Internet Engineering Task Force
INTERNET-DRAFT
draft-tschofenig-nsis-casp-midcom-01.ps

H. Tschofenig, H. Schulzrinne, C. Aoun
Siemens/Columbia U./Nortel Networks

March 3, 2003
Expires: September 2003

A Firewall/NAT Traversal Client for CASP

Status of this Memo

Copyright Notice

Abstract

This document describes a CASP client protocol that allows nodes to signal information to firewalls both in an in-path and off-path fashion. The protocol furthermore allows to establish a NAT binding and to provide the signaling initiator with the NAT information. This is information can then be used within other protocols such as SIP.

Contents

1 Introduction

CASP-NATFW is a client protocol for the Cross-Application Signaling Protocol (CASP) [1]. It is one of a family of CASP client protocols and allows the signaling of firewall and NAT information along the data path (in-path) in a topology independent manner. CASP-NATFW aims to address issues raised and not solved within the MIDCOM working group [2] and uses ideas for in-path signaling using RSVP as described in [3] and in [4].

2 Definitions

Terms in context with trust relationships are described in [5].

The following terms are used in this document:

Policy Rule: The term policy rule is used as defined in [6]. A policy rule consists of two components: a packet filter and an action to be performed on packets matching the packet filter expression. This document uses two actions for a policy rule: allow without logging and allow with logging. Per-default no logging is used. If logging is desired then it has to be specified as described in Section 9. As stated in [6] it was agreed not to specify a deny action for policy rule. Hence there is no such deny action defined in this document.

In the context of NAT, as defined in [7], basic NAT, NAPT and twice NAT could be applied to packet flows matching the packet filter.

Policy Groups: The term policy group is not used in this document since its meaning is partially captured by the packet filter. A packet filter allows various attributes (even lists and ranges of certain attributes) to be specified. In case of in-path signaling only one particular destination IP address (which is available in the CASP NTLP payload) can be specified. More fine grain packet filters have to be specified in the CASP NSLP payload (in this case CASP-NATFW). For off-path signaling this rule must not hold.

Lifetime of Policy Rules: An NSLP is allowed to specify the lifetime for policy rule. The lifetime corresponds to the refresh interval. If no lifetime or refresh interval is selected then a default value is used.

Packet filter (PF): The term packet filter refers to attributes describing subsets of the data traffic for which a specific behavior should be provided. The term flow identifier is also often used in the area of QoS signaling protocols. For NAT traversal a packet filter (or flow identifier) refers to a NAT binding.

The terms in-path (off-path) signaling can be used inter-changable with path-coupled (path-decoupled) signaling.

3 General Limits of In-Path Firewall Signaling

At the beginning of this document it is worth stating that the problem of firewall signaling is addressed by a number of working groups. This section provides a brief overview of some of the recent activities and describes the general limits of in-path firewall signaling.

The following working groups or activities at the IETF have a relationship to policy rule installation and firewall communication in general:

To address a single device at the borders of the access networks (i.e. the first IP device) is covered by the PANA working group to implement the controlled/uncontrolled network access procedures. Thereby authentication of a user or a device with the help of EAP is required to create policy rules at the first IP device. This subsequently allows the end host to forward packets to the Internet or to access services within the local domain.

The MIDCOM working group also aims to install policy rules at firewalls. However, their approach seems to be focused on off-path signaling. Additionally of interest are activities related to Endpoint Firewall Controll, RSIP and Socks.

To provide a generic solution to install state at a possibly large number of firewalls along the path some trust must be placed on devices along the path. If no such trust is available which might be likely the case in an adhoc network scenario as shown in Figure 1 then firewall signaling is doomed to fail.

An adhoc networks 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 unauthenticated. To prevent this type of threat protocols developed in the IPSEC or the IPSRA working group, which establish a security association to protect the data traffic, can be used.

To summarize: In many cases in-path policy rule installation might provide enough security protection to prevent unauthorized nodes from gaining access to network resources. However, due to the absence of per-packet authentication man-in-the-middle attacks of malicious nodes along the path cannot be prevented by installed policy rules.

```
+-------------------------------------------+        +--------//
|                             Adhoc         |        | ISP
|                             Network       |        |
|      regular data                         |        |
|      traffic by          +----------+     |        |
|      node A              |Malicious|      |      +-+--------+
|          +-------------->+  Node    +-----+///-->+ Firewall +-//
|          ^               |    3     |===========>|    1     |
|          |               +----------+ injected   +-+--------+
|          |                            data traffic |
|          |                                |        |
|          |                                |        |
|      +---+-----+        +----------+      |        |
|      +  Node   |        |  Node    |      |        |
|      |    1    |        |    2     |      |        |
|      +---------+        +----------+      |        |
|          ^                                |        +--------//
|          |                                |
+----------+--------------------------------+
           |
        +--+---+
        | Node |
        |  A   |
        +------+
```

Figure 1: General Limits of In-Path Firewall Signaling

4 Trust Relationships

It is unusual to start a protocol description with trust relationships to explain the basic protocol behavior. A protocol for firewall traversal is somewhat different since trust relationships are very important for the protocol design and NAT traversal does not cause problems to the same degree. for its internal mechanisms.

4.1 Peer-to-Peer Trust Relationship

The following scenarios can be considered as the simplest since peer-to-peer trust relationship exist between the participating entities. These trust relationships are either direct or indirect and help to establish security associations dynamically (for example between Host A and the local middlebox i.e. Middlebox 1 within Network A) with the help of an authentication and key exchange protocol. Authentication and authorization of the request to the middlebox device is thereby required for successful protocol completion. Important in this context is the trust relationship between the two networks (i.e. between Middlebox 1 and Middlebox 2). In this scenario it is assumed that no firewall is present within the core network. In case that Middlebox requires authentication of the Host A (or from the user located at Host A) then an "Authentication Required" RESPONSE message with an error code is returned to the initiator. In case of a sender-initiated signaling message transmitted by Host A the requested filter entries at the first middlebox are already installed when the request at the subsequent middlebox fails.

Since end hosts usually do not have (and should not have) topology information of the networks along the path it is not possible to transmit policy rules for both directions (if data traffic later flows in both directions). Hence it is required that both nodes transmit separate signaling messages for each direction containing separate policy rules for each traffic flow (if the data traffic is later sent in both directions). These signaling messages are transmitted by the end hosts and they do not need to travel along the same path because of asymmetric routes (see [8]. Therefore the signaling message which is triggered from the two end hosts in each direction do not necessarily need to install state at the same firewall.

Policy rule installation is based on the information transmitted with the flow identifier object at the CASP NTLP layer and at the packet filter object at the NATFW NSLP layer. The content of both objects might change mid-path (for example when passing a NAT) and is allowed to change mid-session (for example because of mobility). For those cases where the information carried within a packet filter object cannot be interpreted an error message is returned indicating the inadequate information. Packet filter processing failures are possible when for example a Virtual Private Network Identifier such as (Extended) Tunnel ID is transmitted to an IP firewall or when a firewall is unable to install a packet filter with the indicated granularity.

```
+--------------------------+              +--------------------------+
|                          |              |                          |
|              Network A   |              |             Network B    |
|                          |              |                          |
|                +---------+              +---------+                |
|          +-///-+ Middle- +---///////----+ Middle- +-///-+          |
|          |     |  box 1  |   Trust      | box 2   |     |          |
|          |     +---------+ Relationship +---------+     |          |
|          |               |              |               |          |
|          |               |              |               |          |
|          |               |              |               |          |
|          |   Trust       |              |     Trust     |          |
|          | Relationship  |              | Relationship  |          |
|          |               |              |               |          |
|          |               |              |               |          |
|          |               |              |               |          |
|       +--+---+           |              |            +--+---+      |
|       | Host |           |              |            | Host |      |
|       |  A   |           |              |            |  B   |      |
|       +------+           |              |            +------+      |
+--------------------------+              +--------------------------+
```

Figure 2: Peer-to-Peer Trust Relationship

4.2 Intra-domain Trust Relationship

In larger corporations often more than one firewall 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 however it might be necessary to preserve authentication and authorization information within the network. In this case an interaction between the individual middleboxes and a central entity (for example a policy decision point - PDP) might be required. Otherwise it is possible to communicate the authorization decision made at one firewall to another firewall within the same trust domain. Each middlebox can either communicate with the PDP or the PDP issues an authorization token which allows the middleboxes to react independently. Figure 3 refers to this structure. To avoid complex protocol interactions individual middleboxes within an administrative domain should make use of their trust relationship instead of requesting authentication and authorization of the signaling initiator again. This provides both a performance improvement without a security disadvantage since a single administrative domain can be seen as a single entity.

```
+---------------------------------------------------------------+
|                                                               |
|                                                  Network A    |
|                                                               |
|                                                               |
|                          +---------+                +---------+
|          +----///--------+ Middle- +------///------++ Middle- +---
|          |                |  box 2  |                |  box 2  |
|          |                +----+----+                +----+----+
|          |                     |                          |    |
|     +----+----+                |                          |    |
|     | Middle- +--------+       +---------+                |    |
|     |  box 1  |        |                 |                |    |
|     +----+----+        |                 |                |    |
|          |             |                 |                |    |
|          -             |                 |                |    |
|          -             |            +----+-----+           |    |
|          |             |            | Policy   |           |    |
|       +--+---+         +------------+ Decision +-----------+    |
|       | Host |                      | Point    |                |
|       |  A   |                      +----------+                |
|       +------+                                                  |
+---------------------------------------------------------------+
```

Figure 3: Intra-domain Trust Relationship

4.3 Required 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 some authentication 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 a RESPONSE message with an error code "Authentication Required" is returned. However, in many cases the user initiating the signaling message exchange is already aware of such a constraint and received credentials before the message exchange was started. These credentials might be based either on symmetric (shared secret) or based on asymmetric (public key) cryptography. In order to avoid a challenge/response type of message exchange the initiating node (Host A in our scenario) already includes a CMS object to the outgoing signaling message. The CMS object contains the identity of the signaling initiator, the identity of the destination network, the destination address of Host B, a timestamp for replay protection and possibly some other application specific information like an application identifier. CMS allows to use both symmetric and asymmetric credentials.

Figure 4 shows the slightly more complex trust relationships in this scenario.

```
+--------------------------+              +--------------------------+
|                          |              |                          |
|              Network A   |              |             Network B    |
|                          |              |                          |
|                          | Trust        |                          |
|                          | Relationship |                          |
|                +---------+              +---------+                |
|          +-///-+ Middle- +---///////----+ Middle- +-///-+          |
|          |     |  box 1  |      +-------+  box 2  |     |          |
|          |     +---------+      |       +---------+     |          |
|          |               |      |       |               |          |
|          |Trust          |      |       |               |          |
|          |Relationship   |      |       |               |          |
|          |               |      |       |   Trust       |          |
|          |               |      |       |   Relationship|          |
|          |               |      |       |               |          |
|          |               |      |       |               |          |
|          |               |      |       |               |          |
|          |               |      |       |               |          |
|       +--+---+           |      |       |            +--+---+      |
|       | Host +----///----+------+       |            | Host |      |
|       |  A   |           |Trust         |            |  B   |      |
|       +------+           |Relationship  |            +------+      |
+--------------------------+              +--------------------------+
```

Figure 4: End-to-Middle Trust Relationship

4.4 Missing Network-to-Network Trust Relationship

Peer-to-peer trust relationship as shown in Figure 2 is a very convenient assumption that allows simplified signaling message processing. However it is obvious that such an assumption does not always hold. Especially the trust relationship between two arbitrary non-adjacent access networks is likely non-existent 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.

```
+-------------------------+             +-------------------------+
|                         |             |                         |
|             Network A   |             |             Network B   |
|                         |             |                         |
|               +---------+  Missing    +---------+               |
|         +-///-+ Middle- |   Trust     | Middle- +-///-+         |
|         |     |  box 1  |  Relation-  |  box 2  |     |         |
|         |     +---------+    ship     +---------+     |         |
|         |               |             |               |         |
|         |               |             |               |         |
|         |               |             |               |         |
|         |   Trust       |             |      Trust    |         |
|         | Relationship  |             |  Relationship |         |
|         |               |             |               |         |
|         |               |             |               |         |
|         |               |             |               |         |
|      +--+---+           |             |            +--+---+     |
|      | Host |           |             |            | Host |     |
|      |  A   |           |             |            |  B   |     |
|      +------+           |             |            +------+     |
+-------------------------+             +-------------------------+
```

Figure 5: Missing Network-to-Network Trust Relationship

We identified three possible approaches of tackling the problem described in Figure 5.

Receiver-Initiated Signaling: The first approach makes use of receiver-based signaling message exchange. If Host A sends a signaling message toward the destination to Middlebox 1 the message can be properly protected. Middlebox 1 establishes some state information and expects an incoming message initiated by Host B. Signaling message protection between the two networks is difficult. A missing trust relationship does not necessarily mean that no security association establishment is possible. The lacking trust "only" disallows Middlebox 1 to create packet filters at Middlebox 2. Hence, this missing trust is an authorization problem rather than a security association establishment problem. If the CASP message itself is allowed to pass the firewall then it finally reaches Host B. Host B should not experience any difficulties to install filters at the local firewall (Middlebox 2). The message is then forwarded to Middlebox 1 which already waits for the incoming signaling message. Because it is possible to associate existing state information at Middlebox 1 with the incoming message packet filters are installed and the message is finally forwarded to Host A. Authorization for packet filter installation in Network A has to be provided by Host A and for Network B has to be provided by Host B when returning the response message. packet filters are installed for data traffic 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:

- Signaling messages must be allowed to pass firewalls along the path. No authorization for packet filter installation is required at this stage. Blocking signaling messages at firewalls disallows the receiver of the signaling message to return a signaling message.
- The signaling message initiated by the NI will require state installation on all the NFs in the path (if a RSVP PATH message semantic is assumed). CASP NTLP, however, also allows a stateless signaling message routing.

This approach suffers from the following drawbacks:

- If the CASP signaling messages (in this case the "PATH" message) is not allowed to bypass a firewall then no policy rules are created at any node along the path.
- Receiver-initiated signaling has the advantage that the receiver has to accept the creation of the policy rule in his own network to trigger the creation locally. This seems to simplify security processing. If a NAT is present then still a RESPONSE message is required to inform the data message sender about the NAT-binding (i.e. the IP addresses and port information seen by a data traffic receiver).

Access Network-Only Signaling Message Exchange The next approach is based on signaling packet filter information by both hosts into the local access network only. CASP allows to specify 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 [9]. 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:

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

This approach suffers from the following drawbacks:

- To signal policy rules only within the access network (by both end-points) has a number of disadvantage and challenges (see for example [9]). 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). Section 10 addresses message flows for this case. Although its usage is possible with CASP we strongly discourage its usage.
- Some circumstances can lead to ineffective policy rules.

Authorization Tokens: The last approach is based on some exchanged authorization tokens which are created by an authorized entity (such as the PDP) in each access network. Both hosts need to exchange these tokens with some protocols such as SIP or HTTP which is more likely allowed to bypass the firewall. Host A would then 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:

- The exchange of authorization tokens between end-systems must be possible. These protocols must be allowed to pass the firewalls.
- An end-system must be able to request such an authorization token at some entity in the local network.
- The hosts need to have each other's addresses, which is complicated to have if there are NATs deployed on the path (especially with double NAT).

This approach suffers from the following drawback:

- An additional protocol is required for an end host to request an authorization token from an entity in the local network as depicted in Section 10. Note that CASP could be extended to provide this functionality but currently it does not.

4.5 Off-Path Signaling

The separation between signaling message delivery and discovery within at the CASP NTLP protocol allows it to support in-path and off-path signaling easily with the same protocol. Throughout this document in-path signaling was assumed (the Scout protocol is used per-default for next peer discovery) but off-path signaling might be desired in some scenarios where a third-party entity wants to signal some policy rules to a firewall and NATs. This mechanism has disadvantages in larger networks with multiple firewalls since topology information is required in order to install policy rules on the traversed firewalls and NATs.

5 Assumptions

Based on the above-described trust relationships the following protocol assumptions have to be made.

- Middleboxes along the path are CASP-aware. If a middlebox is not CASP-aware then protocol functionality cannot be fully guaranteed (especially if the middlebox cannot be controlled with the help of some protocol at all). The CASP-NATFW NSLP protocol can operate with limitations if a CASP-unaware firewall blocks all CASP signaling traffic. To support CASP-unaware NATs along the path some information needs to be added to a CASP-NATFW message to allow the signaling message receiving entity to verify that the source ip address (and port numbers) have changed. Currently no such object is included in this version of the document.

- The end host should not be required to know the topology of the networks along the path or some other network internal issues. Therefore it is not possible to make an assumption about routing and hence we have to assume asymmetric routes. As a consequence end hosts include unidirectional packet filters only. Within a administrative domain where more information is available this assumption might not hold and the establishment of bi-directional packet filters could be possible.

6 NAT Involvement

Two issues need to be addressed when NATs are present along the path. Since the end host should not a-priori knowledge about the location, number and types of NATs along the path their presence has to be assumed.

First, the CASP signaling messages itself must be able to traverse a non-CASP aware NAT box without major problems. A NAT binding of a non-CASP aware NAT can be established and maintained much easier with TCP than with UDP. CASP recommends the usage of transport protocols such as TCP or SCTP In case that the NAT is CASP-aware problems only occur if source port numbers are fixed. CASP does not require fixed source port numbers to be used.

The second issue addresses data packets for which a NAT binding needs to be requested. When an end host starts to transmit scout packets to discover the presence of firewalls and NATs along the path it is willing to subsequently transmit data packets which match the packet filter. Subsequently such a firewall-NAT-firewall scenario is described to explain the basic protocol interaction and the usefulness for allowing packet filters to change mid-path (i.e. along the path). Mid-session changes of packet filters happen in mobility cases (for example if the end host obtains a new care-of address).

```
+--------------------------------------------------------------+
|                                                              |
|                                                  Network A   |
|                                                              |
|    PF=(192.168.1.5;              PF=(139.23.203.30;          |
|        tcp+udp;666)                  tcp+udp;7000)           |
|                         +----------+              +----------+
|          +----///------>+   NAT    +------///----->+ Firewall +-->
|          |              |    1     |              |    2     |
|          |              +----------+              +----------+
|          |                                                   |
|     +----+-----+                                             |
|     | Firewall |                                             |
|     |    1     |                                             |
|     +----+-----+                                             |
|          ^                                                   |
|          -  PF=(192.168.1.5;                                 |
|          -      tcp+udp;666)                                 |
|          |                                                   |
|       +--+---+                                               |
|       | Host |                                               |
|       |  A   |                                               |
|       +------+                                               |
+--------------------------------------------------------------+
```

Figure 6: NAT Involvement

In Figure 6 a hosts (Host A) wants to enable transmit data traffic from source IP address 192.168.1.5 to a given destination IP address (not shown in the Figure 6) at port 666 (both udp and tcp). Therefore Host A transmits a CASP-NATFW message to Firewall 1 (after discovering that this firewall is the next CASP supporting node along the data path) to create the corresponding packet filters. Note that the traffic selector is unidirectional. This scenario shows a sender-initiated scenario. Firewall 1 installs two policy rules (one for udp and the other one for tcp) after successful authentication and authorization. After forwarding to the next middlebox (a NAT in this case) a NAT binding has to be created for the given traffic selectors. The externally visible packet filter (IP address changed to 139.23.203.30 and port number=7000) is then forwarded to the next firewall (Firewall 2). Firewall 2 again creates policy rules after authentication and authorization. Then the signaling message is forwarded towards the destination.

After the signaling messages reaches the destination IP address or until no further firewall can be reached (for example because the message is rejected at a non CASP-aware firewall) a RESPONSE message is returned (if requested by the signaling message initiator). A RESPONSE message would contain a Status object which includes information about the applied packet filter and whether the message reached its target or not. In case of NATs along the path the packet filter information is then included in protocols like SIP to communicate on which protocol/port data will be sent.

In case no RESPONSE message is sent back, the CASP-NATFW aware NFs on the path will return a RESPONSE message with an unsuccessful end to end message delivery error when an associated timer to the existing installed state (relevant to the reception of the CREATE message) expires. The CASP-NATFW NI will receive only one RESPONSE message it may receive more than one in particular cases. It is up to the NI to decide if it has to proceed with the application or not. Every CASP-NATFW on the path will need to filter out the associated RESPONSE, messages to the same original CREATE message, sent by the CASP-NATFW NFs on the upstream. In case a RESPONSE message provides a different filter within the installed policy rule attribute, the RESPONSE message will be forwarded on the downstream towards the CASP-NATFW aware NI.

Section 10 additionally addresses some message flows with NAT involvement.

7 Operation

CASP-NATFW defines the following message types:

Path: A PATH message allows a receiver-initiated reservation approach. This message does not cause packet filters to be installed although all objects are present. This message is then used as a trigger to cause a CREATE to be returned. The PATH message transmitting entity includes the objects which are later used (if not modified) by the sender of the CREATE message. The PATH message allows receiver-initiated signaling to be supported.

Create: A CREATE message allows to establish or update NSLP state (i.e. policy rules) at one or more firewall(s) along the path. Verification is necessary to ensure that policy rule creation is allowed by the requesting entity and that no other local security policy is violated. In case a security policy is violated or the creation of the policy rule(s) is not permitted, a RESPONSE message with a "Security Policy Violated" error code is returned. If the CREATE message is used without a previous PATH message then it represents a typical sender-initiated reservation.

Release: A RELEASE message is used to delete installed NSLP state at a firewall and to release a NAT binding without waiting for a soft-state timeout. This message can only delete previously installed state. Referring to previously installed state can easily be done using the session identifier. Only authorized parties are allowed to delete installed state, this includes the creator of the state or other parties trusted by the state creator (useful for fail over of the state creator).

Response: A RESPONSE message is either sent to acknowledge a previous message or to indicate an error. In case of an acknowledgement it is required that the signaling message initiator requests the transmission of a response message. Therefore the Next object, discussed in Section 9, is set to the Response message. No state information is modified by processing and forwarding an acknowledgement. If an error has to be returned then the error code inside the RESPONSE message allows to specify more detailed error information. Such an error code might for example indicate missing user specific credentials, a missing authorization token or a security policy violation. Detailed error codes have to be defined in future versions of this document.

Query: A QUERY message triggers a RESPONSE message to return installed state information. The main purpose of this message is to provide diagnostic facilities. An initiator must only be able to query owned state information. Otherwise the entire set of policy rules of a firewall could be retrieved which causes security concerns. An adversary would have a simple mechanism to retrieve a lot of useful information for subsequent attacks.

Trigger: The TRIGGER message is an asynchronous event notification sent by a CASP-NATFW aware node. Unlike the CREATE message it does not create or modify NSLP state at nodes between the initiator and the target of the TRIGGER. As a difference to the PATH message also no NTLP routing state is created at nodes between the initator and the target of the TRIGGER. Some sort of trigger message is required to support access network signaling message exchanges as described in Section 10 and in Section 4.4. (TBD: This usefulness of this message or other technical alternatives require some investigation.)

The following table shows the basic message behavior whereby the following abbreviations are used: MAY (O), MUST NOT (–), MUST (M) or NA (Not Applicable))

The operations specify which message might indicate information to trigger which other message in response by the other end. Some messages (such as an error message) are created automatically without previous indication.

Msg/Next Msg	Path	Create	Release	Response	Query	Trigger
Path	NA	M	–	O	–	–
Create	O	O	–	O	–	–
Release	–	–	O?	O	M	–
Response	–	–	–	NA	–	–
Query	–	–	–	M	NA	–
Trigger	O	O	O?	–	–	NA

Note that the ”Must” entries in the table above indicate only the default behavior. For example: A PATH message must be followed by a CREATE message. However in case of an error a RESPONSE message (with an error code) will be returned.

The following issues still require some investigations:

- To enable a bi-directional reservation the sender of a CREATE message has to indicate either another CREATE message in the Next object or a PATH message. It is questionable whether a sender-initiated signaling message should follow a receiver-initiated?

- Is it useful to allow a RESPONSE or a RELEASE message to follow a RELEASE message?

8 Typical Policy Rule Attributes

This paragraph describes some typically used attributes. Other attributes such as flow labels might be used but are considered as an exception of the packet filter. We believe that a granularity at transport layer protocol state-level (syn, syn/ack, ack, etc.) is not required for in-path signaling.

- Source/destination IPv4 and IPv6 addresses
- Port numbers (possibly including ranges and a list of port numbers)
- Transport protocol (for example TCP, UDP)
- SPI (for IPSec protected data traffic)
- Identifiers for AH and ESP (Protocol numbers, next headers fields)

A NAT object returned to the signaling message initiator contains the same attribute types. The NAT object is included as a payload in the Status object. A signaling message originator may also use the NAT object to request a particular NAT binding to take place. The same object is used for this purpose.

There are only two actions defined for a policy rule: "allow / no logging" (default) and "allow / logging". The first action does not require additional objects to be included other than the packet filter. This is the default action. If a "allow / logging" action has to be specified then the Logging Action object, defined in 9, has to be included. This action creates log entries whenever the rule was triggered. End hosts are usually not allowed to specify this behavior because it could be used for a denial of service attack to cause log files to grow quickly and without bounds.

Note that a single packet filter might also specify a range or ports. Furthermore it is also possible to specify more than one policy rule within a single signaling message (e.g. for off-path signaling). This issue, however, requires further investigation.

9 Objects

The following objects are used by the CASP-NATFW client protocol:

9.1 Logging Action

This object indicates which packet filter(s) want to have logging specified. Note that end host are usually not allowed to specify this behavior for in-path signaling. It might however be requested within the network or in case of off-path signaling. (TBD: Some investigation is required to evaluate whether this action is really required.)

9.2 ApplicationID

This object contains an identifier to provide more information about the data for which the policy rule is installed. Application-level firewalls and firewalls with stateful inspection are able to use this information.

Providing a wrong application identifier for a given data traffic would then cause a processing failure. Such a behavior is more secure than a traditional packet filter firewall. Note however that encrypted end-to-end traffic might reduce this advantage to some degree. A local security policy might indicate that this information is required before creating policy rules. A missing ApplicationID object would then cause a "Application ID require" RESPONSE message with an error code is returned.

9.3 Next

The Next object indicates the next request that the signaling message receiver should generate if the incoming message was successfully processed. Section 7 shows possible combinations of messages. For example, a CREATE message might contain a Next object which is set to CREATE causing another create message to be returned. Such a message flow would represent a bi-directional reservation. A frequently used object is the response object providing indications about a previously submitted message.

9.4 Authorization Token

This object is used as described in Figure 5 of Section 4.4. More description will be added in the near future (see Section 13).

9.5 CMS Credential Object

This object allows user specific cryptographic credentials to be transmitted to specific CASP peers (or networks) along the path. Figure 4 describes a scenario where such an object is required. Attributes included in this object are also briefly mentioned in Section 4.3.

9.6 Time

This object indicates that filters should be installed somewhere in the near future. This might be required in the context of in-advance QoS reservation for a conferencing scenario. If this object is not present, the current time is used.

9.7 Age

The Age object is used to quickly determine whether any of the NSLP object has changed (for example packet filter), to avoid a bit-by-bit comparison. The Age object might be useful for messages which refresh established state information only. Uniqueness of the Age object is only required only within a session. Whenever state information has to be modified then a new value has to be placed in the Age object. A high-resolution timestamp is typically used for this purpose.

9.8 Status

The Status object is used to deliver status information inside the RESPONSE message. This object might return error notifications or information about installed packet filters (e.g. NAT-Object). Delivering packet filter information is helpful for application (such as SIP, H323, MEGACO, MGCP etc.) that need to deliver IP address, protocol type and port information to the initiator in case of NATs along the path.

10 Basic Protocol Behavior

The following message flows try to show the basic protocol behavior and possible combinations regarding sender- and receiver-initiated messages flows, uni-directional or bi-directional packet filters, different trust assumptions and NAT and/or firewall traversal. The subsequently shown figures do not include message flows for next-peer discovery (for example using the Scout protocol).

10.1 Receiver-Initiated Message Flow with Firewalls

The following message flow shows the protocol behavior in case of a receiver-initiated signaling message exchange with two administrative domains (Network A and B) and two firewalls located at the borders. For the message flow a peer-to-peer trust relationship is assumed. Cryptographic credentials which support end-to-middle authentication (Host A-to-FW 2) can be included by Host A into the PATH message. The usage of receiver-initiation has the advantage that Host B has to assist in policy rule installation at Firewall B.

In Figure 7 the sender indicates in the PATH message which policy rule to install by adding this information to the packet filter. Host A uses the IP address 139.23.203.23 and the destination IP address (Host B) is 17.12.23.5. Note that the transport protocol is not mentioned since it is not helpful. The first firewall (FW 1) installs the indicated policy rule (packet filter with "allow / without logging" action). The message is forwarded to the next CASP aware node (FW 2). Because of the peer-to-peer trust assumption FW 2 trusts FW1 for the correctness of the provided parameters. The identity of the signaling message originator might be included in the signaling messages addressed toward the other end host. Policy rules are installed at both firewalls. When the signaling message reaches Host B then a CREATE message is returned in response and includes the same packet filter (unmodified). Note that the packet filter is always directional (especially for the CREATE message in response to a PATH message this is applicable). The CREATE message installs the policy rules at the two firewalls. The CREATE message finally reaches Host A who can immediately start to transmit data traffic towards Host B.

The following issues arise with the description of the message flow of Figure 7:

```
+--------------------------------+       +------------------------------+
| +------+            +-------+ |       | +------+         +--------+ |
| |Host A|  Network   | FW 1  | |       | | FW 2 | Network | Host B | |
| +--+---+     A      +--+----+ |       | +--+---+    B    +---+----+ |
+-+--+-------------------+------+       +----+-----------------+----+-+
  |Path(PF=              |                   |                      |
  |(src=139.23.203.23,   |                   |                      |
  | dst=17.12.23.5,      |                   |                      |
  | sport=5000,          |                   |                      |
  | dport=600)           |                   |                      |
  |--------------------->|                   |                      |
  |                      |Path(PF=           |                      |
  |                      |(src=139.23.203.23,|                      |
  |                      | dst=17.12.23.5,   |                      |
  |                      | sport=5000,       |                      |
  |                      | dport=600)        |                      |
  |                      |------------------>|                      |
  |                      |                   |Path(PF=              |
  |                      |                   |(src=139.23.203.23,   |
  |                      |                   | dst=17.12.23.5,      |
  |                      |                   | sport=5000,          |
  |                      |                   | dport=600)           |
  |                      |                   |--------------------->|
  |                      |                   |Create(PF=            |
  |                      |                   |(src=139.23.203.23,   |
  |                      |                   | dst=17.12.23.5,      |
  |                      |                   | sport=5000,          |
  |                      |                   | dport=600)           |
  |                      |                   |<---------------------|
  |                      |Create(PF=         |                      |
  |                      |(src=139.23.203.23,|                      |
  |                      | dst=17.12.23.5,   |                      |
  |                      | sport=5000,       |                      |
  |                      | dport=600)        |                      |
  |                      |<------------------|                      |
  |Create(PF=            |                   |                      |
  |(src=139.23.203.23,   |                   |                      |
  | dst=17.12.23.5,      |                   |                      |
  | sport=5000,          |                   |                      |
  | dport=600)           |                   |                      |
  |<---------------------|                   |                      |
  |                   Data Traffic (unidirectional)                 |
  |================================================================>|
```

Figure 7: Receiver-Initiated Message Flow with Firewalls

- Should packet filter information included in the PATH and CREATE message. packet filter information in the PATH message could be temporarily stored at middleboxes (firewalls in this example). The CREATE message would then only refer to existing state information.
- It does not seem to be useful to have a stateless version of the PATH message. Do we want to support such a stateless version?
- If the Path message fails then no policy rules are installed. The signaling message flow has to be restarted.

Figure 7 does not contain NATs, micro-/macro-mobility specific message flows or any form of tunneling. Hence no mid-path packet filter modification is necessary, otherwise such a packet filter modification would be required. Entities, which are aware of micro-/macro-mobility protocols (for example a MAP or a home agent), are no middleboxes in the traditional sense. Since they have an impact on the packet filter and on the data traffic it would be necessary to treat them as artificial middlebox to properly address flow identifications along the path. If no such treatment takes place then the wrong policy rules are installed at firewalls with the consequence that the entire protocol interaction is useless. In this description we assume that packet filter attributes are based on information used for routing (i.e. IP addresses).

10.2 Sender-Initiated Message Flow with Firewalls

The following message flow shows the protocol behavior in case of a sender-initiated signaling message exchange with two administrative domains (Network A and B) and two firewalls (FW 1 and FW 2). No NAT and other devices requiring modifications to the packet filter are used. This message flow also assumes a peer-to-peer trust relationship. Cryptographic credentials which support end-to-middle authentication (Host A-to-FW 2) can be included by Host A into the CREATE message.

```
+------------------------------+        +-----------------------------+
| +------+            +------+ |        | +------+         +--------+ |
| |Host A|  Network   | FW 1 | |        | | FW 2 | Network | Host B | |
| +--+---+     A      +--+---+ |        | +--+---+    B    +---+----+ |
+-+--+-------------------+-----+        +----+-----------------+----+-+
  |Create(PF=            |                   |                      |
  |(src=139.23.203.23,   |                   |                      |
  | dst=17.12.23.5,      |                   |                      |
  | sport=5000,          |                   |                      |
  | dport=600)           |                   |                      |
  |--------------------->|                   |                      |
  |                      |Create(PF=         |                      |
  |                      |(src=139.23.203.23,|                      |
  |                      | dst=17.12.23.5,   |                      |
  |                      | sport=5000,       |                      |
  |                      | dport=600)        |                      |
  |                      |------------------>|                      |
  |                      |                   |Create(PF=            |
  |                      |                   |(src=139.23.203.23,   |
  |                      |                   | dst=17.12.23.5,      |
  |                      |                   | sport=5000,          |
  |                      |                   | dport=600)           |
  |                      |                   |--------------------->|
  |                      |                   |      [Response]      |
  |                      |                   |<---------------------|
  |                      |    [Response]     |                      |
  |                      |<------------------|                      |
  |      [Response]      |                   |                      |
  |<---------------------|                   |                      |
  |                   Data Traffic (unidirectional)                 |
  |================================================================>|
```

Figure 8: Sender-Initiated Message Flow with Firewalls

The message flow in Figure 8 is similar to Figure 7. The CREATE message contains the packet filter and immediately (after authentication, authorization and verification) causes the installation of policy rules. The signaling message sender might request a RESPONSE message. In case of NATs along the path such a RESPONSE message is very useful to return NAT binding information.

This scenario does not require packet filter modification along the path. No NAT binding is returned with the optional RESPONSE message.

The following issue arises with the description of the message flow of Figure 8:

- If a verification error is caused during the CREATE message processing then some firewalls might have installed policy rules whereas others have never seen the signaling message. A RESPONSE message indicating an error could leave installed state in place or cause already established state to be removed automatically.

10.3 Receiver-Initiated Message Flow with a Firewall and a NAT

The message flow in Figure 9 introduces a middlebox with NAT functionality (NAT 1), in addition to a firewall at Network B, along the path between Host A and Host B. Note that NAT 1 might additionally have firewall functionality which would require to install pinhole opening and NAT binding policy rules. The message flow assumes that Host A with source IP address 10.1.0.5 wants to transmit data traffic at source port 1200 (for example UDP / not shown in this example) to destination address 17.12.23.5 at destination port number 600. Host A does not requires a particular NAT binding, hence no NAT-Object is required within the initial PATH message. In any case a NAT binding will be included within the NAT-Object returned in the RESPONSE message. Instead the provided NAT binding is provided as a NAT-Object in response. If Host A would like to request a particular NAT binding then the NAT-Object has to be included in the initial PATH message.

As soon as the signaling message reaches NAT 1 a NAT binding is requested and the result of this request is placed into the Traffic selector field (i.e. src ip address is changed from 10.1.0.5 to 139.23.203.30 and the sport is rewritten from 1200 to 5000). When the signaling messages is successfully processed by FW 2 and forwarded to Host B a CREATE message with the indicated packet filter is returned. A copy of the received packet filter is placed into the NAT-Object. By returning the NAT-Object information, Host A is able to learn which IP address and port , hence no NAT-Object is required within the initial PATH message. In any case a NAT binding will be included within the NAT-Object returned in the RESPONSE message. The CREATE message is routed backwards toward Host A (since the path is pinned down).

The exchange of end-to-end messages after a successful signaling message exchange might be required to exchange parameters about the subsequent data traffic. Finally Host A starts to transmit data packets to Host B.

```
+-------------------------------+       +-----------------------------+
| +------+            +------+ |       | +------+         +--------+ |
| |Host A|  Network   | NAT 1| |       | | FW  2| Network | Host B | |
| +--+---+     A      +--+---+ |       | +--+---+    B    +---+----+ |
+-+--+-------------------+-----+       +----+-----------------+----+-+
  |Path(PF=              |                  |                      |
  |(src=10.1.0.5,        |                  |                      |
  | dst=17.12.23.5,      |                  |                      |
  | sport=1200,          |Path(PF=          |                      |
  | dport=600)           |(src=139.23.203.23,|                      |
  |--------------------->| dst=17.12.23.5,  |                      |
  |                      | sport=5000,      |Path(PF=              |
  |                      | dport=600)       |(src=139.23.203.23,   |
  |                      |----------------->| dst=17.12.23.5,      |
  |                      |                  | sport=5000,          |
  |                      |                  | dport=600)           |
  |                      |                  |--------------------->|
  |                      |                  |                      |
  |                      |                  |Create(PF=            |
  |                      |                  |(src=139.23.203.23,   |
  |                      |Create(PF=        | dst=17.12.23.5,      |
  |                      |(src=139.23.203.23,| sport=5000,          |
  |                      | dst=17.12.23.5,  | dport=600);          |
  |Create(PF=            | sport=5000,      | NAT-Object=          |
  |(src=10.1.0.5,        | dport=600);      |(src=139.23.203.23,   |
  | dst=17.12.23.5,      | NAT-Object=      | dst=17.12.23.5,      |
  | sport=1200,          |(src=139.23.203.23,| sport=5000,          |
  | dport=600);          | dst=17.12.23.5,  | dport=600))          |
  | NAT-Object=          | sport=5000,      |<---------------------|
  |(src=139.23.203.23,   | dport=600))      |                      |
  | dst=17.12.23.5,      |<-----------------|                      |
  | sport=5000,          |                  |                      |
  | dport=600))          |                  |                      |
  |<---------------------|                  |                      |
  |                      |                  |                      |
  |                For example: SIP Signaling                      |
  |<~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~>|
  |                      |                  |                      |
  |                Data Traffic (unidirectional)                   |
  |===============================================================>|
```

Figure 9: Receiver-Initiated Message Flow with a Firewall and a NAT

10.4 Sender-Initiated Message Flow with a Firewall and a NAT

Figure 10 shows a sender-initiated signaling message flow whereby FW 2 in Network B initially rejects the signaling message due to an authentication/authorization failure. The returned RESPONSE message includes among the error code, information about the entity creating the error (in this case FW2@NetworkB) and optionally a challenge value. The challenge value allows Host A to either provide a freshness guarantee based on the challenge value and/or based on a timestamp. The usage of CMS allows Host A and Network B to use symmetric and asymmetric credentials for authentication. In any case a Credential object is attached to the CREATE signaling message. The Credential object securely binds a timestamp or a sequence number (to prevent replay attacks), identities, lifetime and possibly packet filter information to the cryptographic credentials. The RESPONSE message might return a NAT-Object if a NAT was present along the path.

Host A retransmits a new signaling message. After verification of the request and the credentials FW 2 forwards the message to Host B. As in previous examples Host B returns a RESPONSE message with a NAT-Object back to Host A.

```
+------------------------------+        +------------------------------+
| +------+             +------+ |        | +------+         +---------+ |
| |Host A|  Network   | NAT 1| |        | | FW  2| Network | Host B  | |
| +--+---+     A      +--+---+ |        | +--+---+    B    +---+-----+ |
+-+--+-------------------+-----+        +----+-----------------+----+--+
  |Create(PF=            |                   |                      |
  |(src=10.1.0.5,        |Create(PF=         |                      |
  | dst=17.12.23.5,      |(src=139.23.203.23,|                      |
  | sport=1200,          | dst=17.12.23.5,   |                      |
  | dport=600)           | sport=5000,       |                      |
  |--------------------->| dport=600)        |                      |
  |                      |------------------>|                      |
  |                      |Response(ErrorCode=|                      |
  |Response(ErrorCode=   |"Auth. Required",  |                      |
  |"Auth. Required",     | FW2@NetworkB,     |                      |
  | FW2@NetworkB,        | challenge=0x7a..8,|                      |
  | challenge=0x7a..8,   |<------------------|                      |
  |<---------------------|                   |                      |
  |Create(PF=            |                   |                      |
  |(src=10.1.0.5,        |                   |                      |
  | dst=17.12.23.5,      |Create(PF=         |                      |
  | sport=1200,          |(src=139.23.203.23,|                      |
  | dport=600)           | dst=17.12.23.5,   |Create(PF=            |
  | Credentials(...))    | sport=5000,       |(src=10.1.0.5,        |
  |--------------------->| dport=600)        | dst=17.12.23.5,      |
  |                      | Credentials(...)) | sport=1200,          |
  |                      |------------------>| dport=600)           |
  |                      |                   |--------------------->|
  |                      |                   |    Response(         |
  |                      |  Response(        |    NAT-Object(...))  |
  |    Response(         |  NAT-Object(...)) |<---------------------|
  |    NAT-Object(...))  |<------------------|                      |
  |<---------------------|                   |                      |
  |                      |  SIP Signaling    |                      |
  |<~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~>|
  |                      |                   |                      |
  |                   Data Traffic (unidirectional)                 |
  |================================================================>|
```

Figure 10: Sender-Initiated Message Flow with a Firewall and a NAT

The message flow shows the following protocol features:

- End-to-Middle Authentication by including a CMS object (Credential object) to the signaling message after the authentication/authorization failure. If the Credential object is included into the first CREATE signaling message then no such error message is returned. However in that case replay protection can only be based on timestamps (loosely synchronized clocks).
- A NAT-Object is included in the RESPONSE message which provides information about the NAT binding.
- The RESPONSE message indicating an error could also return a NAT-Object to provide initial information about the existence of a NAT.
- The same protocol operations can be used without NATs (only firewalls).

10.5 Sender-Initiated NAT/Firewall Traversal with Authorization Token

The next scenario is slightly more complicated in the sense that authorization information for Network B is provided by Host B. Host B first request an authorization token from an entity in the local network by some means. This token is then communicated to Host A using an end-to-end protocol such as SIP or HTTP. This token then provides the necessary trust for Network B to allow the CREATE message to install policy rules at FW 2. Note that this message flow is different compared to the scenario described in Figure 10. In this case no pre-established cryptographic credentials between Host A and Network B are present before the protocol is used between Host A and Host B.

The sender-initiated message flow is similar to the above-described flows with the only exception that the Authorization Token is included. The token is removed at FW 2 after successful verification.

```
+------------------------------+        +-----------------------------+
| +------+            +------+ |        | +------+         +--------+ |
| |Host A|  Network   | NAT 1| |        | | FW  2| Network | Host B | |
| +--+---+     A      +--+---+ |        | +--+---+    B    +---+----+ |
+-+--+-------------------+-----+        +----+-----------------+----+-+
  |                      |                   |   Authorization      |
  |                      |                   |       Token          |
  |                      |                   |      Request         |
  |                      |                   |<---------------------|
  |                      |                   |   Authorization      |
  |                      |   End-to-End      |       Token          |
  |                      |   Communication   |      Response        |
  |                      |  (Authorization   |--------------------->|
  |                      |     Token)        |                      |
  |<----------------------------------------------------------------|
  |Create(PF=            |                   |                      |
  |(src=10.1.0.5,        |                   |                      |
  | dst=17.12.23.5,      |Create(PF=         |                      |
  | sport=1200,          |(src=139.23.203.23,|                      |
  | dport=600); Token)   | dst=17.12.23.5,   |Create(PF=            |
  |--------------------->| sport=5000,       |(src=10.1.0.5,        |
  |                      | dport=600); Token)| dst=17.12.23.5,      |
  |                      |------------------>| sport=1200,          |
  |                      |                   | dport=600)           |
  |                      |                   |--------------------->|
  |                      |                   |    Response(         |
  |                      |                   |    NAT-Object(...))  |
  |                      |  Response(        |<---------------------|
  |                      |  NAT-Object(...)) |                      |
  |    Response(         |<------------------|                      |
  |    NAT-Object(...))  |                   |                      |
  |<---------------------|                   |                      |
  |                      |  SIP Signaling    |                      |
  |<~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~>|
  |                      |                   |                      |
  |                   Data Traffic (unidirectional)                 |
  |================================================================>|
```

Figure 11: Sender-Initiated NAT/Firewall Traversal with Authorization Token

10.6 Sender-Initiated Firewall Signaling only at the Access Network

Sometimes people argue that the signaling message exchange should be done locally at the network access only because per-flow signaling messages are not processed in the core network. Instead of sending the signaling messages from one access network to the other whereby the signaling messages are transparent in the core each host transmits signaling messages independently in its own network. Although the concept sounds very simple at the first glance it turns out to be very complex in the generic case. Most difficulties appear because of the asymmetric routing architecture. Establishing policy rules in the uplink direction is fairly simple and requires only a mechanism which allows some sort of scoping (i.e. signaling messages have to terminate somewhere in the access network) without actually indicating the end-point. Casp provides means for scoping and local access network signaling. However the installation of policy rules on the downlink direction is complicated because some topology information inside the network must be known in order to avoid policy rule creation at the wrong devices. Hence there is a built-in risk to cause the protocol to fail (i.e. to install policy rules at the wrong location).

For the message flow described in Figure 12 we assume the following protocol behavior:

- Host A and Host B initiate a bi-directional packet filter establishment with a scope restricted to the local access network only. Without some sort of bi-directional signaling message exchange, a TRIGGER message is required to initiate a downlink Traffic Selection establishment.
- Based on the characteristics of local signaling message exchanges at both access networks, assumptions about the topology must be made (or some topology information must be known).
- In this simplified message flow no NAT device is present.
- Host A has a-priori knowledge about the packet filter for the inbound traffic (i.e. src=17.12.23.5 and sport=601).

```
+-------------------------------+        +------------------------------+
| +------+             +------+ |        | +------+        +---------+ |
| |Host A|  Network    | FW 1 | |        | | FW 2 | Network | Host B | |
| +--+---+     A       +--+---+ |        | +--+---+    B    +---+----+ |
+-+--+-----------------+-----+           +----+-----------------+----+-+
  |Create(PF=(src=139.23.205.5,                |                      |
  | dst=17.12.23.5, sport=5000, dport=600);    |                      |
  | Next=Create(PF=(src=17.12.23.5,            |                      |
  | dst=139.23.205.5,sport=601,dport=5001));   |                      |
  | Scope=NetworkA)      |                     |                      |
  |--------------------->|                     |                      |
  |Create(PF=            |                     |                      |
  |(src=17.12.23.5,      |                     |                      |
  | dst=139.23.205.5,    |                     |                      |
  | sport=601,           |                     |                      |
  | dport=5001))         |                     |                      |
  |<---------------------|                     |                      |
  |                      |   End-to-End        |                      |
  |                      |  Communication      |                      |
  |                      |  (PF) - Optional    |                      |
  |<--------------------------------------------------------------->|
  |                      |Create(PF=(src=17.12.23.5,                  |
  |                      | dst=139.23.205.5, sport=601, dport=5001);|
  |                      | Next=Create(PF=(src=139.23.205.5,          |
  |                      | dst=17.12.23.5, sport=5000, dport=600));  |
  |                      | Scope=NetworkB)     |                      |
  |                      |                     |<---------------------|
  |                      |                     |Create(PF=            |
  |                      |                     |(src=139.23.205.5,    |
  |                      |                     | dst=17.12.23.5,      |
  |                      |                     | sport=5000,          |
  |                      |                     | dport=600))          |
  |                      |                     |--------------------->|
  |                 Data Traffic (bi-directional)                     |
  |<================================================================>|
```

Figure 12: Sender-Initiated Firewall Signaling only at the Access Network

With the initial CREATE message Host A already supplies packet filter information for the bi-directional reservation (i.e. the CREATE message by Host A is followed by another CREATE message from FW 1). To kept the CREATE signaling message within the local access network scoping is used. Indicating a particular IP address might also be possible but often the endpoint is unknown to the end host. As a result of successful processing a CREATE message is returned in response with the already provided packet filter.

Optionally an end-to-end message communication might follow to transmit packet filter information from Host A to Host B. In most cases some communication is however required. Similar as in Network A a CREATE message is initiated by the end host with the Next object set to another CREATE message.

Finally if everything was successful data can be exchanged in both directions on port $5001 < -601$ and a $5000- > 600$.

10.7 Sender-Initiated NAT and Firewall Traversal within the Access Network

The message flow described in Figure 13 extends the description in Figure 12 by using a uni-directional signaling exchange. As a consequence of this extension a TRIGGER message is required to cause a downlink signaling message to be sent within Network B. In order to avoid this message Network B could intercept the end-to-end message exchange to trigger a signaling message to Host B. However this approach might suffer from the problem to be able to read and evaluate end-to-end signaling messages.

In addition, a NAT device is used in Network A which requires Host A to request a NAT binding and the corresponding NAT-Object which is then communicated to Host B. Using the packet filter information inside the NAT-Object Host B learns the public IP address and port information of the data traffic transmitted by Host A.

The access network signaling message exchange requires some topology information as explained in previous figures. The TRIGGER message must cause a downlink signaling message to be initiated by a network device which where the data traffic of Host A is sent through. This particular issue will be explained in more detail in a future version of the document.

A even more difficult example would address a topology where each network is equipped with a NAT. The same is true for packet filter installation for data traffic flowing in both directions with one or two NATs.

```
+------------------------------+        +-----------------------------+
| +------+            +------+ |        | +------+        +---------+ |
| |Host A|  Network   | NAT 1| |        | | FW 2 | Network| Host B  | |
| +--+---+     A      +--+---+ |        | +--+---+   B    +---+-----+ |
+-+--+-------------------+-----+        +----+----------------+-----+-+
  |Create(PF=            |                   |                      |
  |(src=192.168.1.5,     |                   |                      |
  | dst=17.12.23.5,      |                   |                      |
  | sport=5000,          |                   |                      |
  | dport=600);          |                   |                      |
  | Scope=NetworkA)      |                   |                      |
  |--------------------->|                   |                      |
  |Response(             |                   |                      |
  |NAT-Object=           |                   |                      |
  |(src=139.23.203.30,   |                   |                      |
  | dst=17.12.23.5,      |                   |                      |
  | sport=8000,          |                   |                      |
  | dport=600))          |  End-to-End       |                      |
  |<---------------------| Communication     |                      |
  |                      | (NAT-Object)      |                      |
  |<--------------------------------------------------------------->|
  |                      |                   |Trigger(PF=           |
  |                      |                   |(src=139.23.205.30,   |
  |                      |                   | dst=17.12.23.5,      |
  |                      |                   | sport=8000,          |
  |                      |                   | dport=600);          |
  |                      |                   | Scope=NetworkB)      |
  |                      |                   |<---------------------|
  |                      |                   |Create(PF=            |
  |                      |                   |(src=139.23.205.30,   |
  |                      |                   | dst=17.12.23.5,      |
  |                      |                   | sport=8000,          |
  |                      |                   | dport=600))          |
  |                      |                   |--------------------->|
  |                  Data Traffic (uni-directional)                 |
  |================================================================>|
```

Figure 13: Sender-Initiated NAT and Firewall Traversal within the Access Network

11 Security Considerations

Installing packet filters to one or more firewalls is a security sensitive process. Security protection of signaling messages is necessary in order to defeat a number of threats. This section gives a brief discussion of possible threats and addresses their corresponding countermeasures.

11.1 Threats

Denial of Service: Denial of service attacks can be launched by modifying messages used during the discovery process. A client could then be forced to contact a "wrong" firewall which is outside the data path. Furthermore it is possible to flood a firewall with bogus request and thereby cause massive state and computational resources to be allocated as part of the key exchange process. Furthermore an adversary can modify the packet filter of a request to cause a large number of packet filters to be allocated. An adversary might also remove administrator installed packet filters which are not related to previous packet filter installations by users.

Man-in-the-Middle: MITM attacks are possible during the discovery process where the entity of a firewall is discovered. In this case the user might be convinced to communicate with a firewall which is not the case. Many of these attacks are related to the discovery mechanism and therefore also described in [1]. Further threats which are not specific to the scout mechanism but also related to the next-hop discovery mechanism require further investigation (such as SLP, DHCP, DNS, etc.). The authors of some of these configuration mechanisms have already identified potential vulnerabilities and provide the corresponding security protection.

Eavesdropping: An eavesdropper might be able to learn some installed packet filters by listening to the signaling message communication between a client and a firewall. Furthermore it might be possible to learn an exchanged authorization tokens between the two entities or between entities along the path. Since the session identifier is used to uniquely identify state established along entities along the path an adversary might reuse this identifier to refer to existing state information.

Integrity Violation: By modifying a request message, an adversary can delete installed firewall filters, install filters using a different authorization identity or to create filters with a large lifetime.

Masquerading: An adversary might gain information by querying installed packet filters at a firewall by masquerading the identify of a real user. This might be used for subsequent attacks.

Rogue Firewall: An adversary at a compromised firewall might exploit an existing trust relationship to install or remove filters at other firewalls. Furthermore it is possible to return a NAT object with wrong information causing subsequent data traffic to be send to an arbitrary location.

Unauthorized Access: A regular user might install firewall filters although he is not allowed because of missing authorization. Administrators are usually very concerned about installing packet filters from users access from an external network.

Replay Attacks: An adversary might eavesdrop CASP-NATFW signaling messages and use them later for a replay attack. Furthermore an adversary might be able to collect authorization tokens and reuse them in a different context or later in time to open holes into a firewall.

Privacy Violation: Adversaries can learn about the NI and NR's identities participating in the message exchange by eavesdropping information exchanged between the two end-systems. Especially authorization tokens exchanged between end-systems outside the CASP protocol (as explained in Section 4.4) represent a vulnerability.

11.2 Countermeasures

To prevent the above-listed attacks a number of countermeasures are taken:

Denial of Service: To limit denial of service attacks a number of countermeasure were taken. First the scout protocol (and other configuration mechanisms) experience some protection to prevent basic attacks. Furthermore it is necessary to mutually authenticate and authorize both peers after establishing a transport layer connection as described in [1]. Since the authentication and key exchange protocol requires state and computational resources it has to be resistant against denial of service attacks. When transmitting CASP-NATFW specific information protection of the requests itself is necessary to prevent an adversary from object modification which otherwise would cause unpredictable behavior.

Man-in-the-Middle: MITM attacks during the discovery phase are prevented by secure configuration mechanisms. The scout protocol experiences limited security protection by its nature. However an authentication and authorization step is required after learning the identity of the next CASP peer. MITM adversaries will experience difficulties launching a successful attack after transport layer connection establishment because of the signaling message protection.

Eavesdropping: Eavesdropping of signaling messages is prevented by using either IPSec ESP (without NULL encryption) or by using TLS (with encryption cipher-suites). It is therefore not possible to learn authorization tokens, session identifiers or other firewall packet filter specific information that might be useful for an adversary eavesdropping on for example a wireless link. With the suggested security protection eavesdropping is therefore only possible at CASP-NATFW aware nodes participating in the signaling message exchange. This is, however, intentional and required for the operation of the protocol.

Integrity Violation: Modifying the content of signaling packets is prevented by either IPSec or TLS. Exchanged information thereby experiences both confidentiality as well as integrity protection. The usage of integrity protection with IPSec ESP is strongly recommended.

Masquerading: Spoofing an identity to be able to delete or query installed packet filter information is prevented by authentication of the originator (i.e. data origin authentication) of transmitted signaling messages. For the establishment of the required security associations mutual authentication is assumed.

Rogue CASP-NATFW Node: Firewalls are security sensitive network devices. An adversary can use a compromised firewall in a number of ways. To prevent a compromised firewall to harm other firewalls, trust might be limited and strong verification of request might be required. In case of missing peer-to-peer trust relationships more sophisticated protocol handling (as described in 4.3 and 4.4) is necessary. Such a handling makes it more difficult for an adversary to perform a successful attack. Note that any malicious CASP-NATFW (or CASP node in general) can impact the security of other entities (not just firewalls).

Unauthorized Access: Differentiation of access rights between various users and user-groups is common. The same type of authorization mechanisms based on access control lists can be applied. If authorization tokens are used then additionally a locally known user must be able to request such a token. For the trust relationship described in 4.3 one administrative domain must have a pre-established security association. The establishment of such this security association is usually bound to specific access control rights.

Privacy Violation: Encryption of information about user identities contained in authorization token prevents an adversary from obtaining user specific information. Currently only a keyed message digest function (HMAC) is provided to protect the authorization token content against modification. Either a custom mechanisms for encrypting some token parts or CMS encryption could be used to provide the necessary protection. Further investigation is required.

Linking authorization between different protocols, to strict policy rule creation by the end host, is possible with authorization tokens which contain information about the application, policy rule, authorization decision, lifetime, etc. An authorization token can be based on CMS or on a custom security mechanism such as defined in [10, 11].

To summarize: CASP uses security mechanisms described in [1]. Securing the messaging layer in a CASP-peer to CASP-peer fashion is provided either by IPsec or by TLS. In some cases security protection between neighboring peers is not sufficient. Non peer-to-peer protection of client layer objects is provided by CMS which allows CASP-NATFW objects defined in this document to be encapsulated and protected by CMS.

12 Conclusion

CASP-NATFW aims to provide a long-term solution to communicate with NATs and Firewalls with the following properties:

Routing of Signaling Messages: CASP with its scout discovery mechanisms allows signaling messages to follow the path of the data traffic towards a destination. This assumes that standard routing is used. CASP, however, operates independent of the underlying routing mechanism. Route changes can be detected by the scout protocol and signaling message transmission is adopted accordingly. Other mechanisms for detecting route changes can also be used such as routing protocols.

Security Protection: Creating holes into a firewall is a sensitive task that requires trust and an appropriate security protection of the signaling messages in order to be successful. Trust assumptions between the participating entities thereby determine whether the task of installing packet filters at a firewall is possible at all. CASP-NATFW thereby reuses the security mechanisms introduced by CASP. Still some additional security mechanisms described in this document have to be used to provide secure protocol operation.

Flexibility in Message Delivery: Signaling messages can be triggered by any node along the path. In most cases, however, it is the responsibility of the signaling message initiator (typically the end host) to provide the necessary information policy rules install. CASP messages might terminate at any CASP peer along the path. Hence it is not necessary to forward the messages to the final destination. The decision whether to furthermore forward the signaling message toward the destination can be caused by the initiator (by including CASP specific information) or the decision could also be forced for example by a non CASP-aware firewall. Such a device might not forward CASP message. An example is an authorization failure generated because of lacking trust (and proper credentials by the signaling initiator).

Error Resilience: CASP was designed based on the soft-state principle to allow orphan states to time-out automatically.

End Host Topology Unawareness: Routing signaling messages along the data path allows CASP aware nodes to reflect topology information into the processing of CASP signaling messages. Processing of Filters is an example where local topology and protocol information need to be available to ensure proper behavior. Filter handling is already defined in CASP [1]. Defining them at the CASP M-Layer is necessary since this object is used by more than one client layer protocol. The Filter used in CASP-QoS [12] messages might require modification by a NAT along the path. Mid-path modification of the packet filter allows the end host to be topology unaware. If topology information needs to be incorporated into the signaling message processing then it should be done at the locations where the corresponding information is easily available (for example at the individual CASP-NATFW aware nodes along the path).

13 Open Issues

- The format of the objects need more work.
- The structure of the authorization token needs more investigation. There is also a question about a custom token format or a CMS object. Both have advantages and disadvantages.
- Terminology needs to be aligned with the Midcom Requirements and Framework drafts. Issues (such as groups of policy rules) discussed in these documents have to be mapped against the issues in this draft.
- Packet filter attributes need some work to avoid the complex verification in case of overlapping rules. It must not be possible to prevent an administrator-created deny policy rule to become ineffective by an added allow policy rule with an overlapping port range. Hence it might be necessary to have an additional verification step to prevent these type of problems.
- The NAT-Object might not necessarily be required, the approach taken in [6] could be used. The policy rule creator uses a filter with an internal address/port pair, an optional inside address/port pair (called in this document a local destination address/port pair used for twice NAT) with no parameters, as well as the external address/port pair (remote entity that will receive the data flow). In case there is a NAT on the path, the NAT will provide an outside address/pair (translated address/port) if it was firewall the outside address/pair would be the external address/pair.

14 Acknowledgements

We would like to thank (in alphabetical order) Steffen Fries, Xiaoming, Fu, Joerg Ottensmayer and Martin Reinhardt for their comments to this draft.

A Object Format Details

For concreteness, we describe a strawman packet format below.

All CASP messages are composed of one or more TLV (type-length-value) objects. Within each object, elements are aligned on multiples of their size, to speed processing. All objects have lengths of a multiple of 32 bits. The length field in the object indicates the number of 32-bit words.

We describe messages and objects as pseudo-C structures. Elements are enumerated in transmission order. We use the data types uint8, uint16, uint32, uint64, uint128 to identify unsigned integers with 8, 16, 32, 64 or 128 bits, respectively.

Definitions for IPv4 and IPv6 address for the usage with Traffic Selectors are already provided in [1].

IPSec ESP and AH SPIs is four bytes in length.

```
typedef struct uint32 SPI;
```

Using a custom authorization token format might be more lightweight. (TBD: Authorization tokens can either be defined as CMS objects or as a objects with a custom structure. Using CMS object would simplify its definition and would allow a more generic usage. However CMS objects are larger in size than custom build tokens. Some investigation is required to find the optional usage.)

The following fields could be included in such a token:

```
typedef struct {
   uint32 ID;
   Identity token_creator, token_requestor, token_user;
   Identity src_addr, dst_addr;
   NTP_TIMESTAMP timestamp;
   uint8 AlgorithmID;
   uint8 HMAC[20];
   ...Object describing the authorized PF....
} AuthToken;
```

An authorization token is identified by a 32-bit number. The src_addr and the dst_addr attribute might contains an IPv4, IPv6 address or a FQDN. The Identity can either be a generic Unicode and ASCII ID, a FQDN or a URI. Unicode Identifiers (Unicode_ID), ASCII Identifiers and FQDNs are defined in [13]. The Uniform Resource Identifiers (URI) is defined in [14].

Since a NAT may change the source address it is possible to specify a FQDN, URI or an ASCII/Unicode ID or to omit the field. The token_creator specifies the identity of the entity which was responsible for the creation of the token. Information about this entity is necessary to route the token to the same entity for verification. Information about the entity requesting the token might be required. Finally the user identity obtained from authentication might be included. Especially if authentication to a firewall in the middle of the CASP-chain is required then this information provides additional authorization information.

For cryptographic protection of the authorization token a keyed message digest HMAC [15] is used whereby the used algorithm (MD5, SHA-1) is indicated in the AlgorithmID field. The secret key necessary for the HMAC computation needs to be locally known only since verification is done at the token creator. The format of the NTP timestamp is defined in [16]. Finally the object contains information about the authorized packet filter. Since a NAT might change some of this information its usefulness is questionable.

B Authors' Addresses

Henning Schulzrinne
Dept. of Computer Science
Columbia University
1214 Amsterdam Avenue
New York, NY 10027
USA
EMail: schulzrinne@cs.columbia.edu

Hannes Tschofenig
Siemens AG
Otto-Hahn-Ring 6
81739 Munich
Germany
EMail: Hannes.Tschofenig@siemens.com

Cedric Aoun
Nortel Networks
France
EMail: cedric.aoun@nortelnetworks.com

References

[1] H. Schulzrinne, H. Tschofenig, X. Fu, and A. McDonald, "Casp - cross-application signaling protocol," internet draft, Internet Engineering Task Force, 2003. Work in progress.

[2] P. Srisuresh, J. Kuthan, J. Rosenberg, A. Molitor, and A. Rayhan, "Middlebox communication architecture and framework," Internet Draft, Internet Engineering Task Force, Mar. 2002. Work in progress.

[3] M. Shore, "The TIST (topology-insensitive service traversal) protocol," Internet Draft, Internet Engineering Task Force, May 2002. Work in progress.

[4] M. Shore, "Towards a network-friendlier midcom," Internet Draft, Internet Engineering Task Force, June 2002. Work in progress.

[5] H. Tschofenig and D. Kroeselberg, "Security threats for nsis," internet draft, Internet Engineering Task Force, 2003. Work in progress.

[6] M. Stiemerling, J. Quittek, and T. Taylor, "Midcom protocol semantics," Internet Draft, Internet Engineering Task Force, 2002. Work in progress.

[7] P. Srisuresh and M. Holdrege, "IP network address translator (NAT) terminology and considerations," RFC 2663, Internet Engineering Task Force, Aug. 1999.

[8] L. Amini and H. Schulzrinne, "Observations from router-level internet traces," in *DIMACS Workshop on Internet and WWW Measurement, Mapping and Modeling, (Piscataway, New Jersey)*, Feb. 2002.

[9] J. Manner *et al.*, "Localized RSVP," Internet Draft, Internet Engineering Task Force, May 2002. Work in progress.

[10] L. Hamer, B. Gage, and H. Shieh, "Framework for session set-up with media authorization," Internet Draft, Internet Engineering Task Force, July 2002. Work in progress.

[11] L. Hamer, B. Gage, M. Broda, B. Kosinski, and H. Shieh, “Session authorization for RSVP,” Internet Draft, Internet Engineering Task Force, July 2002. Work in progress.

[12] H. Schulzrinne, H. Tschofenig, X. Fu, and J. Eisl, “A quality-of-service resource allocation client for casp,” internet draft, Internet Engineering Task Force, 2003. Work in progress.

[13] S. Yadav, R. Yavatkar, R. Pabbati, P. Ford, T. Moore, S. Herzog, and R. Hess, “Identity representation for RSVP,” RFC 3182, Internet Engineering Task Force, Oct. 2001.

[14] T. Berners-Lee, R. Fielding, and L. Masinter, “Uniform resource identifiers (URI): generic syntax,” RFC 2396, Internet Engineering Task Force, Aug. 1998.

[15] H. Krawczyk, M. Bellare, and R. Canetti, “HMAC: keyed-hashing for message authentication,” RFC 2104, Internet Engineering Task Force, Feb. 1997.

[16] D. L. Mills, “Network time protocol (version 3) specification, implementation,” RFC 1305, Internet Engineering Task Force, Mar. 1992.