
- src_port = 0
- dst_addr = NAT+FW5
- dst_port = 555

- tgt_addr = NAT6

- tgt_port = 666

Because it is an edge_nat, it sends a return_external_addr packet with address NAT+FW5 and port 555 back to DR. It does so by simply sending it back to the source IP address in the IP header of the packet. In this case, it is NAT6. The standard capabilities of NAT6 will send it back to DR, since we are always working on the same connection. Because it is an edge_nat and this is a reserve_external_addr packet, it does not forward the packet.

At this stage, the end host DR has learned what its (reserved) external address is, even if it can not be used. It is now publicly reachable, and path-coupled NSIS signaling in direction DS->DR can start.

5. Firstly, DR tells DS about it's freshly reserved outside address through some higher layer protocol, using the third-party box.

6. DS now initiates signaling to DR by sending a create object to the brand new public address of DR. It looks like:

- dst_addr = NAT+FW5

- dst_port = 555

- src_addr = DS

- src_port = 111

7. The firewall FW1 gets it, and installs the requested pinhole. (Note this IS a tight pinhole with well defined source and destination). It then forwards the packet.

8. NAT2 gets the packet. Because it is NOT coming from it's external address, it realizes it is being asked to forward DS's future data packets, and so, it will have to rewrite it's source address. To do so, NAT2 picks a random free port (which turns out to be 222), and installs a NAT rule that says: "Whatever packet comes from DS:111, heading for NAT+FW5:555 will be rewritten so that the source address looks like NAT2:222". That is usually known as Source NAT. The NSIS create request is then rewritten to look like:

- dst_addr = NAT+FW5

- dst_port = 555

- src_addr = NAT2
- src_port = 222

Because it is not an edge NAT, it simply forwards the modified packet.

9. NAT+FW3 gets the packet next. Because it is NOT coming from the extarnal_addr of the NAT+FW, The firewall part gets it first, and installs the filter rule that says: "Allow traversal of packets going from NAT2:222 towards NAT+FW5:555". It then hands it to the NAT part.

The NAT part gets it then. It is not coming from its external address, and so, it does as NAT2, binding a port (333) and installing a rule that says: "Whatever packet comes from NAT2:222, heading for NAT+FW5:555, will be rewritten so that the source address looks like NAT+FW3:333". It will then rewrite the create object to:

- dst_addr = NAT+FW5
- dst_port = 555
- src_addr = NAT+FW3
- src_port = 333

Note that the box won't send a packet back to DS informing it of its external address, because DS will never need that.

10. FW4 gets the create object, and installs the rule "Allow traversal of packets going from NAT+FW3:333 towards NAT+FW5:555" It then forwards the object.

11. NAT+FW5 gets the create object. It arrived at its external address, so it realizes it doesn't have to change the source address of the future data packets of DS, but rather its destination. It also means that the NAT part will have to handle it first. It then tries to find out where it has to re-destined it to, by looking up its reservation tables. It finds the previous reservation, by matching it with their dst_addr and dst_port of the create object:

- src_addr = 0.0.0.0
- src_port = 0
- dst_addr = NAT+FW5
- dst_port = 555

- tgt_addr = NAT6
- tgt_port = 666

And proceeds to fill it in with the information of the create object (src_addr and src_port):

- src_addr = NAT+FW3
- src_port = 333
- dst_addr = NAT+FW5
- dst_port = 555
- tgt_addr = NAT6
- tgt_port = 666

It then installs a NAT rule with that information. It reads: "Whatever packet comes from NAT+FW3:333, heading for NAT+FW5:555 will be rewritten, so that its destination address looks like NAT6:666". The reservation is erased and the rule starts working. The NAT binding becomes thus usable.

The object is modified, so that it now looks like:

- dst_addr = NAT+FW3
- dst_port = 333
- src_addr = NAT6
- src_port = 666

The FW part now gets the object, and installs the rule: "Allow traversal of whatever packet that comes from NAT+FW3:333 heading for NAT6:666". The packet is then forwarded.

12. NAT6 gets the packet. As it comes from the external address, it does as NAT+FW5, looking up the reservation list and filling it in with:

- src_addr = NAT+FW3
- src_port = 333
- dst_addr = NAT6

- dst_port = 666
- tgt_addr = DR
- tgt_port = 888

It then installs the rule: "Whatever packet comes from NAT+FW3:333, heading for NAT6:666 will be rewritten, so that its destination address looks like DR:888". The rule reservation is erased, and the NAT binding becomes active. The object is rewritten as:

- src_addr = NAT+FW3
- src_port = 333
- dst_addr = DR
- dst_port = 888

The object is thus forwarded.

13. FW7 gets the packet now, and installs the rule: "Allow traversal of whatever packet that comes from NAT+FW3:333 heading for DR:888". It forwards the packet.

14: DR gets (finally) the packet. It realizes it is a create object headed for him, to the port which he expected, and so it sees everything went well. A reply to the packet is send, and the NAT's on the way, knowing the already established connection, will route it to DS. The packet is a path_succesful message, which simply means "Everything is fine, send data whenever you want".

7. NSIS NAT and Firewall transitions issues

NSIS NAT and Firewall transition issues are premature and will be addressed in a separate draft (see [16]). An update of this section will be based on consensus.

8. Security Considerations

Security is of major concern particularly in case of firewall traversal. Generic threats for NSIS signaling have been discussed in [5] and are applicable here as well. It is necessary to provide proper signaling message protection and proper authorization. Note that the NAT is likely to be co-located with a firewall and might therefore require packet filters to be changed in order to allow the signaling message to process and to traverse. This section aims to raise some items for further discussion and illustrates the problems the authors faced when creating a security solution for the NAT/ Firewall NSLP.

Installing packet filters provides some security, but has some weaknesses, which heavily depend on the type of packet filter installed. A packet filter cannot prevent an adversary to inject traffic (due to the IP spoofing capabilities). This type of attack might not be particular helpful if the packet filter is a standard 5 tuple which is very restrictive. If packet filter installation, however, allows specifying a rule, which restricts only the source IP address, then IP spoofing allows transmitting traffic to an arbitrary address. NSIS aims to provide path-coupled signaling and therefore an adversary is somewhat restricted in the location from which attacks can be performed. Some trust is therefore assumed from nodes and networks along the path.

Without doubts there is a dependency on the security provided by the NTLP. Section Section 3 and Section 2.2 motivates some trust relationship and authorization scenarios. These scenarios deserve a discussion since some of them (particularly one with a missing network-to-network trust relationship) is different to what is know from QoS signaling. To address some of these trust relationships and authorization issues requires security mechanisms between non-neighboring nodes at the NSLP layer. For the group of authors it seems that peer-to-peer and end-to-middle security needs to be provided. An NSLP security mechanism between neighboring NSLP peers might be necessary if security mechanisms at the NTLP do not provide adequate protection mechanisms. This issue is, however, still in discussion.

As a design goal it seems to be favorable to reuse existing mechanisms to the best extend possible. In most cases it is necessary to carry the objects for end-to-middle as NSLP payloads since the presence of NATs might prevent direct communication. Three security mechanisms have to be considered in more detail in a future version of this document: CMS [17] and Identity Representation for RSVP [14]. The authors believe that CMS more suitable (since it provides much more functionality). The details deserve further

discussion and implementation experience.

With regard to signal between two end hosts even though the receiver is behind a NAT this proposal suggests creating state by the data receiver first. This allows NSIS signaling messages to traverse a NAT at the receiver side (due to the established state at this NAT box) and simplifies security handling. To achieve this behavior it is required to install NSIS NTLP and NSLP state. Furthermore, it is envisioned to associate the two signaling parts (one part from the data sender to the NAT and the other part from the NAT to the data receiver) with the help of the Session Identifier. As such, the discussion in [14] is relevant for this document.

Another interesting property of this protocol proposal is to prevent Denial of Service attacks against NAT boxes whereby an adversary allocates NAT bindings with the help of data packets. Since these data packets do not provide any type of authentication and are not authorized any adversary is able to mount such an attack. This attack has been mentioned at several places in the literature already and is particularly harmful if no NAPT functionality is used (i.e. if a new NAT binding consumes one IP address of a pool of IP addresses). Using the protocol described in this document additional security can be achieved and more fairness can be provided.

9. Open Issues

At least the following issues require further discussion:

- Message format: The exact message format is still to be determined, both in regards of bit level details and on parameters, such as the need for an object header length, since, until now, that is a constant.

- Message type numbering

- Error codes: error codes have to be defined still. Among others, we will need: missing authorization, out of resources, unable to understand the packet, or maximum resources for that individual already allocated.

- middlebox default policies: allow for the configuration of the default policies of the box. For a NAT+Firewall box, for instance, the firewall default policy might be "accept", and so, no packet filters would have to be installed on that regard (we would still need the NAT bindings, though).

- IPv6 flow label usage

- Stacking

- Edit Section 6 "Solution Examples"

- Edit Security Consideration section

- Edit Appendix A.

10. Contributors

A number of individuals have contributed to this draft. Since it was not possible to list them all in the authors section, it was decided to split it and move Marcus Brunner and Henning Schulzrinne into the contributors section. Separating into two groups was done without treating any one of them better (or worse) than others.

Normative References

[1] Hancock et al, R., "Next Steps in Signaling: Framework", DRAFT draft-ietf-nsis-fw-05.txt, October 2003.

[2] Brunner et al., M., "Requirements for Signaling Protocols", DRAFT draft-ietf-nsis-req-09.txt, October 2003.

[3] Schulzrinne, H. and R. Hancock, "GIMPS: General Internet Messaging Protocol for Signaling", DRAFT draft-ietf-nsis-ntlp-00.txt, October 2003.

[4] IANA, "Special-Use IPv4 Addresses", RFC 3330, September 2002.

[5] Tschofenig, H. and D. Kroeselberg, "Security Threats for NSIS", DRAFT draft-ietf-nsis-threats-01.txt, January 2003.

[6] Srisuresh, P., Kuthan, J., Rosenberg, J., Molitor, A. and A. Rayhan, "Middlebox communication architecture and framework", RFC 3303, August 2002.

Informative References

[7] Srisuresh, P. and M. Holdrege, "IP Network Address Translator (NAT) Terminology and Considerations, RFC 2663", August 1999.

[8] Srisuresh, P. and M. Holdrege, "Network Address Translator (NAT)Terminology and Considerations, RFC 2663".

[9] Srisuresh, P. and E. Egevang, "Traditional IP Network Address Translator (Traditional NAT), RFC 3022", January 2001.

[10] Tsirtsis, G. and P. Srisuresh, "Network Address Translation - Protocol Translation (NAT-PT), RFC 2766", February 2000.

[11] Carpenter, B. and S. Brim, "Middleboxes: Taxonomy and Issues", RFC 3234, February 2002.

[12] Srisuresh, P., Tsirtsis, G., Akkiraju, P. and A. Heffernan, "DNS extensions to Network Address Translators (DNS_ALG)", RFC 2694, September 1999.

[13] Braden, B., Zhang, L., Berson, S., Herzog, S. and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", September 1997.

[14] Yadav, S., Yavatkar, R., Pabbati, R., Ford, P., Moore, T., Herzog, S. and R. Hess, "Identity Representation for RSVP", RFC 3182, October 2001.

[15] Tschofenig, H., Schulzrinne, H., Hancock, R., McDonald, A. and X. Fu, "Security Implications of the Session Identifier", June 2003.

[16] Aoun and others...., C., "NAT/Firewall NSLP migration, routing, NTLP requirements and off-path Considerations", October 2003.

[17] Housley, R., "Cryptographic Message Syntax (CMS)", RFC 3369, August 2002.

[18] Manner, J., Suihko, T., Kojo, M., Liljeberg, M. and K. Raatikainen, "Localized RSVP", DRAFT draft-manner-lrsvp-00.txt, November 2002.

[19] Tschofenig, H., Buechli, M., Van den Bosch, S. and H. Schulzrinne, "NSIS Authentication, Authorization and Accounting Issues", March 2003.

[20] Amini, L. and H. Schulzrinne, "Observations from router-level

internet traces", DIMACS Workshop on Internet and WWW Measurement, Mapping and Modelin Jersey) , Februar 2002.

[21] Adrangi, F. and H. Levkowetz, "Problem Statement: Mobile IPv4 Traversal of VPN Gateways", draft-ietf-mobileip-vpn-problem-statement-req-02.txt (work in progress), April 2003.

[22] Ohba, Y., Das, S., Patil, P., Soliman, H. and A. Yegin, "Problem Space and Usage Scenarios for PANA", draft-ietf-pana-usage-scenarios-06 (work in progress), April 2003.

[23] Shore, M., "The TIST (Topology-Insensitive Service Traversal) Protocol", DRAFT draft-shore-tist-prot-00.txt, May 2002.

[24] Tschofenig, H., Schulzrinne, H. and C. Aoun, "A Firewall/NAT Traversal Client for CASP", DRAFT draft-tschofenig-nsis-casp-midcom-01.txt, March 2003.

[25] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M. and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

[26] Brunner, M., Stiemerling, M., Martin, M., Tschofenig, H. and H. Schulzrinne, "NSIS NAT/FW NSLP: Problem Statement and Framework", DRAFT draft-brunner-nsis-midcom-ps-00.txt, June 2003.

Authors' Addresses

Martin Stiemerling
Network Laboratories, NEC Europe Ltd.
Kurfuersten-Anlage 36
Heidelberg 69115
Germany

Phone: +49 (0) 6221 905 11 13
EMail: stiemerling@netlab.nec.de
URI:

Hannes Tschoefenig
Siemens AG
Otto-Hahn-Ring 6
Munich 81739
Germany

Phone:
EMail: Hannes.Tschofenig@siemens.com
URI:

Miquel Martin
Network Laboratories, NEC Europe Ltd.
Kurfuersten-Anlage 36
Heidelberg 69115
Germany

Phone: +49 (0) 6221 905 11 16
EMail: miquel.martin@netlab.nec.de
URI:

Cedric Aoun
Nortel Networks

France

EMail: cedric.aoun@nortelnetworks.com

Appendix A. Inter-working of SIP with NSIS NATFW NSLP

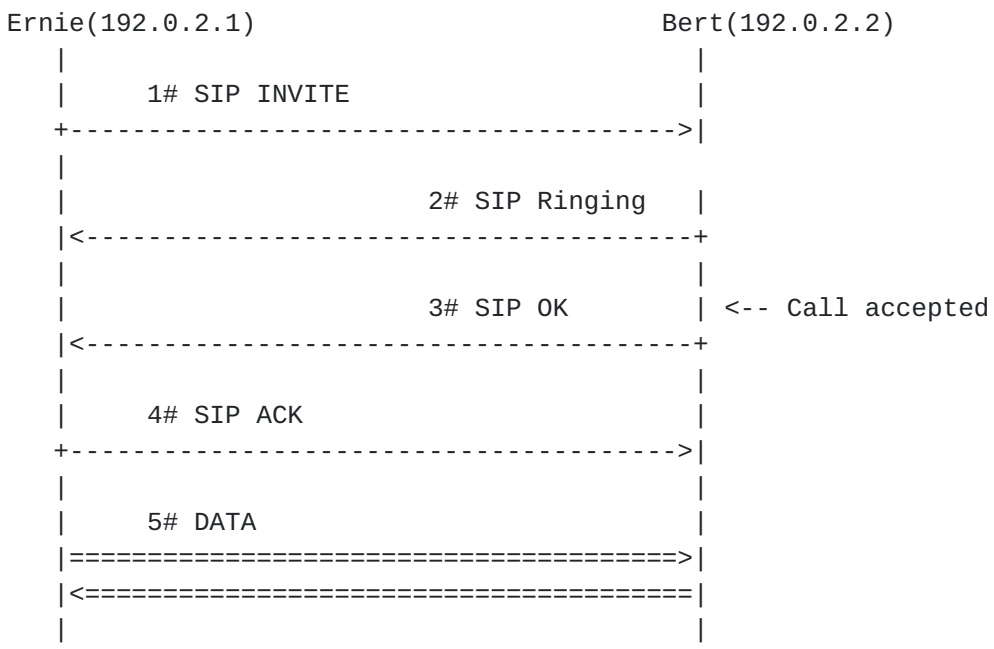
This document aims at pinpointing the problems of using SIP in nowadays networks, focusing on the problems derived of NAT's, Firewalls and multi-path communications. It is intended to fit in a scenario description that shows the necessity of NSIS, as well as depicting it's requirements. However, note that there are a number of other solutions available. For example the IETF Midcom working group is working on [6].

A.1 The Session Initiation Protocol

[25] describes the Session Initiation Protocol, an application-layer control protocol that can establish, modify, and terminate multimedia sessions. This often involves several flows for video and voice, which are transported over new connections. These use of dynamically allocated ports which results in protocol complexity which can not be handled by nowadays NAT's and Firewalls.

Session initiation when one or both of the users is behind a NAT is also not possible, given the impossibility to address a private IP over the internet. Moreover, network deployments often allow for different paths per connection and direction, making the setup of the middleboxes even more complicated.

The following figure depicts a typical SIP connection:



1# SIP Invite (192.0.2.1:? -> 192.0.2.2:SIP): I Listen on 192.0.2.1:1000 Ernie invites Bert to the conference, and informs it's awaiting media data on port 1000.

2# SIP Ringing (192.0.2.2:SIP -> 192.0.2.1:?): Ringing Bert's phone The ringing simply implies that there's something sip aware on Bert's side, and that it's ringing his phone

3# SIP OK (192.0.2.2:SIP -> 192.0.2.1:?): Call accepted, I listen on 192.0.2.2:2000 This OK means that the Bert took the phone off hook, and thus accepted the call. It also informs Ernie that Bert is awaiting his media data at port 2000

4# SIP ACK (192.0.2.1:? -> 192.0.2.2:SIP): All is fine, start transmitting. ACK means the ports are accepted and the call can start in the selected data ports on both sides.

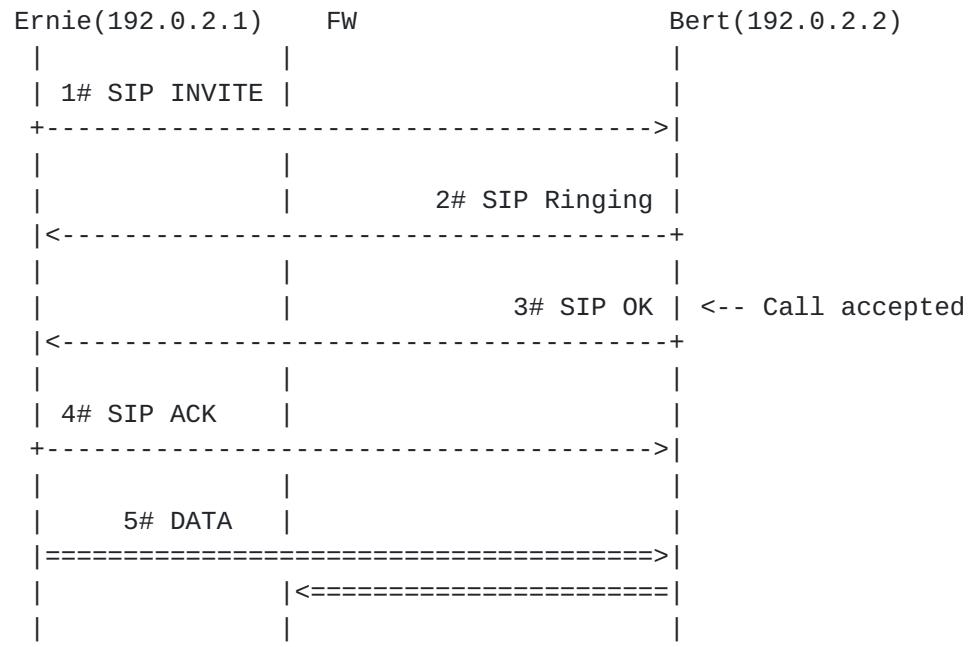
5# DATA (192.0.2.1:? -> 192.0.2.2:2000 and 192.0.2.2:? -> 192.0.2.1:1000): Voice,image, video.. This is the actual data being transmitted.

In the above example, SIP is used successfully to establish a communication, which includes negotiating the data ports for the actual transmission. Unfortunately, this scheme will not work for more complex setups.

Let's now consider one firewall in the data path, be it on Ernie's or Bert's network, or elsewhere in the middle. We assume that the

firewall is allowing traffic directed to the SIP port. As to the rest of the ports, a typical setup involves outgoing connections being allowed, and incoming connections being dropped, except for those already established. That is, we allow packets to go out and their replies to come in, but disable all other traffic.

In this case, the connection is as follows, for the case of a firewall on Ernie's network:



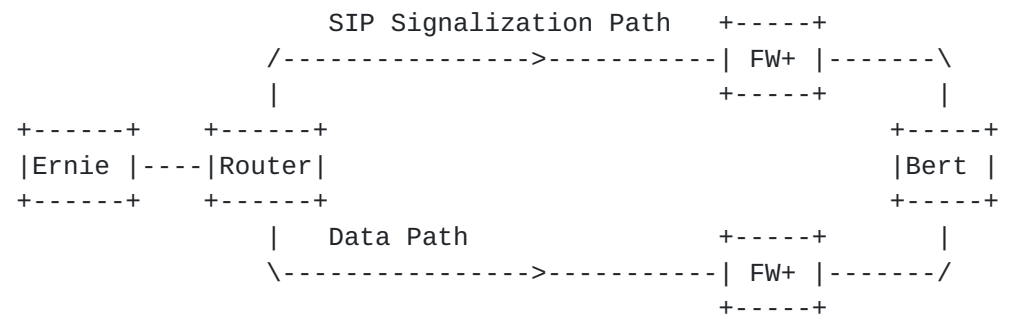
Notice how the SIP messages #1 and #4 traverse the firewall, because they are outbound, and how 2# and 3# traverse it too, because they are replies to the connection established at 1#.

Notice now how 5# can go outwards, but Bert can not go through the firewall to reach Ernie's port 1000. The reason is the connection is a new one, and the firewall won't allow it through.

Bert will now get media from Ernie, but Ernie is never going to get anything from Bert. The call is thus considered unsuccessful. The reason is that the application level port negotiation is never acknowledge by the network-transport layer firewall, which doesn't know what to expect. We would still face the same problem if the connection used a SIP Proxy, for it would only translate names into IP addresses.

Let us now assume that we indeed have an application layer firewall,

be it by design, or because we load some sort of SIP module to it. The previous case would now work, since the firewall can now understand the packets going through it and open the necessary ports. Still, we cannot assume that SIP signalization packets and the actual data follow the same path. The following figure shows a likely setup. FW+ stands for one or more firewalls:



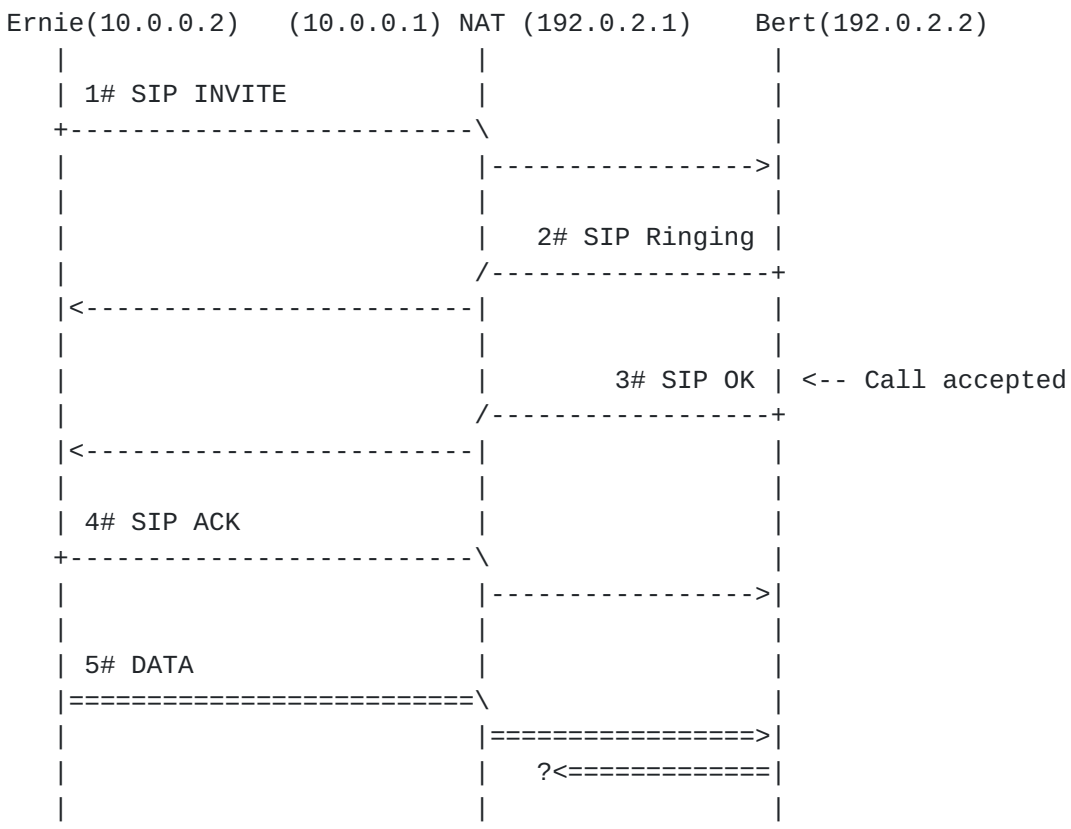
The SIP packets with the information about the listening ports now travels on the SIP Signalization path, and so the firewalls on that path can read them. The Data, though, is traveling through the Data path, and the firewalls in that path never get to see the Invite and Ok packets. They are thus unable to open the ports.

Two issues are arisen here: first, we need on-path signalization unless we already know the path our packets will take; a highly unlikely situation in today's internet. Second, if we patch the firewalls to understand SIP, we will provide any caller with a hole-puncher for the firewall, since SIP is not provisioned with proper authentication mechanism.

It is now clear that tight firewalls prevent SIP from successfully working. There is still another obstacle: NATs.

NATs provide for a link between two different address spaces, typically connecting a private range network to a public range one. As a consequence, connections going from the inside (usually the private range) are translated using the NAT's public interface address, and the replies are routed back. The public side of the network can only see the NATs public interface, and know nothing of the private network inside. This means computers outside the NAT won't be able to address computers inside the NAT.

Let us analyse the SIP example when Ernie is behind a NAT. The following figure depicts a typical session:



The communication is analogous to the one in the previous examples, except for the fact the NAT is rewriting the source address of the packets as they traverse it.

For instance, packet 1# is going from 10.0.0.2:? towards 192.0.2.2:SIP. The NAT box intercepts the message and puts 192.0.2.1:? as the source address and port, with ? being a dynamically picked port, which might be different from the original one 1# used.

On the way back, Bert is replying to the source of the IP packet, that is, 192.0.2.1, and so, when 2# reaches 192.0.2.1, the NAT know it is a reply from 1#, because it established a NAT binding, and this replaces the destination address, 192.0.2.1:? with 10.0.0.2:? and forwards the packet inside the NAT.

As a result, Ernie never knows there is a NAT in his communication path, since he sends and receives packets from 192.0.2.2 normally. This means that the INVITE packet will tell Bert to send data back to 10.0.0.2, a private IP. Once the signalization is finished, and the actual DATA transmission starts, Bert tries to connect to 10.0.0.2, a private IP address, from the internet; The routers don't know how to route this, and the packet is eventually dropped.

One possible solution would be for Ernie to know the NAT exists, and already indicate that it listens on 192.0.2.1, and not 10.0.0.2. That, still would not work, since the NAT binding is not performed at the NAT box.

A.2 Conclusions

The above examples display the inability to use standard SIP through tight firewalls or NATs, and points at the necessity of a secure on-path protocol to negotiate firewall pinholes and NAT bindings.

Appendix B. Ad-Hoc networks

Some forms of ad-hoc networks exist where trust in the network is not justified. Figure Figure 31 mainly illustrates the problems of malicious NSIS entities graphically:

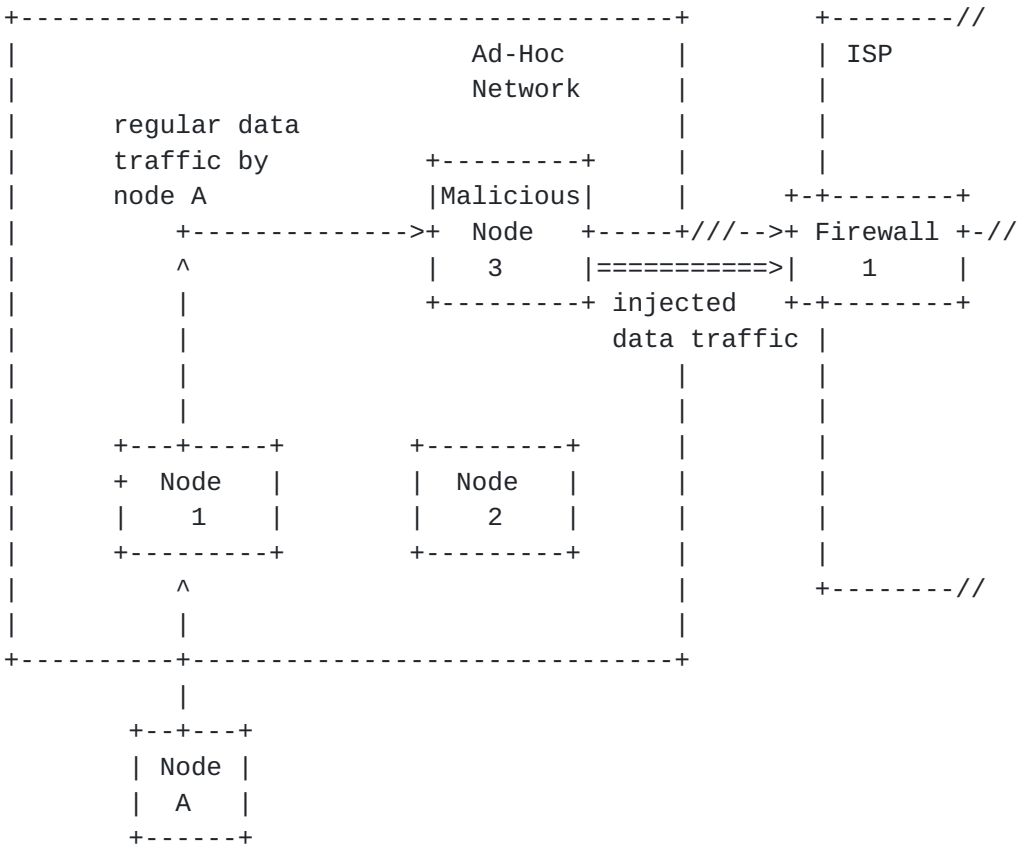


Figure 31: Limits of packet filter security

An ad-hoc network consists of a number of nodes between the end host (Node A) and the ISP to which Node A wants to get access. Although Node A uses an authentication and key exchange protocol to create a policy rule at the firewall 1 it is still possible for an untrusted node (in this case Node 3) to inject data traffic which will pass Firewall 1 since the data traffic is not authenticated. To prevent this type of threat two approaches are possible. First, a restrictive packet filter limits the capabilities of an adversary. Finally, there is always the option of using data traffic protection.

Appendix C. Interworking of Security Mechanisms and NSIS NATFW NSLP

TBD

Appendix D. Solution approaches in case of missing authorization

D.1 Solution Approach: Local authorization from both end points

The first approach makes use of local authorization from both end points. If Host A sends a signaling message toward the destination to Middlebox 1 the message will perform the desired action in Network A. Middlebox 1 establishes some state information and forwards the signaling message towards Host B. Signaling message protection between the two access networks might be difficult. A missing trust relationship does not necessarily mean that no security association establishment is possible. The lacking trust disallows Middlebox 1 (or indirectly Host A where the signaling message was initiated) to create packet filters at Middlebox 2. We assume that the NSIS signaling message is allowed to pass the firewall then it finally reaches Host B. Due to the missing authorization no packet filter specific state is created. The filters will be installed later after receiving an authorization from Host B. When Host B returns a confirmation or acknowledgement then Middlebox 2 treats it as an authorization and finally triggers filter creation. The message is then forwarded to Middlebox 1, where filters are either already installed or require an additional confirmation. Finally the signaling message is forwarded to Host A, which can be assured that subsequent data traffic can be transmitted end-to-end from Host A to Host B. The same procedure has to be applied again to signal information for the other direction (Host B to Host A).

The following behavior has to be assumed in order for this approach to be applicable:

1. Signaling messages must be allowed to pass firewalls along the path.

2. NSIS signaling must operate in the described manner which could be described as: Install where you have authorization - delay and forward where you have no authorization.

This approach suffers from the following drawbacks:

1. Firewalls which block NSIS signaling from external networks or nodes prevent a successful operation.

2. A full roundtrip is required to signal packet filter information. The NSIS signaling message must therefore provide the capability to route signaling message in both direction which might either require state installation at nodes along the path (route pinning) or a stateless version via record-route. Some risk of DoS protection might exist.

D.2 Solution Approach: Access Network-Only Signaling

The next approach is based on signaling packet filter information by both hosts into the local access network only. An NSIS allows specifying such a behavior by indicating the signaling endpoint with the help of scoping (for example with domain name or a "local network only" flag). Scoping means that the signaling message although addressed to a particular destination IP address terminates somewhere along the path. If packet filters for both directions have to be installed then the signaling messages have to make packet filter installations up- and downstream along the data path. Similar to proposals in the area of QoS signaling some problems are likely to occur. One such problem is that downstream signaling in general causes problems because of asymmetric routes. In particular it is difficult to determine the firewall where the downstream data traffic will enter a network. The problem of triggering downstream reservations is for example described in [18] . Another problem for example is the placement of a firewall or NAT along the path other than in the access network. This would prevent a successful data exchange.

The following behavior has to be assumed in order for this approach to be applicable:

1. It must be possible to trigger a signaling message exchange for a downstream signaling message exchange at the firewall where the data traffic enters the network.

2. No other firewalls or NATs are present along the path other than in the access network.

This approach suffers from the following drawbacks:

1. To signal policy rules only within the access network (by both end-points) has a number of disadvantage and challenges (see for example [18]). The complex message processing caused by this approach strongly argues against it although it might sound simple (and even might be simple in restricted environments).

2. Complex topologies might lead to ineffective policy rules (i.e. data traffic hits firewalls hits wrong firewalls).

D.3 Solution Approach: Authorization Tokens

The last approach is based on some exchanged authorization tokens which are created by an authorized entity (such as the PDP) or by a trusted third party. Both end hosts need to exchange these tokens

with protocols such as SIP or HTTP since these protocols are likely to be allowed to bypass the firewall. The basic idea of this approach is to provide an end host, which requests access to the network, with credentials (referred as authorization tokens). These tokens have to possess some properties, namely:

1. They have to be restrictive by including lifetimes, source and destination identifiers, usage indication and more.

2. They have to provide basic replay protection to prevent unauthorized reuse.

3. The have be cryptographically protected to prevent manipulations.

4. There has to be a mechanism to dynamically create them for a specific reason and to distribute them to the end points.

5. It has to be possible to exchange tokens via a trusted third part in cases where no direct communication between the end hosts is possible (due to NAT).

6. The token can be created locally at the network or by a trusted third party.

An example of a possible signaling communication could have the following structure: After exchanging the tokens between the two end hosts. Host A would include the received authorization token to the signaling message for Network B. When the signaling message arrives at Middlebox 2 then the token is verified by the token-creating entity. In order to prevent parties from reusing the token timestamps (e.g. token creation, token lifetime, etc.) have to be included. Adding IP address information about Host A would create difficulties in relationship with NATs. Information about Host B might be possible to include in order to limit attacks where a token is lost and reused by a different host for a different purpose. The goal is to restrict the usage of the token for a specific session. The content of the token only needs to be verified by the originator of the token since it only has to be verified locally. Since authorization needs to be linked to the authorized actions, which have to be performed on the packets matching the packet filter, the token may include the associated action or a reference to it. The following behavior has to be assumed in order for this approach to be applicable:

1. The exchange of authorization tokens between end-systems must be possible. These protocols must be allowed to pass the firewalls.

2. An end-system must be able to request such an authorization token at some entity in the local network or at a trusted third party.

This approach suffers from the following drawback:

1. Possibly an additional protocol is required for an end host to request an authorization token from an entity in the local network.

Appendix E. Acknowledgments

We would like to acknowledge Vishal Sankhla and Joao Girao for their input to this draft.

Intellectual Property Statement

Full Copyright Statement

Acknowledgment

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