STEM: Secure Telephony Enabled Middlebox[+]

Brennen Reynolds[1] and Dipak Ghosal[2]
1 - Department of Electrical and Computer Engineering
2 - Department of Computer Science
University of California, Davis, CA 95616, USA

Abstract

Dynamic applications, including IP telephony, have not seen wide acceptance within enterprises because of problems caused by the existing network infrastructure. Static elements, including firewalls and network address translation devices, are not capable of allowing dynamic applications to operate properly. The Secure Telephony Enabled Middlebox (STEM) architecture is an enhancement upon the existing network design to remove the issues surrounding static devices. The architecture incorporates an improved firewall that can interpret and utilize information in the application layer of packets to ensure proper functionality. In addition to allowing dynamic applications to function normally, the STEM architecture also incorporates several detection and response mechanisms for well known network-based vulnerabilities. This paper describes the key components of the architecture with respect to the SIP protocol.

Index Terms

IP Telephony, Firewall, Network Address Translator (NAT), Denial of Service (DoS), Network Security

I. Introduction & Background

The services operating over data networks have evolved from short text messages to real-time audio and video. This evolution has required the elements in the data path to become more sophisticated and intelligent. Today, enterprise networks are connected together through a large number of static devices including routers, firewalls and network address translation devices (middleboxes). These devices only operate with traffic that can be expressed as a static set of rules. They are capable of handling the applications currently deployed within the enterprise, but that is changing.

With the deployment of new applications including Internet Protocol (IP) telephony and video conferencing, problems have begun to appear. The problems stem from the dynamic nature of the underlying protocols upon which these applications are built. The application sessions for both major IP telephony protocols, Session Initiation Protocol (SIP) [1] and H.323 [2], can not be captured using a traditional filter. This dynamic behavior makes it impossible to create a set of predefined static filters to match each session. The problems related to running dynamic applications through static devices fill a range of categories including application vulnerabilities, denial of service, and scalability issues.

This problem has been identified and several solutions have been proposed. One solution outlined in [3] examines the possibility of adding a new device to the network that would work in parallel with a firewall or network address translation (NAT) device. All traffic related to IP telephony is routed through the new device. A new protocol, Firewall Control Protocol [3], was designed to allow the IP telephony proxy device to directly interact with the firewall. The argument for implementing a separate device to handle one particular protocol is that it reduces the overall complexity of the firewall and frees implementers from single vendor dependence when new applications are developed. This solution works well if the number of applications requiring a proxy is small but decreases the overall effectiveness and security gained by the firewall when there are multiple devices accessible from public networks.

Another solution presented in [4] involves a redesign of the middlebox architecture. They created adaptation layers between the internal components of a firewall: operating system, firewall and protocol parser. These layers allow each component to be independent of the others. Furthermore, there is the ability to interact with external components, but there are no provisions regarding the method of communication with these devices. The ability to externally configure the firewall device in real-time is a major advantage over current firewall architectures, but there are some very serious security risks. By including the capability to control the firewall from an external source, an authentication mechanism must be incorporated into the firewall to ensure that only legitimate users manipulate the box's configuration. This security mechanism is not considered in their design and is required for real world deployment.

In addition to the above literature, a number of proposals have been submitted to the IETF in the form of Internet Draft reports [5]. This problem is significant enough that an IETF working group, Middlebox Communication (Midcom), was formed to initiate the design of a protocol to allow an external entity to control a middlebox.

The remaining part of this article is organized as follows. Section II describes the components of the new Secure Telephony Enabled Middlebox (STEM) architecture and the protocols used for communication between them. Section III examines how

[+]This work as support by NSF grants NCR-9703275 and ANI-9741668

the architecture handles several of the most common call scenarios. Section IV enumerates the major network vulnerabilities and shows how protection mechanisms have been incorporated to guard against them. Section V explains how the architecture is implemented and tested. Section VI provides a conclusion and an outline of future work.

II. Secure Telephony Enabled Middlebox Architecture

As the previous work has shown, simply designing a single device is not sufficient to solve the problem. The solution must be a system comprised of several devices working together to provide the necessary functionality. The Secure Telephony Enabled Middlebox (STEM) architecture is designed to address the problems of deploying dynamic applications in an enterprise network. A core middlebox will be present, but will be internally different from those deployed today. This new firewall will be capable of interpreting currently deployed applications as well as adapt to handle future ones. To provide this capability, dynamically loadable parsers for different applications will be built. A logical entity called the Security Manager implements the security policy for the network and the enforcement mechanisms to control the devices providing connectivity between the enterprise LAN and other networks.

A. Architecture Components

The STEM architecture shown in Figure 1 is similar to the basic network required for SIP deployments [1]. The firewall component was included in the STEM architecture because it is an essential component of an enterprise's network as well as the source of many problems for dynamic applications. The Security Manager (SM) is a logical component that will be used to control the operation of the firewall and media / signaling (M/S) gateways within the network. The SM also incorporates the security policy settings for the enterprise domain that affect how IP telephony services operate. The SIP proxy allows incoming calls from the Internet by relaying SIP control messages. The final element in the STEM architecture is the IP telephony enabled terminals. These can either be software running on workstations or IP phones.

Security Manager: The Security Manager's primary function is to ensure the network consistently operates a high level of security without degrading the level of service. It is a logical entity that has its various functional components implemented on top of different physical components in the network. The elements of the SM include a database mapping IP telephony addresses to machine address in the enterprise, a call preference database with an entry for each employee, the various threshold levels that are triggered when the network is under attack and the enforcement mechanisms to ensure only authorized users are allowed to use IP telephony services.

- The SM database contains the mapping between user addresses (SIP URL [1]) and the machine addresses (IP address) at which the user is currently receiving calls. This can be implemented by the SIP Location Server. With users becoming more mobile, it is necessary to maintain a relationship between the called user address and the network address of the terminal at which the user is located. When the firewall receives a SIP request message it informs the SM which translates the SIP URL to the appropriate machine address. With the external calling party using the user address and not the machine address, it is irrelevant if the network behind the firewall is using private machine addresses. At least one interface on the firewall must belong to the internal network and therefore be able to route either public or private addresses. The SM also knows if the called user is connected to the internal network via a Virtual Private Network (VPN) tunnel and appropriately instructs the firewall.
- The database containing the users call profile can be placed at the SIP proxy or be added to the Location Server. The profile contains information about incoming call preferences and a listing of SPAM addresses specified by the user. Upon authentication, a user can configure their incoming call preference at the SM. During an incoming call, a query of the called user's profile will determine the SM's response to the firewall. The list of options includes, but is not limited to: automatically accept the call, forward the call to another user or service (e.g. voice mail), automatically drop the call or send a query message to the user with the calling parties information and request further response.
- The other database the SM must have access to, but does not necessarily maintain, is the user authentication database. It is assumed that most enterprise networks require users to log into terminals and a global repository of user authentication tokens is attached to the network. To help fight the problem of too many passwords, when the SM authenticates the user it acts as a proxy and forward messages between the user terminal and the authentication server. Successfully authenticating to the enterprise's authentication server proves to the SM the user can be trusted.

Firewall: The overall function of the firewall still remains to allow only certain traffic to cross between its interfaces. However, the internal structure of the firewall in the STEM architecture is an enhancement upon the existing firewall designs. Conventional packet filter firewalls operate at the network and/or transport layers and are therefore unaware of the application layer. We outline the key aspects of the firewall architecture that is aware of the application running over the network and transport layers. The block diagram of the key components shown in Figure 2 is an extension of the firewall architecture in [4]. All the components of the firewall are created as self-contained entities with a common interface layer, which they interact thereby allowing the inter-component dependence to be kept to a minimum. By creating each component independent of others, they can be dynamically loaded and unloaded without taking the entire device offline. This will allow network administrators to maximize network up-time while performing maintenance on one specific component of the firewall.

[1] For information about the operation and requirements of standard SIP components refer to RFC 2543 published by the IETF [1].

• *Pattern Matcher:* The Pattern Matcher is the most basic component in the firewall and all packet filter firewalls include this component. It allows configuration of static rule-sets using machine addresses, transport protocols, and port number specifications [6]. Each rule-set has an action assigned to it that is executed when the rule is triggered.

• *Protocol Parser:* The most important block in the firewall architecture is the Protocol Parser. This component is comprised of multiple parsers. Each parser is designed to understand the operation of a single complex protocol.

The SIP parser includes a call monitor component that is responsible for ensuring that each call follows the protocol specified state transitions. The dynamic port numbers are extracted from the call setup and passed to the Pattern Matcher to open the appropriate pinholes. There is the possibility of two calls selecting the same port numbers. Therefore, the SIP parser must monitor both the port and internal IP address associated with a data stream. It will instruct the Pattern Matcher to completely close a port only after all streams have terminated. Additionally, the parser also extracts the codecs advertised by the terminals during call setup. This information is passed to the Flow Monitor to detect malicious streams.

• *Flow Monitor:* The Flow Monitor is designed to handle malicious data streams. It monitors the data rate of the call streams and if they exceed the threshold set by the bandwidth requirements advertised during the call setup the firewall can respond. The triggered response can be set by the individual network administrators and could include dropping packets of the malicious stream or applying a traffic throttling algorithm.

• *External Interface:* The External Interface component allows the firewall to communicate with other devices. It is responsible for parsing incoming messages for other components and generating appropriate response messages.

Media / Signaling Gateway: The media / signaling (M/S) gateway is responsible for translating calls between a circuit switched network and a packet switched network. In the STEM architecture, the M/S gateway will also have to interact with the SM. Similar to the firewall, it will have to request how the called user wishes to handle an incoming call. To prevent unauthorized users from connecting to the network and initiating calls using the M/S gateway, all users must authenticate with the SM before the gateway will allow outgoing calls to be placed.

User Terminals: Two types of User Terminals exist on an enterprise's network; a personal computer with IP telephony software and a dedicated hardware device capable of IP telephony functions (i.e., a IP telephone). Both types must be capable of communicating with the SM in addition to understanding SIP (or H.323).

B. Protocols

The STEM architecture requires two control protocols to function. One is used between the SM and the enforcement devices (firewall and M/S gateway) and the other between the SM and the User Terminals.

Security Manager to Enforcement Devices: The frequency and volume of communication between the SM and the firewall or M/S gateway is high enough that a permanent connection should be established. To improve the security of this connection, it could be established out-of-band with respect to the local area network. Isolating the connection to a separate physical medium reduces the chance of a third party being able to tamper with it. At the very least, the communication should be strongly encrypted. The functionality of the protocol itself is being developed by the IETF Midcom Working Group. Their work has not yet resulted in a complete protocol design, but it has resulted in the creation of both a protocol requirements [7] and architectural framework [8] document. The protocol capable of fulfilling the requirements outlined in these two documents will be able to perform all necessary functions the STEM architecture requires.

Security Manager to Terminals: Communication between the Security Manager and the User Terminals is different than that between the SM and the firewall. The SM can be communicating with thousands of terminals simultaneously and only for a brief period of time. The protocol used must be lightweight and provide several key functionalities. It must allow the users to be authenticated and be able to protect the content of the messages from eavesdropping during transmission.

III. Call Setup Scenarios

A large number of call scenarios exist in a converge network comprised of the PSTN and an IP network. In this study we consider three categories, namely, Net-to-Net, Phone-to-Net and Net-to-Phone calls.

A. Net-to-Net

All Net-to-Net calls must pass through the firewall in the STEM architecture. Both incoming and outgoing calls are handled in a similar manner with the exception of the initial call setup. The numbers within the parenthesis in Figure 3 indicate the step in the sequence of interactions involved in an incoming Net-to-Net call.

The calling terminal sends a TCP SYN packet to port 5060 (well known port of SIP server) of the destination terminal. The SIP Protocol Parser in the firewall receives this packet and identifies the destination port. It forwards all incoming TCP SYN packets to the enterprise's SIP Proxy regardless of the destination IP address of the packet. The SIP Proxy completes the three-way handshake with the calling party. At this point the calling party sends the SIP INVITE request over the TCP connection. The Protocol Parser identifies the request and contacts the SM with the relevant information. The SM responds to allow the call and provides the current IP address of the user. The request is passed to the SIP Proxy and forwarded to the destination terminal.

For the rest of the call setup the SM is not involved and the Protocol Parser assumes a passive role, extracting information and passing it to other components within the firewall. While all call control messages are relayed through the SIP Proxy, the RTP stream is created directly between the calling and called terminals. Call termination is achieved in one of two ways: 1) the Protocol Parser detects a BYE message from one of the terminals, or 2) the Flow Monitor does not observe any traffic for the corresponding data stream over a given interval. Upon detecting a call termination, the Pattern Matcher is instructed to call all pinholes that were being used in the call.

For outgoing calls, users must inform the SM of the machine address they are calling. The SM instructs the firewall to allow the outgoing SYN packet. After the TCP connection is setup, the Protocol Parser and the firewall operate in the same manner as with an incoming call.

This category of calls also includes users connected to the enterprise network remotely (e.g. through a VPN). The call model for this scenario is essentially two Net-to-Net calls with the users Home Agent [9] acting as the bridge between the two calls.

B. Phone-to-Net

In a Phone-to-Net call, a terminal connected to the PSTN initiates a call to a terminal in the IP network. This cross network call happens without either end terminal knowing the other is on a different network because of the ENUM extension to the DNS service [10]. The Signaling System 7 (SS7) network uses the information within the dialed number to route the call to the responsible M/S gateway. The incoming call is assigned a voice port on the PSTN side of the gateway. The M/S gateway resolves the dialed phone number into a SIP address. This is done by submitting the dialed number to the SM and receiving the IP address of the end terminal. The gateway must then establish a SIP connection with the called terminal. At this point the gateway translates messages between the PSTN and the IP network.

C. Net-to-Phone

The Net-to-Phone call flow is similar to the Net-to-Net call but the call is routed through the M/S gateway and out over the PSTN instead of through the firewall and over the Internet. The sequence of interactions between the STEM architecture components in a Net-to-Phone call is shown in Figure 4.

Before a user can make a call they must authenticate themselves to the SM. After completing the authentication process, the caller must inform the SM of their intent to make an outgoing call to the PSTN. The SM will instruct the M/S gateway to accept an outgoing call from the user's terminal. A TCP connection is established directly between the calling terminal and the M/S gateway. The calling user sends the SIP INVITE message over the TCP connection to the gateway. It is the responsibility of the gateway to convert the INVITE message into the appropriate Signaling System 7 (SS7) messages. A voice port within the gateway is allocated for the call and the SS7 messages are sent out to setup the call in the PSTN. When the call terminates, the gateway deallocates the voice port and terminates the TCP connection with the calling terminal.

IV. Network Vulnerabilities and Counter-measures

There are a large number of different types of vulnerabilities present in converged networks. This section focuses on a small key subset and discusses how the counter-measures are implemented in the STEM architecture to reduce the impact of these vulnerabilities.

A. Denial of Service

The class of attacks labeled as denial of service (DoS) is very large. Therefore, four attacks specifically targeted at converged networks have been investigated.

Net-to-Net:

- *TCP SYN Flood:* A TCP SYN flood is launched by an attacker who sends a large number of TCP SYN packets to a destination terminal [11]. Within the STEM architecture, the SIP Proxy will be the target of a SYN flood. The result of the attack is that the target is unable to accept any new connections. It is difficult to detect a SYN flood if the inspection device does not maintain some form of state. In the STEM architecture SYN floods are handled by the firewall, thus preventing them from penetrating the internal network. The Flow Monitor keeps a running counter of incoming SYN packets for each source IP addresses. If the number of incoming SYN packets exceeds a certain threshold, the Flow Monitor instructs the Pattern Matcher to drop any future SYN packets from that address for a given duration.
- *SIP INVITE Flood:* An INVITE request flood is specific to IP telephony. The attacker establishes a legitimate TCP connection with the target and then proceeds to generate a large volume of SIP INVITE requests. As a result, the user or SIP server is overwhelmed with incoming call requests. The SIP protocol allows for multiple INVITE messages to be sent between terminals to implement services like call hold and call park. The state machine, Figure 5, in the SIP Protocol Parser allows multiple INVITE messages to pass. However, there is a tolerance level within the parser to catch misuse of this capability. A counter monitors the number of INVITE messages sent per stream over a given interval of time. If preset threshold levels are exceeded the parser can instruct the Pattern Matcher to drop subsequent control and audio packets associated for that call stream.

• *Malicious RTP Streams:* Another DoS attack specific to an IP telephony environment is the use of malicious RTP streams to saturate a network link. To accomplish this the attacker initiates a call to the target. When the two parties begin exchanging voice data the attacker constructs very large RTP packets as shown in Figure 6 or sends multiple packets with the same RTP sequence numbers. The target may not be aware they are under attack if the extra data is converted into frequencies that humans are unable to hear or are dropped because of duplicate sequence numbering. To prevent these types of attacks, the firewall relies on the Protocol Parser and the Flow Monitor. The codecs and bandwidth requirements for each call are announced during setup and extracted by the parser. This information is provided to the Flow Monitor which monitors the data rate for each call stream. If a stream is detected exceeding some defined threshold, it can be terminated by the firewall or throttled to a lower data rate.

Phone-to-Net and Net-to-Phone: The limited number of voice ports in the M/S gateway makes it a good target for denial of service attacks. Within the STEM architecture, a DoS from the internal LAN is not possible. A user must first authenticate and inform the SM before an outgoing PSTN call can be made. Therefore, it is impossible for an unauthorized user to create a flood of outgoing calls that utilizes all the voice ports in the M/S gateway.

A DoS attack launched from the PSTN has the potential of tying up all the voice ports and perhaps even saturating the signaling link. These types of attacks can be countered by using different types of throttles available in the SS7 protocol. One potential approach could be to have the M/S gateway send out Transfer Controlled (TFC) messages [12] when all the voice ports are occupied. These messages will cause the upstream Signaling Transfer Point (STP) [12] to throttle call setup requests sent to the M/S gateway. Additionally, application level call gapping methods can be designed to mitigate these types of DoS attacks.

B. Eavesdropping

Within the STEM architecture there are two information flows susceptible to eavesdropping: the control flow and the data flow. The control flows in the STEM architecture include the communication between User Terminals and the SM, the SM and the firewall, and the SM and the M/S gateway. The information within these flows can contain user authentication and device configuration messages. Protecting this information from reuse by unauthorized parties is essential. The STEM architecture ensures the protection of the data by using protocols that encrypt the data section of each packet at the end terminal. Furthermore, to detect session replay attacks a time stamp is included in the encrypted section of the packet. The receive terminal can therefore check the time the packet was created and transmitted. Packets containing time stamps outside a threshold will be discarded.

Any communication over the network not classified as a control flow is a data flow. The STEM architecture does not directly protect against data flow eavesdropping. Each application protocol should implement a form of payload encryption to guard against eavesdropping. Both SIP and H.323 have provisions defined in the protocol to provide this feature. Within the STEM architecture, all IP telephony terminals must implement the encryption functionality outlined in the protocol specifications. For SIP this includes both the hop-by-hop payload encryption as well as the hop-by-hop encryption of the via field to hide the route a flow traverses.

C. Other Network Attacks

Internal Local Area Network DoS: The possibility that a user within an enterprise will launch a DoS attack against another internal user does exist. However, the STEM architecture does not include any counter-measures to this type of attack. The attack path does not include the firewall or the SM; therefore, preventing this type of attack is not possible with this architecture.

Session Hijacking: Another network based attack is session hijacking. This attack allows an unauthorized third party to impersonate one end of the session. The ability to perform this attack stems from the design of the transport and link layer protocols. It is not possible to protect against this type of attack from the device level. An integrity checking technology (e.g. IPSec) must be used to ensure the parties are who they claimed to be.

Virii, Trojan Horses and Spy-ware: The release and spread of virii and Trojan horses continues to increase each year. The STEM architecture is able to help stop malicious software that "phones home" (initiates a network connection to an external machine). The firewall's Pattern Matcher and Protocol Parsers are able to block well known Trojan ports in addition to detecting Trojans that masquerade as a legitimate application but do not transmit the correct data. Host only virii must still be dealt with by anti-virus software.

V. Implementation of the STEM Architecture

A prototype of the STEM architecture is currently being developed to evaluate the security of the system and determine basic performance metrics. The firewall device is being built upon the Linux 2.4 series kernel and uses the Netfilter framework [13] to implement the SIP Protocol Parser, Call Monitor and Flow Monitor. Several open source SIP softphones and server implementations will be used for testing to ensure that the firewall operates correctly. The Security Manager will be built as a stand alone application and initially will use a proprietary protocol to communicate with the firewall. When the IETF's Midcom Working Group publishes the protocol guidelines, the SM and the firewall will be modified to use this protocol. Both home-grown and publicly available tools will be used to test the security aspects of the system. The system will also undergo several stress tests to determine the performance capabilities and scalability.

VI. Conclusions and The Future

The deployment of IP telephony and other dynamic applications will allow enterprises to become more cost effective and offer a higher level of integration. However, before these applications can be widely adopted several major obstacles must be overcome. In this article it was shown that dynamic applications cannot function properly when operating over static network devices. The previous work on this problem has brought a solution closer to reality but not completely. The Secure Telephony Enabled Middlebox (STEM) architecture presented in this paper is a comprehensive solution to the problem. The STEM architecture was designed to be technologically neutral, thereby allowing it to work in a diverse range of network environments. Ensuring the security of the network was a top priority in the STEM architecture. Major network-based security vulnerabilities present in an IP telephony deployment have been enumerated and the counter-measures within the STEM architecture have been discussed. In the near future, patches and hacks to existing networks will be used in enterprises wishing to deploy dynamic applications. It will take several years for vendors to develop solutions using technology like STEM to correctly solve the problems of dynamic applications and static network devices.

References

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

[2] ITU-T Recommendation H.323, "Visual Telephone Systems and Equipment for Local Area Networks which Provide a Non-guaranteed Quality of Service," May 1996.

[3] J. Kuthan, "Internet telephony traversal across decomposed firewalls and NATs," Proceedings of the 2nd IP Telephony Workshop, New York, April 2001.

[4] U. Roedig, R. Ackermann, and R. Steinmetz, "Evaluating and improving firewalls for IP-telephony environments," Proceedings of the 1st IP Telephony Workshop, Berlin, April 2000.

[5] J. Rosenberg, R. Mahy, and S. Sen, "Internet Draft: NAT and firewall scenarios and solutions for SIP," Work in Progress, November 2001.

[6] Richard Stevens, *TCP/IP Illustrated Volume 1: The Protocols*, vol. 1, Addison Wesley Longman, Inc., Reading, MS, 1st edition, 1994.

[7] R. Swale, P. Mart, P. Sijben, S. Brim, and M. Shore, "Internet Draft: Middlebox communications protocol requirements," Work in Progress, November 2001.

[8] P. Srisuresh, J. Kuthan, J. Rosenberg, A. Molitor, and A. Rayan, "Internet Draft: Middlebox communication architecture and framework," Work in Progress, December 2001.

[9] C. Perkins, "RFC 2002: IP Mobility Support," October 1999.

[10] P. Faltstrom, "RFC 2916: E.164 number and DNS," September 2000.

[11] CERT Coordination Center, "CERT Advisory CA-1996-21: TCP SYN flooding and IP spoofing attacks," November 2000.

[12] Travis Russell, *Signaling System 7*, McGraw-Hill, New York, NY, 3rd edition, 2000.

[13] "Netfilter - firewalling, NAT, and packet mangling for Linux 2.4," Web Site - http://www.netfilter.org/, 2002.

Brennen Reynolds is a M.S. candidate at the University of California, Davis. His research interests are in IP telephony, security in converged networks, intrusion detection systems and network-based vulnerabilities. He received his B.S. in computer engineering from the University of California, Davis, USA in 2001.

Dipak Ghosal received his B.Tech degree in Electrical Engineering from Indian Institute of Technology, Kanpur, India, in 1983, MS degree in Computer Science from Indian Institute of Science, Bangalore, India, in 1985, and Ph.D degree in Computer Science from University of Louisiana, Lafayette, USA, in 1988. From 1988 to 1990 he was a Research Associate at the Institute for Advanced Computer Studies at University of Maryland (UMIACS) at College Park, USA. From 1990 to 1996 he was a Member of Technical Staff at Bell Communications Research (Bellcore) at Red Bank, USA. Currently, he is with the faculty of the Computer Science Department at the University of California at Davis, USA. His research interests are in the areas of IP telephony, peer-to-peer systems, mobile and adhoc networks, and performance evaluation of computer and communication systems.

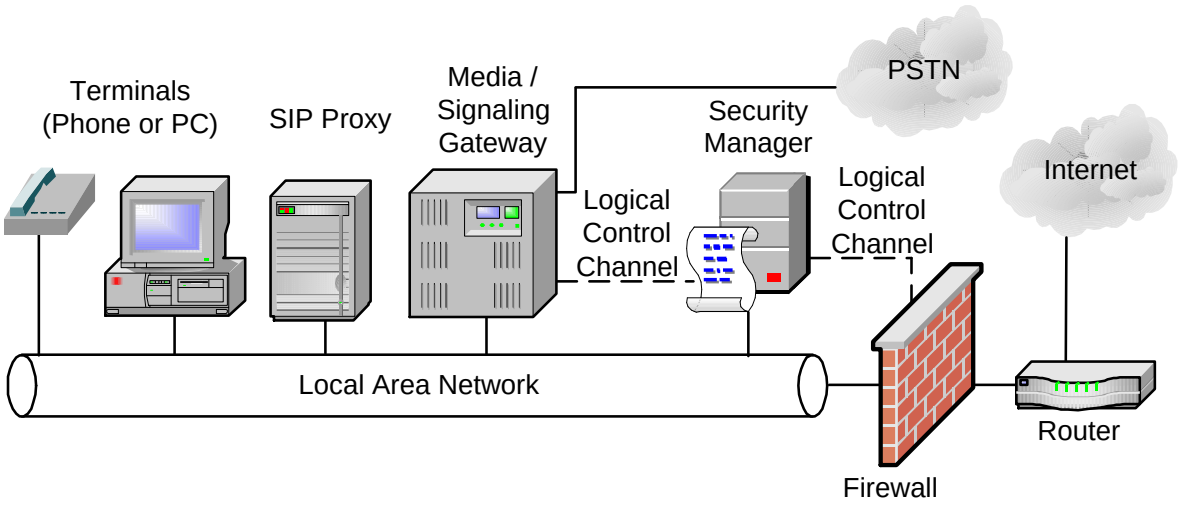

Fig. 1. Secure Telephony Enabled Middlebox Network Entities

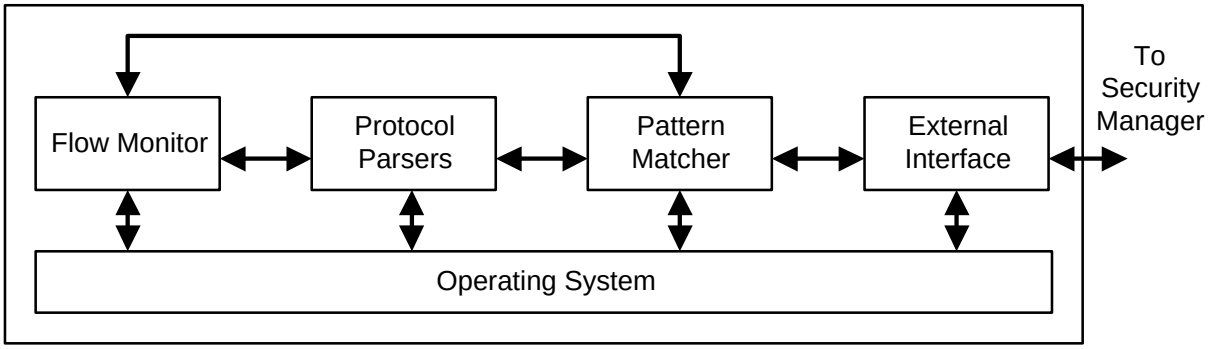

Fig. 2. Firewall Architecture Block Diagram

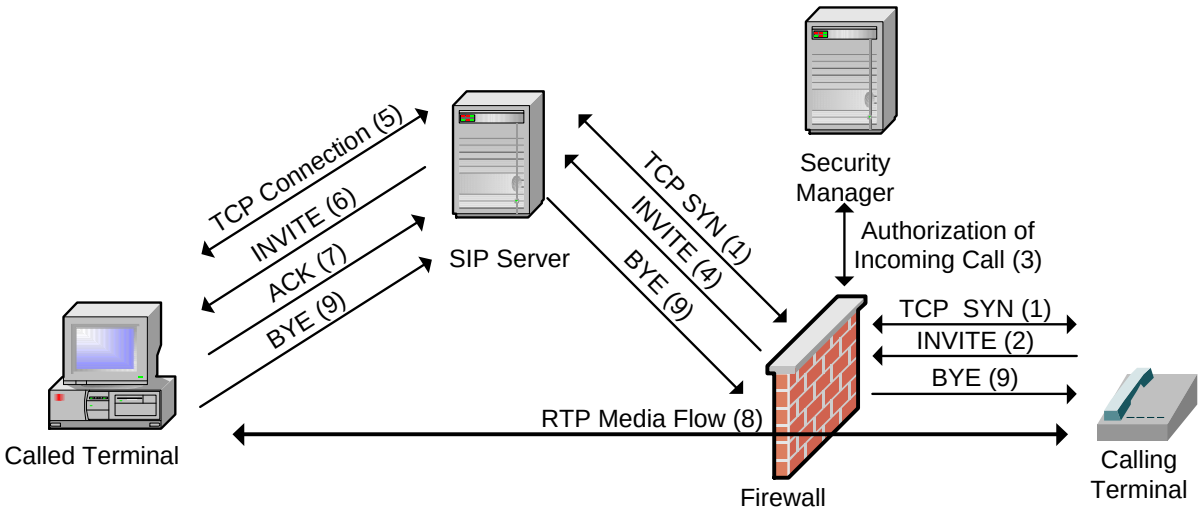

Fig. 3. Incoming Net-to-Net Call Flow

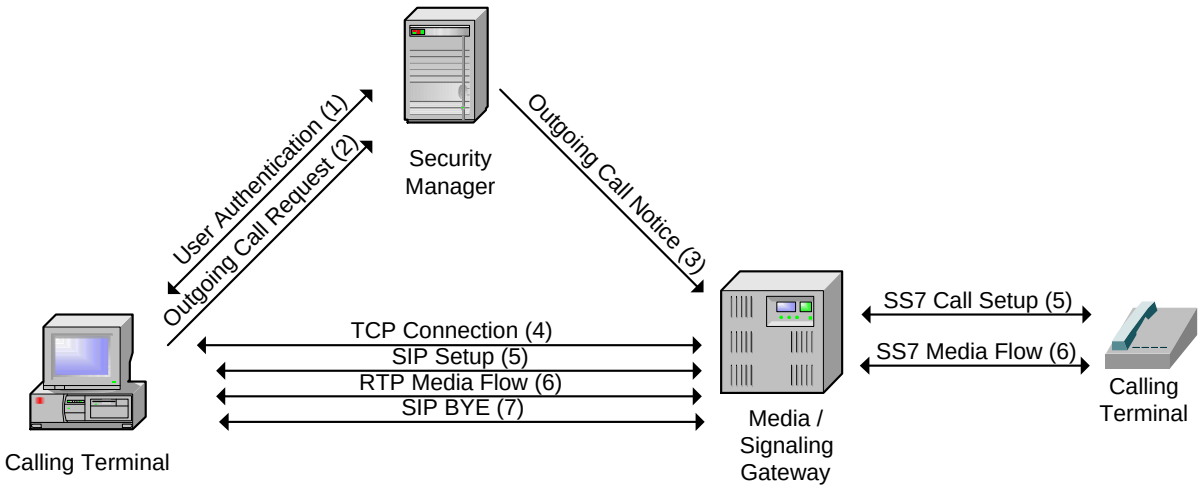

Fig. 4. Net-to-Phone Call Flow

SIP Stream ID	# of INVITE / unit time
1	1
2	3
3	25
4	2

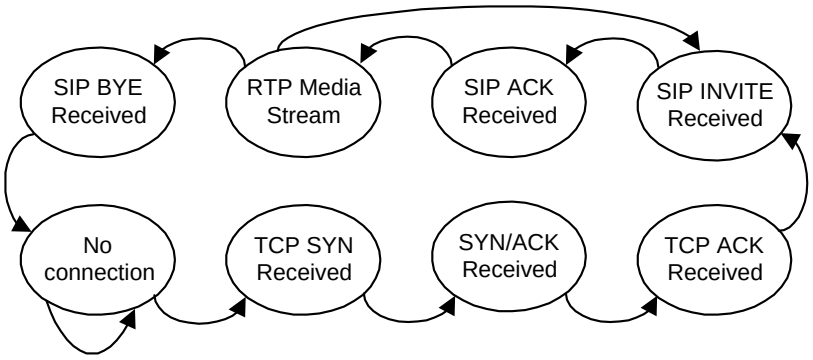

Fig. 5. SIP Protocol Parser State Machine and Tolerance Table

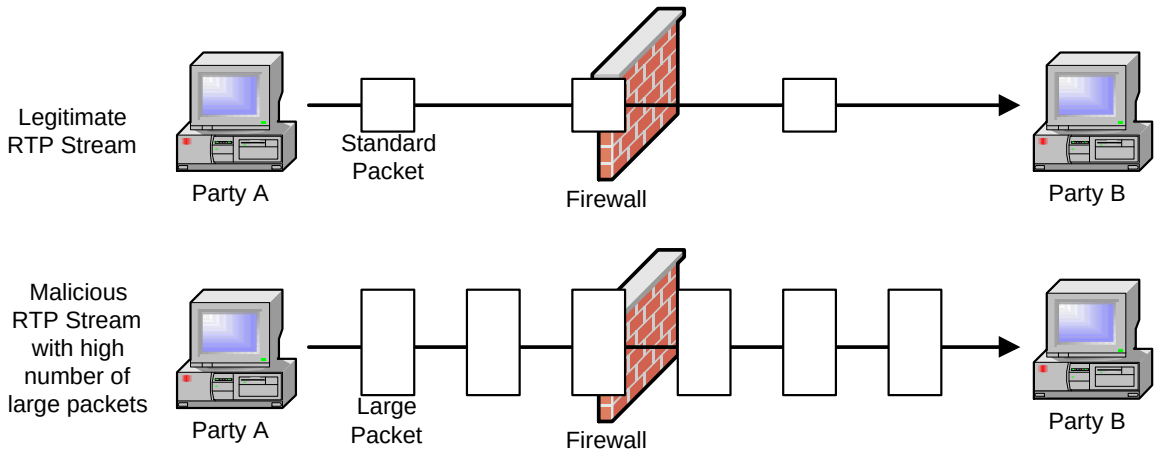

Fig. 6. Legitimate and Malicious RTP Packet Streams