Design and Implementation of a Global Manager System for Firewall Devices

Mi-Joung Choi, Hae-Young Lee, Ji-Young Kong, Young-mi Shin and J. Won-Ki Hong
Dept. of Computer Science and Engineering
POSTECH, Pohang Korea
{mjchoi, hae, konga, dry, jwkhong}@postech.ac.kr

Abstract

A firewall is a security device placed between an internal network and the Internet. It is designed to protect the internal network from unauthorized user access. Various firewalls are widely used in many organizations. Due to the spread of various firewalls, a secure and flexible Firewall Global Manager (FGM) is necessary. For flexibility, the global manager must be divided into components and needs an open management protocol, such as the Simple Network Management Protocol (SNMP). Yet the SNMP has a security defect and a Management Information Base (MIB) for firewalls is insufficient. Therefore, an MIB for firewalls should first be defined, and secure communications between the manager system and managed objects must be provided. This paper explores the topic of a global manager for secure and flexible firewall management. We define a firewall MIB and present the FGM requirements. We also present an FGM architecture and design that satisfy the requirements. We also explain our FGM implementation for commercial firewall management.

Keywords
Firewall, Firewall Global Manager, Security, SNMP, MIB, Web-based User Interface

1. Introduction

The Internet has made vast amounts of information available to computer users at home, in business, and in education. 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 companies that do business on the Internet are often concerned that the Internet could form a security risk. How can an organization prevent users who access their public Web site from accessing sensitive private network resources? Further, what of internal employees who wish to transmit highly sensitive data from the corporate intranet to the outside word? The popular and easiest way to solve these problems is to use firewalls [1, 2, 3].

A firewall is a security device placed between an internal network and the Internet. It is designed to prevent unauthorized users from gaining access to confidential corporate and customer data in the internal network. Using firewalls, organizations can protect internal networks from this security risk. For this reason, firewalls are generally used in many organizations. Therefore, a management system for firewalls from a central location is also necessary.

Today, the management systems of firewalls use proprietary protocols and manage only one's own 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, an open management protocol can be considered. Using Simple Network Management Protocol (SNMP) [4], the management system could manage diverse firewalls equipped with an SNMP agent. In general, the firewalls are not equipped with an SNMP agent for security reasons and a Management Information Base (MIB) for firewalls is not sufficient. Therefore, an MIB for firewalls and security communication consideration of SNMP is necessary.

The firewall devices are diverse and the diffusion environments of firewalls are also various. To apply the 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. Because of this reason, communications between the management components must be secure. Moreover, the firewall is a device guaranteeing security, and the management system for firewalls must be foolproof. For this, the security mechanism must be supported between the management components.

In this paper, we present a Web and SNMP-based global manager system of firewalls considering security and flexibility. We first propose an MIB definition for firewalls that can provide complete management functionalities. We then propose the architecture for a Firewall Global Manager (FGM) that can provide a secure and flexible management of firewalls. Finally, we present the design and implementation of an FGM, which is a firewall global manager that we have developed for Web-based firewalls management. Our work is significant by making a secure and flexible manager for firewalls suitable for Web and SNMP-based firewall management.

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 define the MIB for firewalls comparing standard firewall monitoring MIB. In Sections 4 and 5, we present the FGM requirements and design, respectively. In Section 6, we describe the implementation of our proposed FGM architecture. In Section 7, we summarize our work and discuss possible future work.

2. Related Work

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

2.1 Firewall

A firewall protects networked computers from hostile intrusion that could 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]

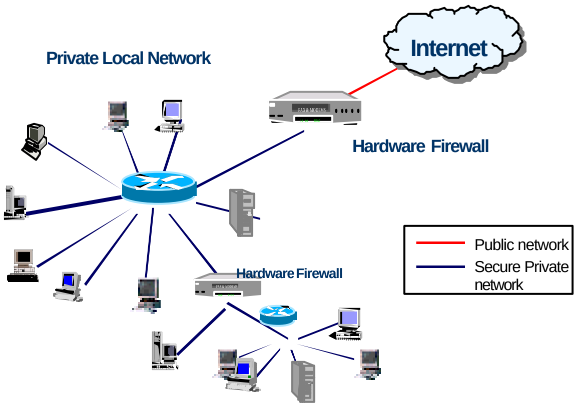

Figure 1. Hardware firewall

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

In addition to protecting your network from an external attack, a firewall can also be used to prevent the abuse of your Internet resource from within, by logging and controlling the use of the Internet by users on your LAN. By logging every connection between your LAN and the Internet, firewalls enable you to account for the usage of the Internet by your employees, making those individuals or departments responsible for their Internet usage. Any internal abuse of your Internet resource can be quickly identified and stopped, leaving you in complete control of your internal network operations.

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 resource, 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 the 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 scope 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 [7] is 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) [8] protocol runs above the Transmission Control Protocol/Internet Protocol (TCP/IP) [9] and below the higher-level protocols, such as HyperText Transport Protocol (HTTP) [10] , Lightweight Directory Access Protocol (LDAP) [11], or Internet Messaging Access Protocol (IMAP) [12]. 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.

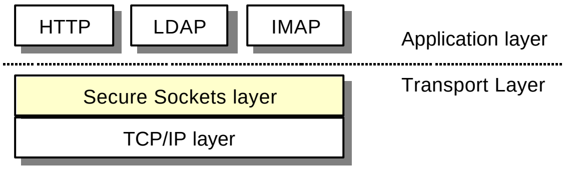

Figure 2. SSL between TCP/IP and high-level application protocols

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) 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) [13]. 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 [14] 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.

2.4 IPSec

The Internet Engineering Task Force's (IETF) IP Security (IPSec) [15] Working Group is developing standards for IP-layer security mechanisms for both IPv4 (the version use 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) [16], 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.

3. Private Firewall MIB

The standard firewall MIB is not meant to be used for reporting about the configuration of a firewall. Currently, most firewalls use unique and/or proprietary protocols and representations for dealing with the configuration and 'policy'. This MIB does not have many 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 is necessary [5].

The main functions of a firewall [17] are to control accesses based on management policies and to configure the Virtual Private Network (VPN) [18]. Therefore, for providing firewall management functionality, an MIB should include at least the following five parts: system information, policy definition, VPN settings, log monitoring data, system 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, the IP address and netmask of the three interfaces needs to be configured. The MIB for system information has this whole information.

To protect the internal network, security policies that define a set of rules for network traffic need to be created. For example, we can allow only 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 the destination address, service type, scheduling, logging or not, action, QoS, and so on. Figure 3 shows the entries of policy tables.

```
• 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

The schedule is defined as an MIB table, the entry of the table is the schedule name, start & end times (year, month, day, hour, minutes). The policySchedule of 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. While somewhat complex, administrators can manage different types of policies.

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, start IP, end IP, enable PPTP or not, are needed. To create site-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 a traffic, a security, an event log. The MIB of monitoring has configuration parts for logging behavior and tables for each log type. We referred to the standard firewall monitoring MIB [5] and the Cisco monitoring MIB [7] explained in the Section 2.2 in defining our monitoring MIB.

For periodically monitoring 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 MIB for firewall configuration, a more detailed MIB can be defined for the specific firewall.

4. Requirements

We discuss FGM requirements that we must consider during development in this Section.

4.1 Management of firewalls

In order to implement an FGM, there are functional requirements to be considered. The FGM requires following management functionality: managing each firewalls configuration, monitoring each firewall' s status and logging, 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 supports 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 is monitoring security on networks through several types of network logs: security, event, and traffic. Therefore, the FGM must perform a log analysis of each firewall. Also, the FGM needs to monitor periodically the status of firewall hardware such as cpu temperature, log capacity, up time, and so on, to 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 manages multiple firewalls. Therefore, the FGM needs to support a view of multi-level hierarchical directory structure of devices/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 has information of each administrator and assigns firewalls to each administrator. Next, administrators can access an assigned firewall device/device group. For supporting multiple administrators logging, FGM needs to support simultaneous multiple login sessions and notification mechanism on firewall device information update.

The FGM needs to support import/export methods of the FGM configuration data in file format for backup and reuse.

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 process of firewall management must guarantee security.

It is necessary to limit FGM access to a specific set of users. Simple authentication and access-control mechanisms are the preferred method to provide primary security. Further, the communication between management components, such as between management client and manager server or between manager server and firewalls, must be secure.

4.3 Flexibility

As the usage of the Internet becomes more widespread and the scale of organization increases, firewall usage becomes more various and the firewall products more diverse. The management environment of firewalls must support this variety.

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.

5. Design

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

5.1 FGM Overall Architecture

We design our FGM based on 3-tier architecture: management client, management server, managed devices. Figure 4 shows the overall architecture of our FGM.

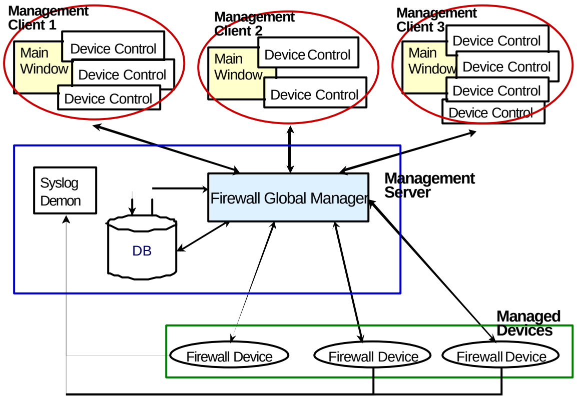

Figure 4. FGM Overall Architecture

The management clients have a main window viewed all managed devices and multiple device control widows viewed the management information of each device. The management server largely consists of three parts: a global manager performing the main management function, a syslog deamon gathering log for log analysis, a DB storing whole management information. The managed devices are firewalls equipped with a management agent.

5.2 Management Communication Protocols

We select that the management user interface is a Web interface. The Web server receives the request from the management clients and sends the requests to an FGM. Each firewall is equipped with an SNMP [4] agent.

Basically, communication between management clients and the Web server is HTTP [10], and the communication between the FGM and firewalls is SNMP. Yet we use 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. Figure 5 shows the communication protocol between management elements: management Web clients, the FGM, and firewalls.

As mentioned in Section 2, HTTPS [14], SSL [8], and IPSec [15] mechanism help secure communication. For secure communication between the management elements, we use these protocol mechanisms..

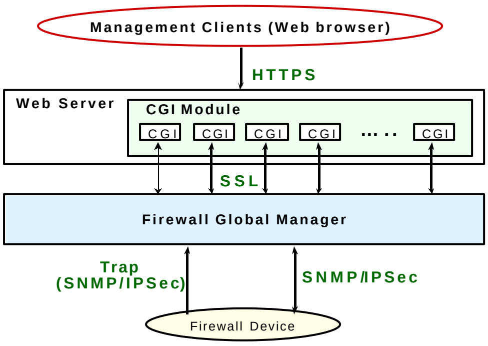

Figure 5. Communication Protocols

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 deamon, 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 Section 5.3.

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 had considered SNMPv3 [20] for security reasons. But the firewalls already have an IPSec module for VPN setting. If we use this IPSec module for security communication below SNMP, the resource overhead for executing SNMPv3 is removed. 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 components.

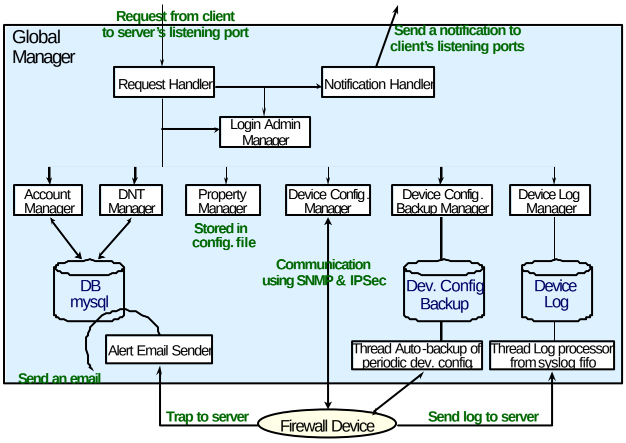

Figure 6. Detailed Architecture of FGM

The request handler receives management requests, processes the requests and calls the other components according to the request. The account manager manages the administrator account information, such as ID, password, the login status, etc. The Device Node Tree (DNT) manager manages the multi-level device tree directory architecture. The data of 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 administrators' demand.

The alert email sender receives a trap notification from firewalls and sends an alert e-mail 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 the management data.

6. Implementation

We have implemented an FGM based on the FGM design presented in the previous section, and have applied the FGM to management of a commercial firewall. The name of the firewall is Broadband Internet Gateway (BIG) [21]. We call this system BGM, which stands for BIG Global Manager. We have developed BGM on later version of Linux 2.2.12 OS, JDK 1.2.2 [22] and ucd-snmp 4.1.2 [23] using g++/gcc compiler.

6.1 Features of BGM

The BGM provides a Web-based management user interface. The BGM supports the whole management functionalities mentioned in Section 4.1.

The BGM supports multiple administrator accounts: one super administrator, and multiple regular administrators. The super administrator is allowed to operate BGM without restriction. Further, the BGM supports simultaneous multiple login sessions. The BGM 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 BGM configuration data in ASCII format for the purpose of backup or reuse.

Through the BGM, we can view and control multiple firewall devices from the same location. The BGM notifies BIG device information update and Device Node Tree (DNT) hierarchy structure update to the other administrators currently logged in. The BGM 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 BGM supports BIG device remote logging through a syslog demon, then analyzes the log data per each BIG device. For example, we analyze traffic data, gain the bandwidth utilization by hosts or protocol and the top 20 IP addresses who have 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 BGM such as the architecture, administration, device grouping, and communication protocol. As shown in Figure 7, the BGM supports secure communication between the management elements.

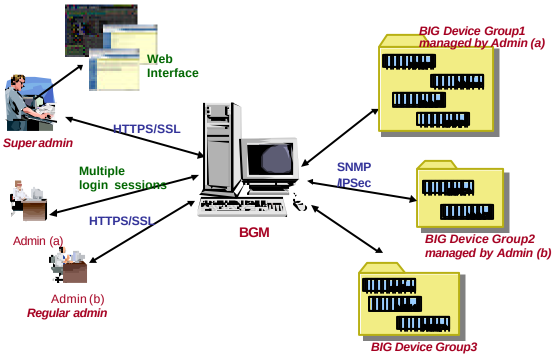

Figure 7. Features of BGM

Figure 8 shows an example of a BGM management user interface. The left part of the figure shows the DNT panel, the hierarchical structure of firewalls, and the right is the main window with the management information. The right upper part shows the menu: BGM 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.

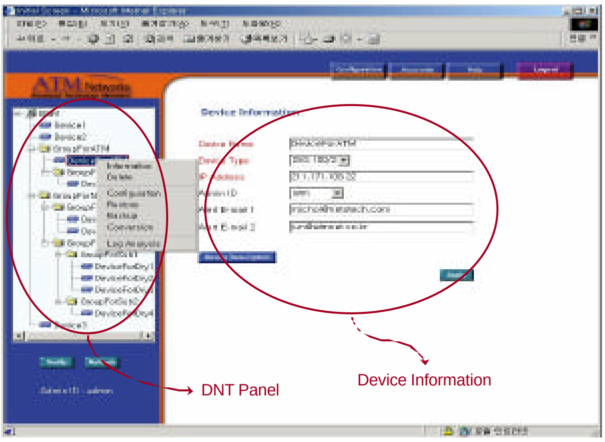

Figure 8. BGM Management Interface

Figure 9 shows the main BIG management user interface. If we double click each device icon on the DNT panel in figure 8, a new window for the BIG device management appears. The left part of the figure shows the management menu, the upper part is showing the current BIG system status. This monitoring data is periodically updated.

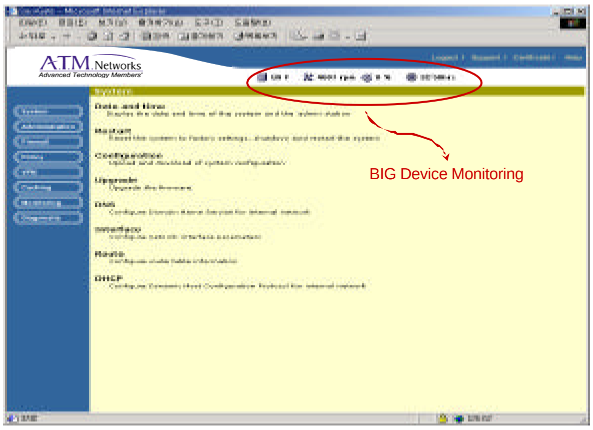

Figure 9. BIG Management Interface

7. 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.

In this paper, we defined the MIB for firewalls configuration. We also presented our design and implementation of an FGM (called BGM) 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] Jim Fitzgerald, "CISCO-FIREWALL-MIB," Cisco Systems Inc., December 1999.

[8] Alan O. Freier, Philip Karlton, "The SSL Protocol Version 3.0," Internet Draft, IETF Transport Layer Security WG, November 1996.

[9] W. Richard Stevens, *TCP/IP Illustrated, Volume 1*, Addison-Wesley, 1994.

[10] W3C, “Hypertext Transfer Protocol-HTTP/1.1,” Internet Draft draft-ietf-http-v11-spec-rev-06, HTTP Working Group, Nov. 18 1999.

[11] W. Yeong, T. Howes, “Lightweight Directory Access Protocol,” RFC 1777, March 1995.

[12] M. Crispin, “INTERNET MESSAGE ACCESS PROTOCOL - VERSION 4rev1, ” Internet Draft, January 2001.

[13] OpenSSL, http://www.openssl.org.

[14] Apache-SSL, http://www.apache-ssl.org.

[15] S. Kent, R. Atkinson, “Security Architecture for the Internet Protocol,” RFC 2401, November 1998.

[16] D. Harkins, D. Carrel, “The Internet Key Exchange (IKE),” RFC 2409, November 1998.

[17] 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.

[18] B. Gleeson, A. Lin, J. Heinanen, G. Armitage, “A Framework for IP Based Virtual Private Networks,” RFC 2764, February 2000.

[19] K. Hamzeh, G. S. Pall, W. Verthein, J. Taarud, “Point to Point Tunneling Protocol (PPTP),” Internet Draft, June, 1996.

[20] Blumenthal, Wijnen, “User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3),” RFC2274, January 1998.

[21] BoradBand Internet Gateway (BIG), A.T.M. Networks, http://www.atmnetworks.co.kr.

[22] UCD-SNMP, http://net-snmp.sourceforge.net.

[23] SUN, “JAVATM 2 SDK, Standard Edition,” http://java.sun.com/products/jdk/1.2.