Network Working Group
INTERNET-DRAFT
Expires as of August 21, 2001

P. Srisuresh
Jasmine Networks
J. Kuthan
GMD Fokus
J. Rosenberg
Dynamicsoft
February, 2001

Middlebox Communication Architecture and framework
<draft-ietf-midcom-framework-00.txt>

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet- Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

Abstract

There are a variety of intermediate devices in the Internet today that require application intelligence for their operation. Many of the applications in use are complex and the datagrams pertaining to these applications cannot be identified by merely examining packet headers. Firewalls and Network Address Translators are typical examples of devices requiring application knowledge. Real-time streaming Voice-over-IP applications such as SIP and H.323 and peer-to-peer applications such as Napster are examples of complex applications. The document specifies an architecture and framework in which trusted third parties can be delegated to assist the intermediate devices with application

level intelligence to perform their operation. Doing this will allow the intermediate devices to continue to provide the services, while keeping the devices application agnostic.

1. Introduction

There are a variety of intermediate devices in the Internet today that require application intelligence for their operation. Many of these devices enforce application specific policy based functions such as packet filtering, differentiated Quality of Service, tunneling, Intrusion detection, security and so forth. Network Address Translators, on the other hand, provide routing transparency across address realms (within IPv4 routing network or across V4 and V6 routing realms). Application Level gateways (ALGs) are used in conjunction with NAT function to provide end-to-end transparency for applications. There may be other types of devices requiring application intelligence for their operation. A middlebox is an intermediate device requiring application intelligence to implement one or more of the functions described. The discussion scope of this document is however limited to middleboxes implementing Firewall and NAT functions only.

Tight coupling of application intelligence with intermediate devices makes maintenance of intermediate devices hard with the advent of new applications. Built-in application awareness typically requires updates of operating systems with new applications or newer versions of existing applications. Operators requiring support for newer applications will not be able to use third party software/hardware specific to the application and are at the mercy of their intermediate device vendor to make the necessary upgrade. Further, embedding intelligence for a large number of application protocols within the same device increases complexity of the device and the device is likely to be error prone and degrade in performance.

This document describes a framework in which middlebox application intelligence can be moved from intermediate devices into external MIDCOM agents. These MIDCOM agents assist middlebox devices through a generic application-independent middlebox communication (MIDCOM) protocol (yet to be devised). The communication between a MIDCOM agent and a middlebox will be transparent to the end hosts that part take in the application, unless one of the end hosts assumes the role of an external agent. Discovery of middleboxes in the path of an application instance is out of the scope of this document.

This document describes the framework in which middlebox communication takes place and the various elements that constitute the framework. Section 2 describes the terms used in the document. Section 3 defines the architectural framework of a middlebox for communication with MIDCOM agents. The remaining sections cover the components of the framework and operational considerations. Section 4 illustrates the various types of MIDCOM agents. Section 5 considers the role of Policy server and its function with regard to MIDCOM agent authorization. Section 6 addresses operational considerations in deploying a protocol adhering to the framework described here. Section 7 is an applicability statement identifying the location of middleboxes, which are the targets of the MIDCOM protocol. Section 8 identifies security considerations for the middlebox in view of the MIDCOM interface.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALLNOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119. Below are the definitions for the terms used throughout the document.

2.1. MiddleBox

Middlebox is a network intermediate device that requires application specific intelligence for its operation. Intermediate Devices implementing policy based packet filtering, intrusion detection, load balancing, tunneling, IPsec security and Network Address Translation (NAT) functions are all examples of a middlebox device. A middlebox may implement one or more of these functions.

2.2. Firewall

Firewall is a policy based packet filtering Middlebox, typically used for restricting access to/from specific devices and applications. The policies are often termed Access Control Lists (ACLs).

Firewall can perform its function as a stateless device to identify single session applications based on IANA defined well-known ports such as telnet, DNS, HTTP and rlogin, by merely examining the IP (IPv4 or IPv6) and transport headers of the datagrams traversing the device. However, bundled session applications such as FTP, H.323, SIP and RTSP, use a control connection to exchange address and port parameters to establish data sessions and session orientations.

Without application specific knowledge of the payload, firewall cannot know the inter-dependency of the bundled sessions and would treat each session as unrelated to one another and may not permit sessions that might need to be permitted.

2.3. NAT

Network Address Translation is a method by which IP addresses are mapped from one address realm to another, providing transparent routing to end hosts. This is achieved by modifying end node addresses en-route and maintaining state for these updates so that datagrams pertaining to a session are forwarded to the right end-node in either realm. Refer [NAT-TERM] for the various types of NAT devices in use.

NAT device alone cannot provide the necessary application/protocol transparency in all cases and seeks the assistance of Application Level Gateways (ALGs) where possible, to provide application level transparency. [NAT-COMP] identifies the protocols and applications that break with NAT enroute.

The term NAT in this document is very similar to the IPv4 NAT described in [NAT-TERM], but is extended beyond IPv4 networks to include the IPv4-v6 NAT-PT described in [NAT-PT]. While the IPv4 NAT [NAT-TERM] translates one IPv4 address into another IPv4 address to provide routing between private V4 and external V4 address realms, IPv4-v6 NAT-PT [NAT-PT] translates an IPv4 address into an IPv6 address and vice versa to provide routing between a V6 address realm and an external V4 address realm.

Unless specified otherwise, NAT in this document is a middle box referring to both IPv4 NAT as well as IPv4-v6 NAT-PT devices.

2.4. Proxy

A proxy is an intermediate relay agent between clients and servers of an application, relaying application messages between the two. Proxies use a special protocol to communicate with proxy clients and relay client data to servers and vice versa. A Proxy terminates sessions with both the client and the server, acting as server to the end-host client and as client to the end-host server.

Applications such as FTP, H.323, SIP and RTSP use a control connection to establish data sessions. These control and data sessions can take divergent paths. While a proxy can intercept both the control and data connections, it is possible for it to intercept only the control connection. This is often the case with real-time streaming applications such as H.323, SIP and RTSP.

2.5. ALG

Application Level Gateways (ALGs) are agents that possess the application specific intelligence and knowledge of an associated middlebox function. An ALG examines application traffic in transit and assists middlebox in carrying out its function.

An ALG may be co-resident with a middlebox or reside externally, communicating through a middlebox communication protocol. It interacts with a middlebox to set up state, access control filters, use middlebox state information, modify application specific payload or perform whatever else is necessary to enable the application to run through the middlebox.

ALGs are different from proxies, in that, ALGs are transparent to end hosts, unlike the proxies. Proxies terminate sessions with both end-hosts. ALGs, on the other hand, do not terminate session with either end-host. Instead, proxies examine and optionally modify application payload content to facilitate the flow of application traffic through a middlebox. The objective of an ALG is to assist middleboxes in carrying out their function. Whereas, the objective of a proxy is to act as a relay agent between application clients and servers.

ALGs are similar to Proxies, in that, both ALGs and proxies facilitate Application specific communication between clients and servers.

2.6 End Hosts

End hosts are entities that are party to a networked application instance. End hosts referred in this document are specifically those terminating Real-time streaming Voice-over-IP applications such as SIP and H.323 and peer-to-peer applications such as Napster. End-hosts referred in the document are also assumed to be traversing a middlebox device.

2.7. MIDCOM Agents

MIDCOM agents are entities performing ALG function external to a middlebox. MIDCOM agents may be located either In-Path or Out-of-path of an application instance.

In-Path MIDCOM agents are those that are naturally within the message path of the application they are associated with. This may be an application proxy or in the extreme case, one of the end-nodes, that is party to the application. Out-of-Path MIDCOM

agents are entities that are not necessarily in the path of application message flows.

2.8. Policy Server

Policy Server is a management entity that interfaces with a middlebox to enforce policies regarding authorization of MIDCOM agents to access and influence middlebox operation. A MIDCOM agent may be pre-configured on the middlebox as a trusted entity, with or without security credentials. In the case where a MIDCOM agent is not pre-configured, the middlebox will consult Policy Server Out-of-band for validating the authorization to accept requests from the agent. A policy server might add or delete MIDCOM agents on a middlebox.

The protocol facilitating the communication between a middlebox and Policy Server need not be MIDCOM protocol

2.9. Middlebox Communication (MIDCOM) protocol

The protocol used by MIDCOM agents to interface with the middlebox and influence its operation. This is a protocol yet to be devised.

3.0 Architectural framework for Middlebox devices

A middlebox may implement one or more of the middlebox functions selectively on multiple interfaces of the device. There can be a variety of MIDCOM agents interfacing with the middlebox to communicate with one or more of the middlebox functions on an interface. As such, the communication protocol MUST allow for selective communication between a specific MIDCOM agent and a middlebox function on the interface. The following diagram identifies a possible layering of the functions in a middlebox and a list of MIDCOM agents that might interact with it.

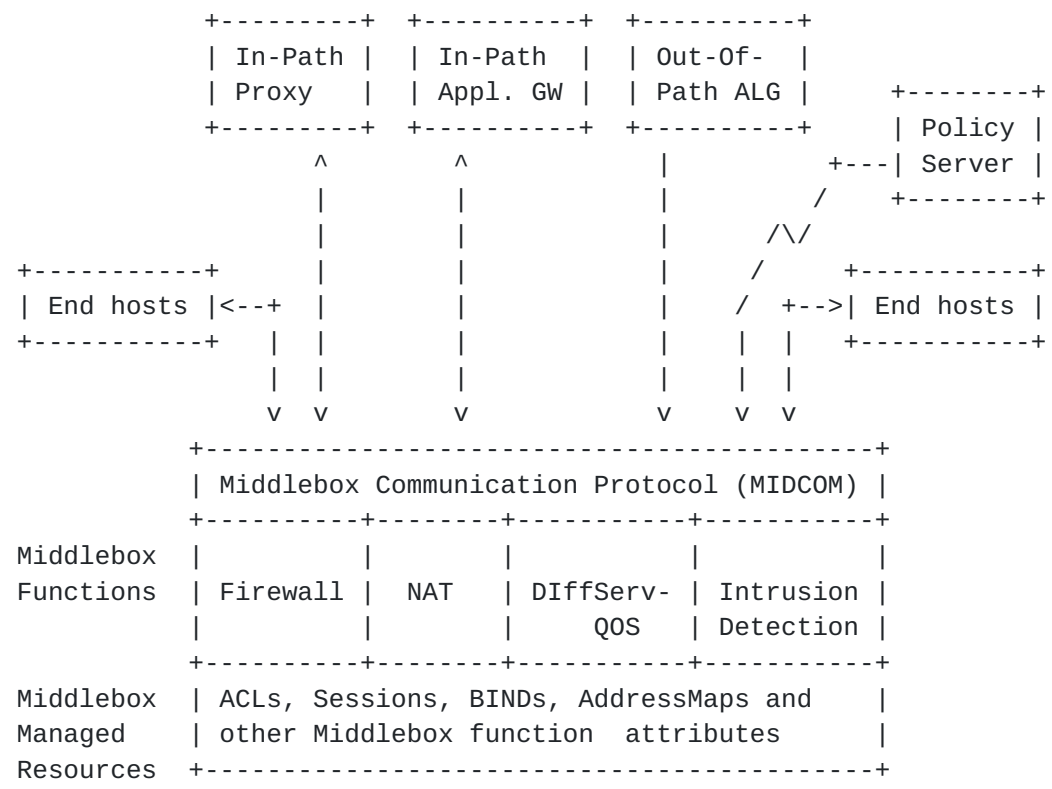

figure 1. MIDCOM agents interfacing with Middlebox devices

Resources such as a Session-Descriptor may be shared across middlebox functions. A Session-Descriptor may uniquely identify a session denoted by the tuple of (SessionDirection, SourceAddress, DestinationAddress, Protocol, SourcePort, DestinationPort). An aggregated Session-Descriptor, on the other hand, may have one of the tuple elements denoted by a regular expression (ex: Any source port). The attributes associated with a Session-Descriptor may be specific to the individual middlebox function. As Session-Descriptors are shared across middlebox functions, a Session-Descriptor may be created by a function, and terminated by a different function. For example, a session-descriptor may be created by the firewall function, but terminated by the NAT function, when a session timer expires.

A middlebox may also have function specific resources such as Address maps and Address binds to enforce NAT function and Access Control Lists (ACLs) to enforce firewall function. Policies governing a function (such as NAT or firewall) may be based on an application which may not lend itself easy to recognize. Application specific agents (co-resident on the same device or external to the device) would be able to examine the IP datagrams and help identify the application the datagram belongs to. The

ALG may also be able to assist the middlebox function in performing functions unique to the application and the middlebox function. For example, an ALG assisting NAT might perform payload translations, in addition to identifying the sessions specific to the application.

A few of the middle-box devices deployed are stateless. There are many that are stateful and maintain per-connection state in the system. As such, the MIDCOM protocol must not assume or require that the middlebox to be one way or another and must work with both stateful and stateless devices.

4.0. MIDCOM Agents

MIDCOM agents are nodes external to a middlebox, possessing a combination of application specific intelligence and knowledge of middlebox function so as to assist middleboxes to perform their function. A MIDCOM agent may communicate with one or more middleboxes. Discovery of the middleboxes a MIDCOM agent communicates with is outside the scope of this document. The focus of the document is the framework in which a MIDCOM agent communicates with a middlebox using MIDCOM protocol, which is yet to be devised.

We will examine two types of MIDCOM agents in the following sub-sections.

4.1. In-path MIDCOM agents

In-Path MIDCOM agents are entities that have a native role in the path of the application traversal (with prior knowledge to one of the application end-hosts), independent of their MIDCOM function. Bundled session applications such as H.323, SIP and RTSP which have separate control and data sessions may have their sessions take divergent paths. In those scenarios, In-Path MIDCOM agents are those that find themselves in the control path. In majority of cases, a middlebox will likely require the assistance of a single agent for an application in the control path alone. However, it is possible that a middlebox function might require the intervention of more than one MIDCOM agent for the same application, one for each sub-session of the application.

Application Proxies are a good choice for In-Path MIDCOM agents as proxies, by definition, are in the path of an application between a client and server. Proxies can be interjecting both the control and data connections. For example, FTP control and

Data sessions are interjected by an FTP proxy server. However, proxies may also be interjecting just the control connection and not the data connections, as is the case with real-time streaming applications such as H.323, SIP and RTSP. Note, applications may not always traverse a proxy and some applications may not have proxy servers available.

4.1.1. In-Path MIDCOM agent illustration

SIP proxies and H.323 gatekeepers may be used as MIDCOM agents to control middleboxes implementing firewall and NAT functions. The advantage of using these as opposed to inventing a brand new agent is that the in-path boxes already possess the application intelligence. You will need to merely enable them to communicate using MIDCOM protocol to be effective MIDCOM agents. Figure 2 below illustrates a scenario where the in-path MIDCOM agents interface with the middlebox. Let us say, the policy Server has pre-configured the in-path proxies as trusted MIDCOM agents on the middlebox and the packet filter implements 'default-deny' packet filtering policy. Proxies use their application-awareness knowledge to control the firewall function and selectively permit a certain number of voice stream sessions dynamically using middlebox communication protocol. In addition, the in-path agents may also enforce application-specific choices locally, such as dropping messages infected with known viruses, or lacking user authentication.

In the illustration below in figure 2, the MIDCOM agents and the middlebox policy server are shown inside the enterprise boundary. The intent however is not to imply that these entities should reside within the enterprise boundary alone. There are no topological restriction to where these entities can be present, so long as the level of trust relationship with middlebox remains the same.

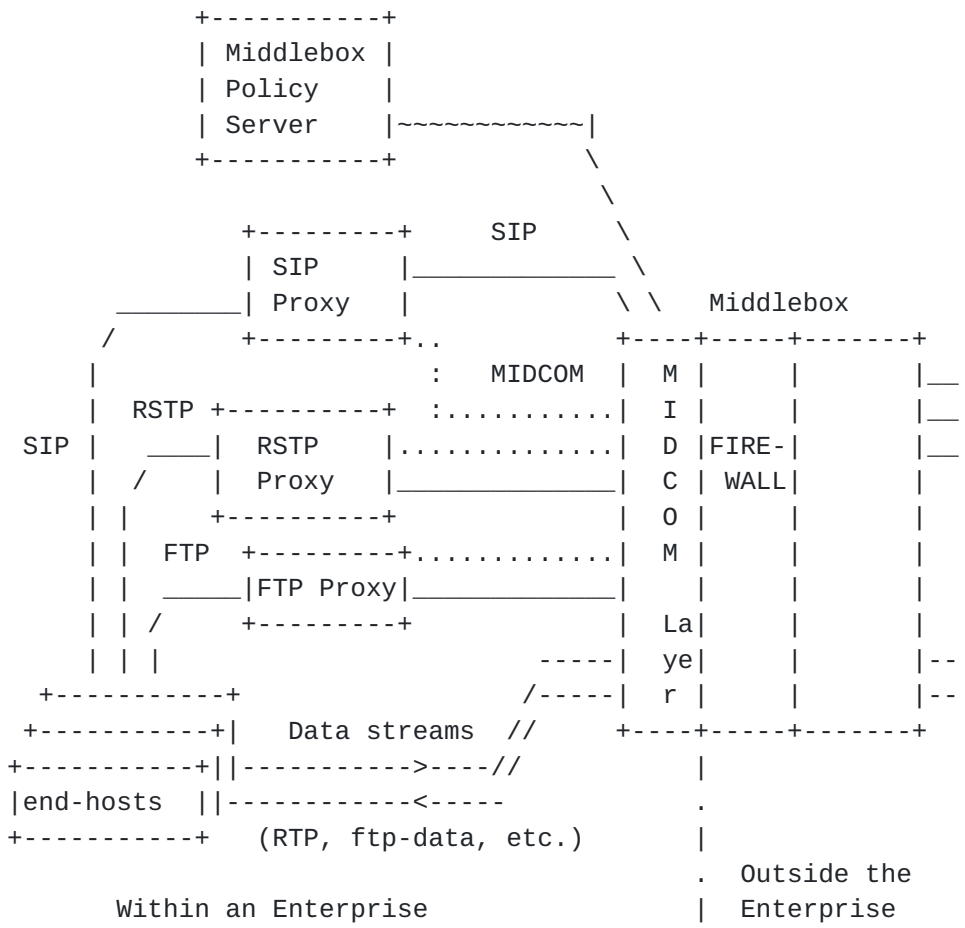

Figure 2: In-Path MIDCOM Agents for Middlebox Communication

4.1.2. Endhosts as In-Path MIDCOM agents

End-hosts are another kind of In-Path MIDCOM agents. Unlike Proxies, End-hosts are direct party to the application and will possess all the end-to-end application intelligence there is to it. End-hosts terminate both the control and data paths of an application. Unlike other MIDCOM agents, end-host is

also able to process secure datagrams. However, the problem would be one of scalability - upgrading all the end-hosts running a specific application.

4.2. Out-of-Path MIDCOM agents

Out-of-Path MIDCOM agents are entities that are not necessarily within the path of application traversal. Out-Of-Path Agents have a role in the application traversal, only by virtue of their MIDCOM function - No native role otherwise. It would be safe to assume that Out-of-Path MIDCOM agents are not in the path of application traversal. Out-of-Path agents have a few benefits. Out-of-Path agents can be implemented in a system, independent of any pre-existing application-aware entity. Unlike In-path agents, there are no topological restrictions to where the agents can be located. Further, multiple application specific agents can be grouped together on the same device.

However, Middleboxes seeking the assistance of Out-of-Path MIDCOM agents MUST explicitly redirect application specific datagrams to the agents. In the case of Bundled session applications, if agent assistance is needed for the control path alone, the middlebox will need to redirect only the control path packets to the agent.

5.0. Policy Server functions

A policy Server interfaces with a middle-box to configure and enforce MIDCOM agent authorization. For example, a policy server has the ability to add or delete MIDCOM agents on a middlebox and to notify a middlebox about the status and security requirements to allow accessibility to MIDCOM agents.

The primary objective of a policy server is to ensure that the security and integrity of a middlebox is not jeopardized. Specifically, the policy server should associate a trust level with each node attempting to connect to a middlebox and provide the security guidelines. Some hosts are less secure than others. The policy server must determine trust guidelines for controlling middleboxes according to the level of (presumed) security of the host making the request. The policy server must also determine guidelines to allow end hosts to communicate with the middle-box.

5.1. Authentication, Integrity and Confidentiality

Host authenticity and individual message authentication are two distinct types of authentications to consider. Host authentication refers to credentials required of a MIDCOM agent

to authenticate itself to the middlebox and optionally that of the middlebox to authenticate itself to MIDCOM agents. When authentication fails, the middlebox MUST not process signaling requests received from the agent that failed authentication.

To protect MIDCOM messages from being tampered with, Individual message authentication may be used [IPsec-AH] in addition to host authentication. Further, message confidentiality may also be administered by employing IPsec ESP protocol [IPsec-ESP] for the MIDCOM messages to/from a middlebox. Alternately, TLS based security may be employed instead of IPSec security. Simple Source-address based security is the least form of security and should be permitted only to the most trusted hosts.

Clearly, the middlebox must be able to perform host level authentication, and be able to authenticate individual messages (using IPsec or TLS based security).

5.2. Registration and deregistration with a middlebox

Prior to administering a middlebox, a registration process MUST take place as part of the MIDCOM protocol. The MIDCOM agent SHOULD initiate the registration process and it is up to the middlebox to accept or reject.

MIDCOM agents, their trust level and accessibility may be pre-registered with the middlebox while provisioning the middlebox function. Either the agent or the middlebox can choose to initiate a connection prior to any data traffic. Alternately, either party (middlebox or the MIDCOM agent) may choose to initiate a connection only upon noticing the application specific traffic.

Clearly, middlebox communication adds a new dimension to the way middlebox functions are used to being provisioned. In order for the middlebox to initiate connection to MIDCOM agents, middlebox function provisioning must be altered to reflect the optional ALG presence. For example, a revised ACL tuple for a firewall may be represented as follows.

(<Session-Direction>, <Source-Address>, <Destination-Address>, <IP-Protocol>, <Source-Port>, <Destination-Port>, <ALG>)

Agent accessibility information must also be provisioned. For a MIDCOM agent, accessibility information includes the IP address, trust level, host authentication profile and message authentication profile.

Techniques described above are necessary for the pre-registration of MIDCOM agents with the middlebox. However, it is possible to retain the provisioning on middlebox unchanged, by requiring MIDCOM agents to initiate the connection to middlebox. When Middlebox notices an incoming midcom connection, the middlebox will consult its Policy Server for authenticity, authorization and trust guidelines for the connection.

At the end of the MIDCOM session, it SHOULD be possible for either the middlebox or the MIDCOM agent to deregister themselves. Deregistration indicates the termination of MIDCOM session and may be prompted by a successful termination or failure of some sort.

6.0. Operational considerations

6.1. Multiple MIDCOM connections between agents and middlebox

A middlebox cannot be assumed to be a simple device implementing just one middlebox function and no more than a couple of interfaces. Middleboxes often combine multiple intermediate functions into the same device and have the ability to provision individual interfaces of the same device with different sets of functions and varied provisioning for the same function across the interfaces.

As such, a MIDCOM agent ought to be able to have a single MIDCOM connection with a middlebox and use the MIDCOM layer on the middlebox to demultiplex to different middlebox functions on the same middlebox interface. Likewise, a middlebox ought to be able to connect to multiple MIDCOM agents, as dictated by the policy server.

6.2. Asynchronous notification to MIDCOM agents

Asynchronous notification by the middlebox to a MIDCOM agent can be useful for events such as Session creation, Session termination, MIDCOM protocol failure, Middlebox function failure or any other significant event. Independently, ICMP error codes can also be useful to notify transport layer failures to the agents.

In addition, periodic notification of statistics update would also be a useful function that would be beneficial to certain types of agents.

6.3. Packet redirection

Middleboxes must have the ability to redirect packets to MIDCOM agents not in-path. The agent in turn would perform the necessary processing (specific to the application and middlebox function) and forward packet to the actual destination in the first-in-first-out (FIFO) order. Packet forwarding by the agent might necessitate the packet to traverse the middlebox for the second time. The middlebox should simply forward the packet the second time around without redirecting to the agent once again. Failing this, the packet would simply be recycling between the two entities. In order to avoid packet recycling between the agent and the middlebox, one might consider adapting tunneling approach for packet redirection between the agent and the middlebox.

6.4. Middlebox variations

As stated earlier, a middlebox could be implementing a variety of functions (ex; NAT and firewall) in the same box. Single or multiple agents may be deployed to offer application intelligence to the middlebox functions. However, the functions are assumed to be independent and the sequence in which these function operations may be performed on datagrams is not within the scope of this document.

6.5. Signaling and Data traffic

A large class of applications we are trying to solve with MIDCOM protocol is focused around applications that have a combination of one or more signaling and data traffic. The signaling may be done out-of-band using a dedicated stand-alone session or may be done in-band with data session. Alternately, signaling may also be done as a combination of both stand-alone and in-band sessions.

SIP is an example of an application based on distinct signaling and data sessions. SIP signaling session is used for call setup between a caller and a callee. MIDCOM agent may be required to examine/modify SIP payload content to administer the middlebox so as to let the media streams (RTP/RTSP based) through. MIDCOM agent is not required to intervene in the data traffic.

Signaling and context specific Header information is sent in-band within the same data stream for applications such as HTTP embedded applications, sun-RPC (embedding a variety of NFS apps), Oracle transactions (embedding oracle SQL+, MS ODBC, Peoplesoft) etc.

H.323 is an example of application that sends signaling in both

dedicated stand-alone session as well as in conjunction with data. Q.931 traffic traverses middleboxes by virtue of static policy, no MIDCOM control needed. Q.931 also negotiates ports for an H.245 TCP stream. A MIDCOM agent is required to examine/modify the contents of the H.245 so that H.245 can traverse it.

H.245 traverses the middlebox and also carries Open Logical Channel information for media data. So the MIDCOM agent is once again required to examine/modify the payload content needs to let the media traffic flow.

MIDCOM protocol is capable of supporting applications with independent signaling and data sessions as well as applications that have signaling and data communicated over the same session.

In the cases where signaling is done on a single stand-alone session, it is desirable to have a MIDCOM agent interpret the signaling stream and program the middlebox (that transits the data stream) so as to let the data traffic through uninterrupted.

7. Applicability Statement

Middleboxes may be stationed in a number of topologies. However, the signaling framework outlined in this document may be limited to only those middleboxes that are located in a DMZ (De-Militarized Zone) at the edge of an enterprise, connecting to the Internet. Specifically, the assumption is that you have a single middlebox (running NAT or firewall) along the application route. Discovery of middlebox along application route is outside the scope of this document.
It is conceivable to have middlebox devices located between departments within the same enterprise or inside service provider's domain and so forth. However, care must be taken to review each individual scenario and determine the applicability on a case-by-case basis.

The applicability may also be illustrated as follows. Real-time and streaming applications such as Voice-Over-IP and peer-to-peer applications such as Napster require administering firewall and NAT devices to let their media streams reach hosts inside a private domain. The requirements are in the form of establishing a "pin-hole" to permit a TCP/UDP session (the port parameters of which are dynamically determined) through a firewall or retain an address/port bind in the NAT device to permit connections to a port. These requirements are met by current generation firewall and NAT devices using adhoc methods, such as combining application intelligence with these NAT devices to identify the dynamic session

parameters and administer the NAT and firewall devices internally as appropriate. The objective of the MIDCOM protocol is to create a unified, standard way to exercise this functionality, currently existing in an ad-hoc fashion in some of the middlebox devices.

By adapting MIDCOM protocol, the NAT and firewall devices will be able to support newer applications they have not been able to support thus far. MIDCOM protocol does not and MUST not, in anyway, change the fundamental characteristic of NAT and firewall functions in the middlebox.

Typically, organizations shield a majority of their corporate resources (such as hosts) from visibility to the external network by the use of a De-Militarized Zone (DMZ) at the enterprise edge. Only a portion of these hosts are allowed to be accessed by the external world. The remaining hosts and their names are unique to the enterprise. Hosts visible to the external world and the authoritative name server that maps their names to network addresses are often configured within a DMZ (De-Militarized Zone) in front of a firewall. Hosts and middleboxes within DMZ are referred to as DMZ nodes.

Figure 3 below illustrates configuration of an enterprise with a DMZ at its edge. Actual configurations may vary. Internal hosts are accessed only by users inside the enterprise. Middleboxes, located in the DMZ may be accessed by agents inside or outside the enterprise.

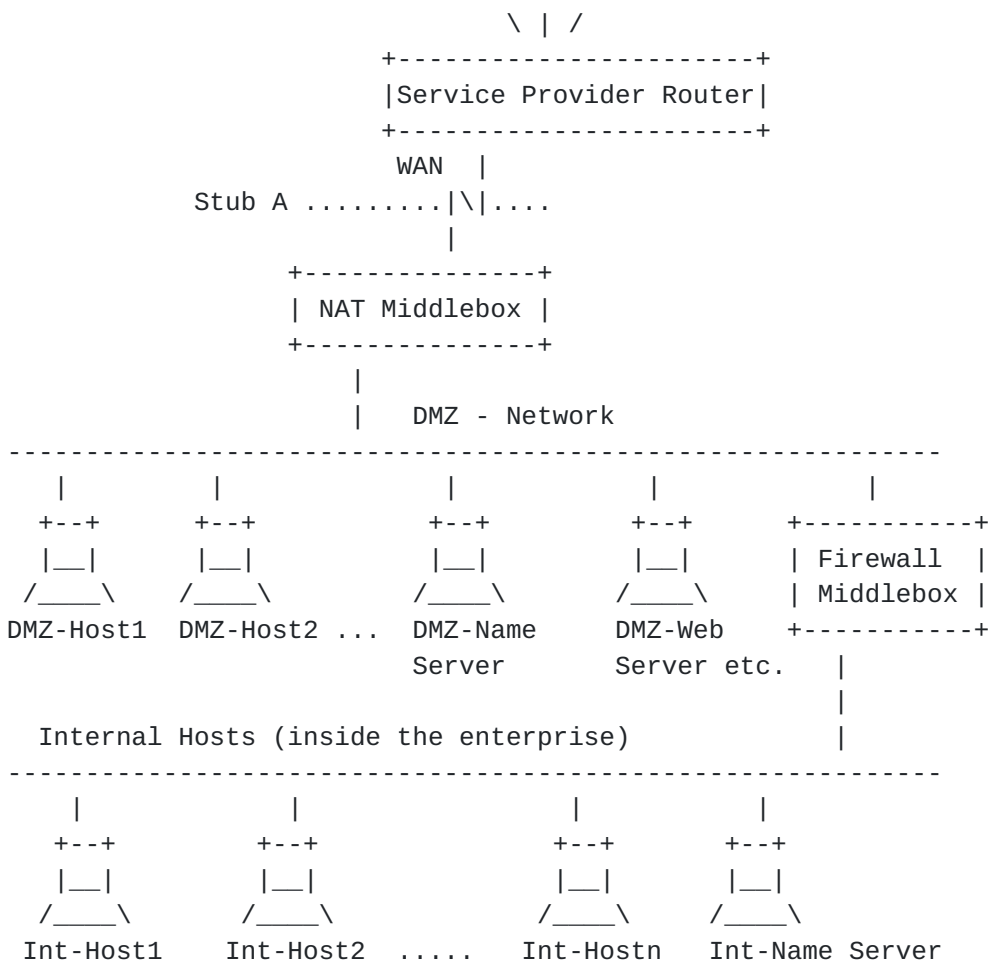

Figure 3: DMZ network configuration of an enterprise.

8. Acknowledgements

The authors wish to express their thanks and gratitude to the following for their valuable critique, advice and input on an earlier rough version of this document. Abdallah Rayhan, Andrew Molitor, Christian Huitema, Joon Maeng, Jon Peterson, Mike Fisk, Matt Holdrege, Paul Sijben, Philip Mart, Scott Brim and Richard Swale.

9. Security Considerations

MIDCOM protocol framework has significant security considerations to the operation of the middlebox. Section 5 is devoted to addressing the security vulnerabilities of the middlebox from MIDCOM agents. There is also a security vulnerability due to shared resources between the middlebox functions co-resident on the same device. It is possible that a middlebox function may be

abruptly disrupted due to malicious manipulation of the resource by an agent purporting to offer ALG service for a different middlebox function. Careful consideration must be given in the protocol design to ensure that agents for one function do not abruptly step over the resources impacting a different function.

REFERENCES

[IETF-STD] Bradner, S., " The Internet Standards Process -- Revision 3", RFC 1602, IETF, October 1996.

[SIP] Handley, M., H. Schulzrinne, E. Schooler, and J. Rosenberg, "SIP: Session Initiation Protocol", RFC 2543, IETF, March 1999.

[H.323] ITU-T Recommendation H.323. "Packet-based Multimedia Communications Systems," 1998.

[RTSP] Schulzrinne, H., A. Rao, R. Lanphier: "Real Time Streaming Protocol", RFC 2326, IETF, April 1998.

[FTP] J. Postel, J. Reynolds, "FILE TRANSFER PROTOCOL (FTP)", RFC 959

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

[NAT-COMP] Holdrege, M. and Srisuresh, P., "Protocol Complications with the IP Network Address Translator", RFC 3027, January 2001.

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

[APPL-ID] Bernet, Y. and Pabbati, R., "Application and Sub Application Identity Policy Element for Use with RSVP", RFC 2872, June 2000.

[RFC 1918] Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot, G. and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, February 1996.

[RFC 1700] J. Reynolds and J. Postel, "Assigned Numbers", RFC 1700

[IPsec-AH] Kent, S., and R. Atkinson, "IP Authentication Header", RFC 2402, November 1998.

[IPsec-ESP] Kent, S., and R. Atkinson, "IP Encapsulating Security Payload (ESP)", RFC 2406, November 1998.

Authors' Addresses

Pyda Srisuresh
Jasmine Networks
3061 Zanker Road, Suite B
San Jose, CA 95134
U.S.A.
Voice: (408) 895-5032
EMail: srisuresh@yahoo.com

Jiri Kuthan
GMD Fokus
Kaiserin-Augusta-Allee 31
D-10589 Berlin, Germany
E-mail: kuthan@fokus.gmd.de

Jonathan Rosenberg
dynamicsoft
200 Executive Drive
Suite 120
West Orange, NJ 07052
U.S.A.
email: jdrosen@dynamicsoft.com