A Secure Web-based Global Management System For Firewall/VPN Devices

Mi-Joung Choi and James W. Hong
Department of Computer Science and Engineering
Pohang University of Science and Technology
{mjchoi, jwkhong}@postech.ac.kr

Abstract

A firewall is a security device placed between a private network and a public network such as the Internet. It is designed to protect the private network resources from unauthorized user access. Today, various firewalls are widely used in many places (e.g., Internet data centers, company headquarters, branch office, telecommuters' homes). What is desperately needed is a management system that can easily configure, monitor and manage multi-site deployed firewalls from a central location. For flexibility, such a management system must be divided into components and needs to use an open management protocol, such as the Simple Network Management Protocol (SNMP). Yet the SNMP has a security defect. Further, the proposed standard Management Information Base (MIB) for firewalls is insufficient for supporting centralized global management of a lot of firewall devices. In this paper, we present the design and implementation of a secure Web and SNMP-based global firewall management system. We have focused on two aspects: 1) extending the existing proposed standard MIB to support the configuration and monitoring of hundreds or thousands of firewall and VPN devices; 2) providing secure communication among global manager system components in order to provide secure firewall management. We also present our work on developing our firewall global manager (FGM) on commercial firewall/VPN devices.

Keywords

Firewall, VPN, Global Management, Secure Communication, SNMP, MIB, Web-based Management System

I. INTRODUCTION

The Internet has made vast amounts of information available to various users. For many people, having access to this information is no longer just an advantage, it is essential. But the Internet is a publicly accessible network, so commercial transactions on the Internet are often concerned about security risks. The most popular and easiest way to reduce security risks is to use firewalls [1, 2, 3].

A firewall is a security device placed between a private network and public network (such as the Internet). It is designed to prevent unauthorized users from gaining access to confidential corporate and customer resources in the private network. Using firewalls, organizations can protect private network resources from security problems. For this reason, firewalls are generally used in many organizations. In large organizations and data centers, the deployment of tens or hundreds of firewalls is common. In many cases, these firewalls may be geographically distributed. Thus, a management system for multi-site deployed firewalls from a central location is necessary.

Today, firewall management systems use proprietary protocols and manage only internal firewall products. Due to the use of a proprietary protocol, the management system cannot manage various firewalls from a central location. To solve this problem, we consider an open management protocol. Using Simple Network Management Protocol (SNMP) [4], the management system can manage diverse firewalls equipped with an SNMP agent. In general, firewalls are not equipped with an SNMP agent for security reasons and a firewall MIB [5] is not sufficient. Therefore, an MIB for firewalls and security communication consideration of SNMP is necessary.

Firewall devices are diverse and the diffusion environments of firewalls are also various. To apply a firewall management system to any circumstance, the management system must provide flexibility. Consequently, the architecture of the firewall management system must be divided into management components during system design. For this reason, communication between management elements must be secure. Moreover, the firewall is a device, which guarantees security and thus the management system for firewalls itself must be foolproof. For this, the security mechanism must be supported in communications between the management elements.

In this paper, we present the design and implementation of a secure Web and SNMP-based global firewall management system. We have mainly focused our research and

development on two aspects: 1) extending the existing proposed standard MIB to support the configuration and monitoring of hundreds or thousands of firewall and VPN devices; 2) providing secure communication among global manager system components in order to provide secure firewall management. We also present our work on developing our firewall global manager (FGM) on commercial firewall devices.

The organization of this paper is as follows. In Section 2, we present related work to the firewall and security issues. In Section 3, we examine the proposed standard MIB for firewalls and present an extended MIB which satisfies the requirements for global management of firewall devices. In Sections 4 and 5, the requirements and design for a firewall global manager are presented, respectively. In Section 6, we describe the implementation of our proposed FGM architecture. In Section 7, we briefly compare our FGM with firewall management systems by focusing on their features. In Section 8, we summarize our work and discuss possible future work.

II. RELATED WORK

In this section, we briefly overview firewalls. Further, we describe the proposed standard firewall monitoring MIB [5]. We also examine security protocol mechanisms for client and server communication.

2.1. FIREWALL

A firewall protects networked resources from hostile intrusion that can compromise confidentiality or result in data corruption or denial of service. It may be a hardware device or a software program running on a secure host computer. Figure 1 shows a hardware firewall providing protection to a local network [6].

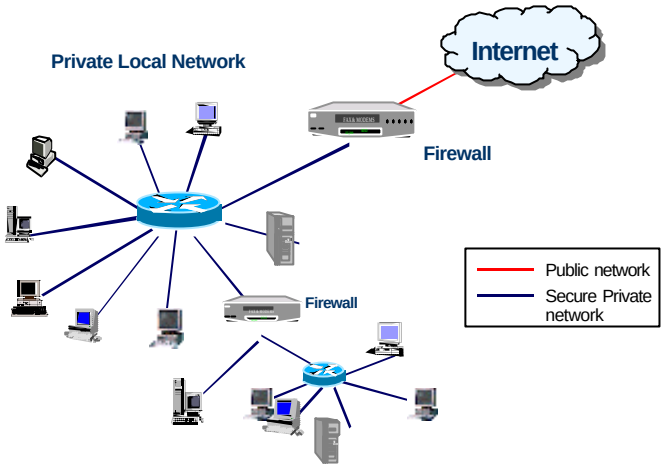

Figure 1. Firewall

The firewall monitors requests for access and authorizes individual users using various authentication methods. The Firewall stores each request for access to a private network - whether attempts are successful or not. The firewall will immediately alert the designated network administrator of any suspicious activity.

In addition to protecting the internal network from an external attack, a firewall can also be used to prevent the abuse of Internet resources from within, by logging and controlling the use of the Internet by users on a local area network (LAN). By logging every connection between the LAN and the Internet, firewalls enable managers to account for Internet usage by employees, making those individuals or departments responsible. Internal abuse of Internet resources can be quickly identified and terminated, achieving in complete control of internal network operations

Moreover, most firewall devices have Virtual Private Networks (VPN) [7] functionalities. VPN offers a secure way of extending services on the corporate network to customers and business partners, IPSec-encrypted VPN tunnels are supported by most firewall devices. VPN enables businesses to achieve a secure connectivity to remote and branch offices without the expense of leased lines, creating another potential revenue stream for service providers.

In this paper, firewall devices mean firewall and VPN devices; that is, firewall devices support their essential functionalities and VPN functionalities.

2.2. FIREWALL MIB

The firewall MIB [5] defines a portion of the Management Information Base (MIB) for use with network management protocols in TCP/IP-based Internets. In particular, it defines objects for monitoring firewall devices. The objects of the firewall MIB are arranged into the following groups: service identifiers (service), firewall event variables and logs (fwevent), firewall status and statistics data (fwquery), and firewall traps (fwtrap). These groups are defined to provide a means of assigning object identifiers, and to provide a method for managed agents to know which objects they must implement.

2.2.1. THE SERVICE IDENTIFIERS GROUP

The service group defines object identifiers (OIDs) for resources, classes of services, and particular services handled by firewalls. These OIDs are used as values in variables in other groups of the MIB to designate a service. In this document and MIB definition, "resource" is defined as any service, application, proxy, hardware unit, utility, operating system, product, engine, etc. on the firewall. Resource can also refer to the firewall as a whole. Further, the term "service" is used interchangeably with "resource" throughout the document and the MIB.

2.2.2. THE FIREWALL EVENT VARIABLES AND LOGS GROUP

The fwevent group defines tables for logging events that take place on the firewall. Management stations are notified of the events via traps from the fwtrap group.

2.2.3. THE STATUS AND STATISTICS GROUP

The fwquery group contains status and statistical information. It includes version information for the firewall and its resources and services. It includes version information, status details, and statistics measured by firewall resources and services.

2.2.4. THE FIREWALL TRAPS GROUP

The fwtrap group defines the traps that a firewall can send. When an event occurs on the firewall, the basic table information is collected, and based on the event, a details table is chosen and its information is collected as well. This information is stored on the firewall and a trap from the fwtrap group is sent. The trap contains the same information contained in the basic table.

The purpose of the MIB defined in firewall monitoring MIB [5] is to provide information for the purpose of monitoring firewall activity. The objects defined here provide information about urgent events, security, health and status, and the performance of a firewall. This information is provided in two ways, via traps and through objects that must be queried. The traps also have associated information that can be queried.

The Cisco firewall MIB [8] provides a good example defining a private firewall monitoring MIB. This MIB is based on the IETF firewall monitoring MIB [5].

2.3. SSL

The Secure Socket Layer (SSL) [9] protocol runs above the Transmission Control Protocol/Internet Protocol (TCP/IP) [10] and below the higher-level protocols, such as HyperText Transport Protocol (HTTP) [11], Lightweight Directory Access Protocol (LDAP) [12], or Internet Messaging Access Protocol (IMAP) [13]. It uses TCP/IP on behalf of the higher-level protocols, and in the process allows an SSL-enabled server to authenticate itself to an SSL-enabled client, allowing the client to authenticate itself to the server, and allowing both machines to establish an encrypted connection.

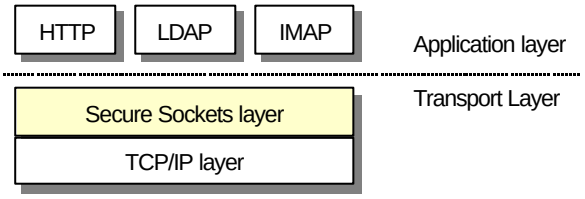

Figure 2. SSL between TCP/IP and high-level application protocols

SSL server authentication allows a user to confirm a server's identity. SSL-enabled client software can use standard techniques of public-key cryptography to verify that a server's certificate and public ID are valid and have been issued by a Certificate Authority (CA) listed in the client's list of trusted CAs. SSL client authentication allows a server to confirm a user's identity using the same techniques as those for server authentication.

An encrypted SSL connection requires all information sent between a client and a server to be encrypted by the sending software and decrypted by the receiving software, thus providing a high degree of confidentiality. Confidentiality is important for both parties in any private transaction. In addition, all data sent over an encrypted SSL connection is protected with a mechanism for detecting tampering--that is, for automatically determining whether the data has been altered in transit.

2.4. HTTPS

The secure HyperText Transfer Protocol (HTTPS) [15] is a communications protocol designed to transfer encrypted information between computers over the World Wide Web (WWW). HTTPS is HTTP using a Secure Socket Layer (SSL) [14]. A secure socket layer is an encryption protocol invoked on a Web server that uses HTTPS. In other words, the main difference is the introduction of a set of new properties and events that deal with SSL security. The HTTPS component implements a standard HTTP client through a simple plug-and-play interface and the added option of SSL security.

Most implementations of the HTTPS protocol involve online purchasing or the exchange of private information. Accessing a secure server often requires some sort of registration, login, or purchase. The HTTPS component supports the HTTP basic authentication scheme through user and password properties. Other authentication schemes can be implemented by using the authorization property.

2.5. IPSEC

The Internet Engineering Task Force's (IETF) IP Security (IPSec) [16] Working Group is developing standards for IP-layer security mechanisms for both IPv4 (the version currently used on the Internet at the time of this writing) and IPv6 (the next generation of TCP/IP). The IPSec architecture includes authentication (how to know if the site communicating to your site really is who it claims to be) and encryption. These mechanisms can be used together or independently.

IPSec is a series of guidelines for the protection of Internet Protocol (IP) communications. It specifies ways for securing private information transmitted over public networks. Services supported by IPSec include confidentiality (encryption), authenticity (proof of sender), integrity (detection of data tampering), and replay protection (defense against unauthorized re-sending of data). IPSec also specifies methodologies for key management. Internet Key Exchange (IKE) [17], the IPSec key management protocol, is a series of steps that establishes keys for encrypting and decrypting information. It defines a common language on which communications between the two parties is based. IPSec and IKE together standardize the performance of data protection, thus making it possible for security systems developed by different vendors to interoperate.

III. EXTENDED FIREWALL MIB

The proposed standard firewall MIB is not ideal for reporting on the configuration of a firewall or for globally management. Currently, most firewalls use unique and/or proprietary protocols and representations for dealing with configuration and 'policy'. This MIB does not have much variables related to configuration items. It would be too difficult to try to create a generic set of MIB objects that could represent most firewall configurations. However, firewall MIBs for configuration and policy are necessary [5]. Therefore, we extend the firewall MIB for firewall configurations.

For extending the MIB for firewall configuration, we first must consider the role of firewall. The main functions of a firewall [18] are to control access according to management policies and to configure the Virtual Private Networks (VPN) [7]. Therefore, to provide firewall management functionality, the MIB should include, at minimum, the following five parts: system information, policy definition, VPN settings, log monitoring data, hardware status.

Generally, the firewall has three interfaces: an internal, an external, and a DeMilitarized Zone (DMZ). The internal interface is for private internal networks, the external interface is connected to the public networks. The interface for a DMZ is an isolated, protected network that allows users to host public services, such as a Web or FTP site, on local servers. Firewall system information, such as the system name, DNS server, system date, default gateway, and the IP address and netmask of the three interfaces needs to be configured. The MIB for system information contains all of the above information.

To protect the internal network, security policies that define a set of rules for network traffic must be created. For example, we can allow only a certain machine or subnetworks to send outgoing traffic to the Internet, or restrict outgoing traffic to only certain protocols, such as HTTP or FTP at the specific time. The policy needs to define the source and destination address, service type, scheduling, logging or not, action, quality of service (QoS), and so on. Figure 3 shows the entries of policy tables.

The schedule is defined as a MIB table, and an entry of the table contains the schedule name, start & end times (year, month, day, hour, minutes). The policySchedule of the policy table in Figure 3 is the schedule name of the schedule table. The other entries in the policy table are defined as MIB tables. Despite some complexity, administrators can manage different types of policies.

```
• table entry of policy MIB
  PolicyListEntry ::= SEQUENCE {
      PolicyNo            INTEGER,
      PolicySource        DisplayString,
      PolicyDestination   DisplayString,
      PolicyService       DisplayString,
      PolicySchedule      DisplayString,
      PolicyLogging       DisplayString,
      PolicyAction        DisplayString,
      PolicyAlias         DisplayString,
      PolicyQos           DisplayString
  }
```

Figure 3. Example of policy table entry

We can configure a VPN between two sites using IPSec, or remote users can use their Internet Service Provider (ISP) to access the corporate Internet via PPTP [19]. The MIB for a VPN has configuration parts for IPSec and PPTP. For PPTP configuration, we require start IP, end IP, or enable PPTP or not. To create a site-to-site VPN, we must create a Security Association (SA) that defines the encryption algorithms and other information that the two sites use to establish the connection. The VPN MIB using IPSec must have the stated data.

For monitoring security and traffic on the network, it is necessary to manage logs for traffic, security, and event. The MIB for monitoring has configuration parts for logging behavior and tables for each log type. We referred to the proposed standard firewall monitoring MIB [5] and the Cisco monitoring MIB [8] explained in the Section 2.2 in defining our monitoring MIB.

To periodically monitor the firewall hardware status, the MIB for system status contains the following information: fan speed, CPU temperature, log capacity, up time, and so on. Trap types for system status and attack trial from unauthorized users must also be defined. Besides these basic MIBs for firewall configuration, a more detailed MIB can be defined for the specific firewall.

IV. REQUIREMENTS

In this section, we discuss the requirements that must be considered for the development of a firewall global management system.

4.1. MANAGEMENT OF FIREWALLS

In order to implement an FGM, some functional requirements must be considered. The FGM requires the following management functionality: managing each firewall's

configuration, and monitoring each firewall's status and logging, and managing FGM configuration and administration.

The basic role of FGM is managing multiple firewall devices from a central location. The FGM must support multiple firewall device view/control panels. FGM administrators can access firewalls and modify firewall configurations. The FGM must support a backup/restore mechanism for firewall configuration per each firewall device, to use other firewall configurations or to backup the current firewall configuration.

Each firewall monitors security on networks through several types of network logs: security, event, and traffic. Therefore, the FGM must perform a log analysis of each firewall. Further, the FGM must periodically monitor the status of firewall hardware, such as CPU temperature, fan speed, log capacity, up time, and so on. This will determine the current firewall hardware status and notify an administrator if the firewall does not respond, or if a problem exists with the value of status parameters.

The FGM must be able to manage lots of (hundreds or thousands) firewalls. Therefore, the FGM must support the view of a multi-level hierarchical directory structure of devices or device groups and manage firewalls according to the directory structure.

There are many administrators for managing multiple firewalls. The FGM also needs to support multiple administrator accounts. The FGM keeps information on each administrator and assigns firewalls to each administrator. Next, administrators can access the assigned firewall device/device groups. For supporting multiple administrators logging, FGM needs to support simultaneous multiple login sessions and a notification mechanism for firewall device information updates.

4.2. SECURITY

Security is an important concern in network management in different applications, especially those that involve equipment configuration or administration. Moreover, a firewall is a security device. Therefore, the firewall management process must guarantee security.

It is necessary to limit the access to FGM to a specific set of users. Simple authentication and access-control mechanisms are the preferred method to provide primary security. In existing management systems for network devices, such as router, switch, and cache engine have only a simple authentication mechanism. But the FGM must provide more secure access-control mechanism. Further, secure communication is required between management elements, such as between the management client and the management server, or between

the management server and firewalls.

4.3. FLEXIBILITY

As Internet usage becomes more widespread and the scale of organization increases, firewall usage will become more various and firewall products more diverse. The management environment of firewalls must support this diversity. For supporting various environments, the architecture of FGM must be flexible. The FGM must be divided into management components for flexibility. Communication between management components must be considered. As a result, the FGM is applicable to any firewall devices and any management environments of firewall spread.

V. DESIGN

In this section, we present our FGM architecture and communication protocol according to the following requirements: management functionality, security and flexibility.

5.1. FGM OVERALL ARCHITECTURE

We have designed our FGM based on a 3-tier architecture. The FGM consists of three management elements: the management client, the management server, and the managed device. Figure 4 shows the overall architecture of our FGM.

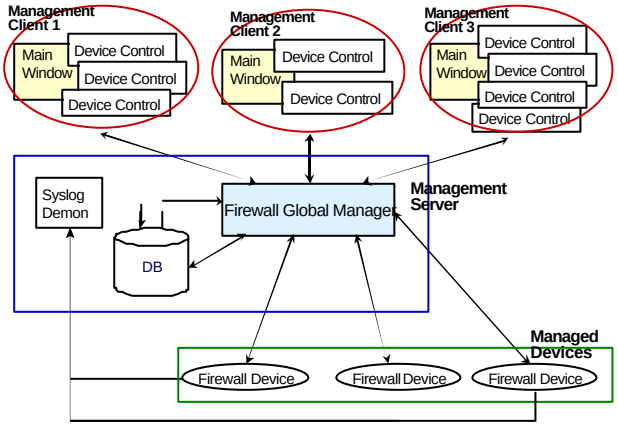

Figure 4. FGM Overall Architecture

The management clients have a main window to view all managed devices and multiple device control widows to view the management information for each device. The management server largely consists of three parts: a global manager performing the main

management function, a syslog daemon gathering log for log analysis, a DB storing whole management information. The managed devices are firewalls equipped with a management agent.

We select that the management user interface is a Web interface. The Web server receives a request from the management clients and sends the request to an FGM. Each firewall is equipped with an SNMP [4] agent. Our FGM is based on the same 3-tier Web-Based Management (WBM) [20] architecture and paradigm. But communication protocols between management elements are more secure than those of the existing Web-based network management systems.

5.2. MANAGEMENT COMMUNICATION PROTOCOLS

Basically, in Web-Based Management (WBM) [20], the communication protocol between management clients and the Web server is HTTP [11], and between the management server and managed device is SNMP. Yet, we use the HTTPS protocol between management clients and the Web server, and IPSec between the FGM and firewalls for security. Therefore, the SNMP protocol runs over the IPSec protocol in our FGM architecture. As mentioned in Section 2, the HTTPS [15], SSL [9, 14], and IPSec [16] mechanisms help secure communication. Figure 5 shows the communication protocol between management elements: management Web clients, the FGM, and firewalls.

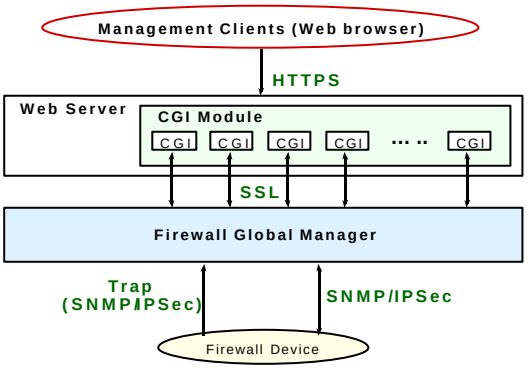

Figure 5. FGM Communication Protocols

Generally, the Web server and an FGM run on the same machine, but in some cases, the multiple FGMs use a single Web server. Thus, it is necessary to communicate between the Web server and the FGM. For flexibility of the FGM, we consider communication between the Web server and the FGM. The SSL connection is selected for security reasons. The

components of the management server, such as the Web server, global manager, syslog daemon, and a DB can be located in a different machine. Naturally, these components are different software modules. We will explain the more detailed architecture of FGM considering the flexibility in the following section.

Many SNMP implementations and network architectures do not support secure communications. Without a secure connection, the firewall configuration could be exposed. Therefore, we must secure communication between the FGM and firewalls. We considered SNMPv3 [21] for security reasons, but the firewalls already have an IPSec module for the VPN setting. If we use this IPSec module for security communication below SNMP, the resource overhead for executing SNMPv3 is removed.

Moreover, according to the report of Hia, H. Erik [22, 23], the SNMPv2c-over-IPSec solution and the SNMPv3 impose similar amounts of computational overhead on network devices. The SNMPv3 solution consumes as much as 24% more network capacity than the SNMPv2c-over-IPSec solution. Conversely, consumes as little as 11% less mean processing time. As a result, we chose SNMP over IPSec.

5.3. FGM DETAILED ARCHITECTURE

The main functions of the FGM were described in Section 4.1. The FGM consists of the following components: a request handler, an FGM administration manager, an FGM configuration manager, a device node tree manager, a device manager, etc. Figure 6 shows the components of FGM and the relationships between the components..

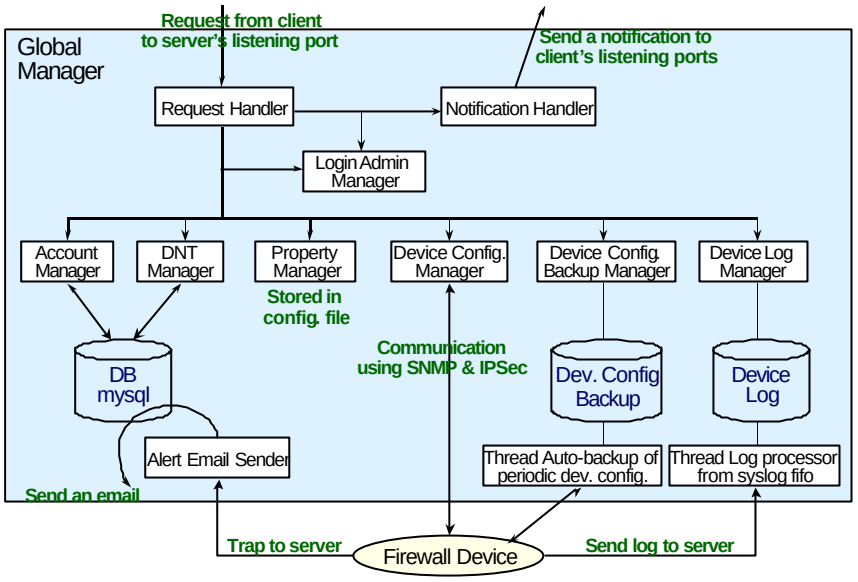

Figure 6. Detailed Architecture of FGM

The request handler receives management requests, processes the requests and calls the other components according to the request. The account manager manages administrator account information, such as ID, password, the login status, etc. The Device Node Tree (DNT) manager handles the multi-level device tree directory architecture. The information on administrators and DNT is stored in DB tables. The property manager handles the configuration parameters of the FGM, such as automatic logout timeout, log directory capacity, log directory location, and so on. The parameters are stored in the configuration file. The login admin manager manages the administrators presently logged in the FGM.

The device configuration manager manages the firewall device configuration, such as system information, policy definition, VPN setting, via SNMP/IPSec protocol. The auto-backup thread of periodic device configuration will backup the device configuration automatically during a fixed period at the device backup directory in the FGM configuration file. The device configuration backup manager manages the auto or manual backup configuration setting of each firewall. The log processor thread stores log data from the syslog First-In-First- Out (FIFO) queue into a device log directory. Next, the device log manager analyzes the log data based on the administrator's demand.

The alert email sender receives a trap notification from firewalls and sends an alert email to the administrators listed in the DB table. The notification manager notifies the DNT hierarchical structure update or device information update to the other administrators currently logged in for the synchronization of management data.

VI. IMPLEMENTATION

We have implemented an FGM based on the FGM design presented in the previous section, and have applied the FGM to the management of commercial firewall devices. The name of the firewall is Broadband Internet Gateway (BIG) [24]. We have developed FGM on a later version of Linux 2.2.12 OS, JDK 1.2.2 [25] and ucd-snmp 4.1.2 [26] using g++/gcc compiler.

6.1. FEATURES OF OUR FGM

Our FGM provides a Web-based management user interface. The FGM supports all the management functionalities mentioned in the Section 4.1. The FGM supports multiple administrator accounts: one super administrator, and multiple regular administrators. The super administrator is allowed to operate FGM without restrictions. Further, the FGM supports simultaneous multiple login sessions. The FGM can manage a multi-level hierarchical directory structure of firewall devices/device groups. That is, a device group can have devices as well as device groups. We can import/export the FGM configuration data in ASCII format for purposes of backup or reuse.

Through the FGM, we can view and control multiple firewall devices from the same location. The FGM notifies BIG device information update and Device Node Tree (DNT) hierarchy structure update to the other administrators currently logged in. The FGM supports a backup/restore mechanism for BIG device configuration per each BIG device.

We can monitor the hardware system status, such as fan speed, CPU temperature, log capacity, up time, only if a firewall device control/monitoring panel is up. The FGM supports BIG device remote logging through a syslog demon, then analyzes log data from each BIG device. For example, we analyze traffic data, gain bandwidth utilization by hosts or protocol and the top 20 IP addresses with the most packets being transmitted through the BIG device and the top 20 protocols appear in the log, such as HTTP, FTP, TELNET, etc.

Figure 7 shows the whole features of FGM such as the architecture, administration, device grouping, and communication protocol. As shown in Figure 7, the FGM supports secure communication between the management elements.

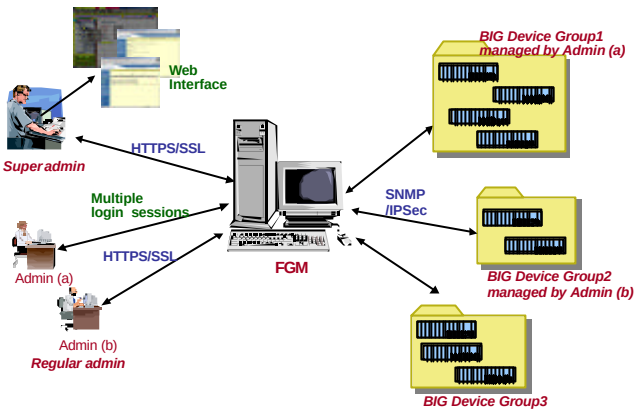

Figure 7. Features of FGM

Figure 8 shows an example of a FGM management user interface. The left part of the figure shows the DNT panel, the hierarchical structure of firewalls. On the right is the main window with the management information. The right upper part shows the menu: FGM configuration, import/export, help, logout. The current management data is the information of a firewall device, such as device name, device type, IP address, administrator ID, email addresses to send alert mail.

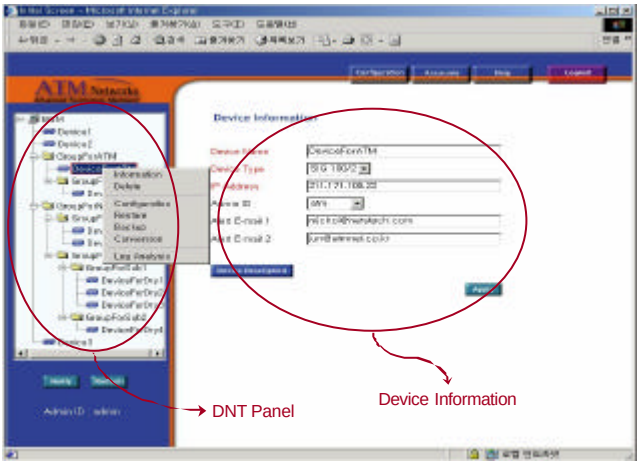

Figure 8. FGM Management Interface

Figure 9 shows the main BIG management user interface. If we double click each device icon on the DNT panel in Figure 9, a new window for the BIG device management appears. The left part of the Figure 9 shows the management menu: system information, administration, policy, VPN setting, caching, monitoring, and so on, the upper part is showing the current BIG system status: fan speed, CPU temperature, log capacity, and up

time. This monitoring data is periodically updated.

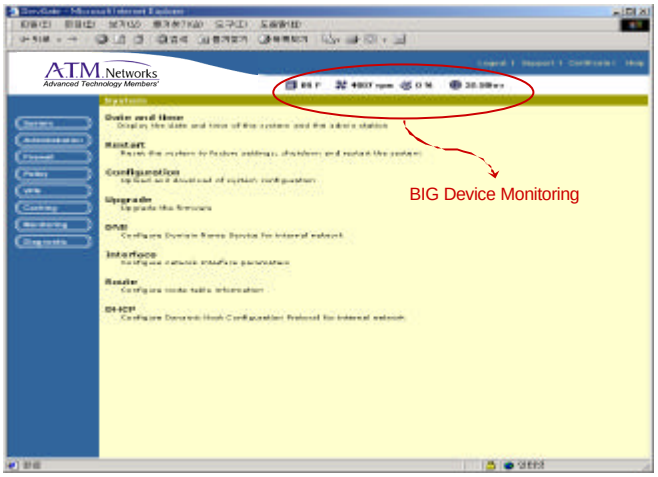

Figure 9. BIG Management Interface

VII. Comparison With Firewall Management Systems

We briefly compare firewall management products by focusing on their features. The firewall products are very diverse and widespread, yet, firewall management systems are few. The known firewall management products are WatchGuard's LiveSecurity Control Center [27], SonicWall's Global Management System [28], Netscreen's Netscreen Global Manager [29], and so on. This table compares the features of commercially available firewall management products and our FGM. Blank columns represent features not supported, or we could not find appropriate information on them from the available literature.

Table 1. Comparison our FGM with other firewall management products

Company & products	Admin.	Mgmt. of topology	Log reporting & Analysis	Multiple login	Import/ Export	OS supported	Communication	UI
WatchGuard's LiveSecurity Control Center	Multiple admins.	Yes	Event, traffic log			Windows 95/98/NT4.0	Encrypted network session (RSA 40-bit RC4 encryption)	Web interface
SonicWall's Global Management System	Single admin.	Yes	Event, Security log	Single user login	Yes	NT4.0/ Windows2000/ Solaris	DES/IPSec tunnel	Web interface
NetScreen's Netscreen Global Manager	Multiple admins.	Yes	Event log	Single user login	Yes	NT4.0/ Windows2000/ Solaris	Proprietary encoding	Microsoft windows interface
OpenService's SystemWatch	Multiple admins.		Event log	Multiple user login session		NT, Solaris	3DES encrypted connection/ SNMP trap support	Web interface

Cisco's PIX Firewall Manager	Multiple admins.	Yes	Event log			NT	Encryption/ SNMP trap support	Web interface (Java applet)
POSTECH's Firewall Global Manager	Multiple admins.	Yes	Security, event, traffic log	Multiple user login session	Yes	Linux, Solaris	HTTPS, SSL, IPSec, SNMP	Web interface

Most firewall management systems support multiple administrations and multiple simultaneous login sessions. However, Netscreen's Netscreen Global Manager [29] supports only one administrator logging in at a time, and manages the internal firewall devices. Basically, the firewall management systems in this table are managing for the internal firewall devices. Therefore, they use their own proprietary communication protocol. Our FGM uses an open management protocol, SNMP, for managing various firewall devices. The OpenService's SystemWatch [30] has its security agent, which resides on network security devices, such as firewalls, intrusion detection systems (IDS) [32], and virtual private networks (VPN) systems. The SystemWatch security agent plays a central role to monitor and manage security devices by keeping security administrators aware of meaningful activities. The goal of SystemWatch is the same as our FGM for managing multiple security devices. However SystemWatch does not use not an SNMP agent but a proprietary security agent for monitoring. The security agent monitors, manages and responds to critical system events on the security devices based on the pre-configured policies. The management system is based on a 2-tier architecture; that is, a management system does not exist and the Web-interface is a management console. The Cisco's PIX Firewall Manager [31] can manage up to only 10 PIX firewall devices. The PIX Firewall Manager supports a specific configuration of SNMP trap and poll activities.

Our FGM can manage thousands of firewall devices and supports not only firewall monitoring, but also firewall configuration globally.

VIII. CONCLUSION AND FUTURE WORK

Organizations are connected through networks and the Internet is widely used in organizations. The Internet is a public network, so has security risks. The firewalls are generally used for protecting internal networks. For managing various firewalls from a central location, a flexible and secure management server and standard management protocol are also necessary. The SNMP is the standard management protocol, but there is insufficient MIB to manage the firewall configuration. The SNMP also has a security defect.

Our FGM is based on the same 3-tier Web-based network management architecture and paradigm. But the existing firewall MIB [5] is insufficient to configure firewall configuration. Therefore, we have extended the firewall MIB for firewall configuration and we propose our firewall MIB as a Internet draft. We also presented our design and implementation of an FGM based on the proposed architecture considering security and flexibility.

We plan to modify the firewall MIB more generally and apply our SNMP agent to other firewalls and manage these firewalls through our FGM.

REFERENCES

[1] Steve Steinke, "Firewalls," Network Magazine, CommWeb, June 2000. http://www.networkmagazine.com/article/NMG20000613S0010/2
[2] D.Brent Chapman, Elizabeth D.Zwicky, Building Internet Firewalls, O' Reilly, May 1996.
[3] William R. Cheswick, Steven M. Bellovin, Firewalls and Internet Security, Addison-Wesley, April 1994.
[4] J. Case, M. Fedor, M. Schoffstall and C. Davin, "The Simple Network Management Protocol (SNMP)," RFC 1157, May 1990.
[5] C. Grall, "Firewall Management Information Base," Internet-Draft, April 1998.
[6] Hughes, "IP Security, Creating Secure Intranets over the Internet," Proc. of INET' 96, Montreal, Canada, Spring 1996.
[7] B. Gleeson, A. Lin, J. Heinanen, G. Armitage, "A Framework for IP Based Virtual Private Networks," RFC 2764, February 2000.
[8] Jim Fitzgerald, "CISCO-FIREWALL-MIB," Cisco Systems Inc., December 1999.
[9] Alan O. Freier, Philip Karlton, "The SSL Protocol Version 3.0," Internet Draft, IETF Transport Layer Security WG, November 1996.
[10] W. Richard Stevens, TCP/IP Illustrated, Volume 1, Addison-Wesley, 1994.
[11] W3C, "Hypertext Transfer Protocol-HTTP/1.1," Internet Draft draft-ietf-http-v11-spec-rev-06, HTTP Working Group, Nov. 18 1999.
[12] W. Yeong, T. Howes, "Lightweight Directory Access Protocol," RFC 1777, March 1995.
[13] M. Crispin, "INTERNET MESSAGE ACCESS PROTOCOL - VERSION 4rev1, " Internet Draft, January 2001.
[14] OpenSSL, http://www.openssl.org.
[15] Apache-SSL, http://www.apache-ssl.org.
[16] S. Kent, R. Atkinson, "Security Architecture for the Internet Protocol," RFC 2401, November 1998.
[17] D. Harkins, D. Carrel, "The Internet Key Exchange (IKE)," RFC 2409, November 1998.
[18] Cheriton, Greenwald, Singhal, and Stone, "Designing an Academic Firewall: Policy, Practice, and Experiences with SURF," Internet Society Symposium on Network and Distributed System Security, San Diego, CA, February 1996.
[19] K. Hamzeh, G. S. Pall, W. Verthein, J. Taarud, "Point to Point Tunneling Protocol (PPTP)," Internet Draft, June, 1996.
[20] J. W. Hong, et. al., "Web-based Intranet Services and Network Management," IEEE Communications Magazine, Vol. 35, No. 10, Oct. 1997, pp. 100-110.
[21] Blumenthal, Wijnen, "User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)," RFC2274, January 1998.

[22] Hia, H. Erik, "Examining How Secured SNMP Network Management Consumes Network Resources," August 2000, http://filebox.vt.edu/users/hhia/SNMP-IPSec-Tests.htm.
[23] Hia, H. Erik, "Deploying Secure SNMP in Low Data Rate Networks," July 2000.
[24] BoradBand Internet Gateway (BIG), A.T.M. Networks, http://www.atmnetworks.co.kr.
[25] SUN, "JAVATM 2 SDK, Standard Edition," http://java.sun.com/products/jdk/1.2.
[26] UCD-SNMP, http://net-snmp.sourceforge.net.
[27] WatchGuard, LiveSecurity Control Center, http://www.watchguard.com.
[28] SonicWall, Global Management System, http://www.sonicwall.com.
[29] Netscreen, Netscreen Global Manager, http://www.netscreen.com.
[30] OpenService, SystemWatch, http://www.open.com.
[31] Cisco, PIX Firewall Manager, http://www.cisco.com.
[32] Mark Cooper, "An Overview of Intrusion Detection Systems," Xinetica White Paper, Xinetica Ltd., 2000.