# The Development of Policy Proxy Agent in Policy-Based Network Security Management System

GEONLYANG-KIM*, HYOCHAN-BANG*, JUNGCHAN-NA*, JONGSU-JANG*
*Information Security Research Division,
Electronics and Telecommunications Research Institute,
161, Gajeong-dong, Yuseong-gu, Daejeon City, 305-350,
REPUBLIC OF KOREA.

*Abstract:* - The security and management of network have become more important because users of network have increased and services of network have been diverse. Many security devices have been developed for securing a large network. But, attacks and damages for the network have not been stopping. So, a study for network security is needed continuously. This paper describes a technique relative to a policy based network security management system managing network security devices centrally. Policy Framework doesn't configure each device separately but centrally. So it is effective for managing large network. Policy Framework is easy to configure diverse systems, provides the method managing those transparently. But, if Policy Framework can't use legacy devices that are arranged and used at current network, that isn't useful. And if Policy Framework don't use legacy devices but only policy clients, much cost and effort will be consumed for installing and arranging policy clients.
This paper describes the method using legacy devices that already is arranged and used at network in Policy Framework. This method allows you to manage network security devices with small cost and effort centrally and efficiently.

*Key-Words:* Policy, PBNM, Policy Proxy Agent, Policy Client.

## 1 Introduction

As the Internet has activated and users of network have become increasing, the scale of network and incidents of network have become increasing. Network security devices have been being developed for securing the large network, and diverse network security solutions have appeared. Network security devices are such as follows, IDS detecting intrusions, IPS detecting and preventing intrusions, Firewall filtering harmful packets, ESM managing several security devices, and so on.
PBNM(Policy-Based Network Management) is a framework for managing several devices consisting of network centrally by using policy.
PBNSS(Policy-Based Network Security System) provides network security service with the concept of PBNM. The range managing systems by using the same policy in common is called the policy domain in PBNM, the policy domain consists of one policy server and several policy clients as shown in Figure 1. The policy server executes the function managing policy, deciding and distributing policy for policy clients, and communicating and exchanging policy with policy server of the other domain. The policy client executes the function enforcing policy that is received from the policy server. When PBNSS provides network security service, it is requested that PBNSS can control the legacy devices (e.g. Firewall, Router, IDS) that secure network in addition to policy server and policy client. We developed PPA(Policy Proxy Agent) for meeting that demands. PPA allows the legacy device act like the policy client without installing something in the legacy device or attaching something on the legacy device additionally.
PPA described in this paper processes the function translating a policy into several commands that the legacy device is acceptable, transferring commands to the legacy device, and transferring the policy application result to the policy server after receiving the result that the legacy device enforces commands received from policy client. PPA and legacy device function together as a policy client. PPA and legacy device are physically divided into two parts in most cases. Therefore if the error is occurred at either PPA or legacy device, the problem of policy inconsistence between PPA and legacy device can be broken out.

But, PPA executes the processing for policy consistence between legacy device and PPA.

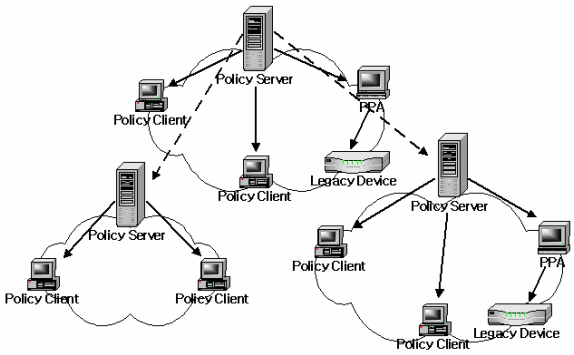


Fig 1. Policy-Based Network Management

When PPA translates policy into command, one policy becomes one and more commands even if the number of policy receiving from the policy server is one. If one of the commands is failed to apply to legacy device, PPA executes the rollback processing of the commands succeeded to apply for meeting atomicity of a policy.

The rest of this paper is organized as follows. Section 2 describes PBNM as the base concept of this paper. Section 3 describes the inner architecture of PPA and the communication of COPS protocol between a policy server and a legacy device, and the method processing policy consistence. Section 3 describes summary and conclusion of this paper.

# 2 PBNM

## 2.1 Policy Domain

The many systems consisting of network have different types such as vendor, version, and so on. So, they are configured and managed by other commands. The policy server of PBNSS manages many diverse systems by using same policy that is interoperable and has common format. The organization such as IETF and DMTF defined the common format such as PCIM(Policy Core Information Model), PCIMe (Policy Core Information Model extensions), and CIM(Common Information Model) for policy of PBNM. We defined NSPIM(Network Security Policy Information Model) by extending PCIM and PCIMe for PBNSS. So, policy server and policy client can process policy created and based on NSPIM. The policy is saved as the format of directory recommended at IETF and DMTF in Policy Repository, and it is accessed using LDAP (Lightweight Directory Access Protocol). And they recommends COPS(Common Open Policy Service) as the protocol transferring policy.

## 2.2 COPS Protocol

Figure 2 shows the messages of COPS protocol exchanging between a policy server and a policy client. At first, if a policy client is started, the policy client tries to connect to a policy server. When the policy client tries to connect to a policy server with Client-Open message, the policy server checks the client type of policy client. If policy server supports the client type, policy server transmits Client-Accept message with the value of Keep-Alive message to the policy client. Otherwise, policy server transmits Client-Close message with error information to the policy client. After policy client is connected with policy server, policy client transmits Request message with role and capability information of the policy client to policy server. Policy server decides policies for transmitting to policy client by referring the role and capability information of the policy client, transforms them into PIB, and transmits it to policy client by using Decision message. The policy client applies PIB, and notifies success or failure of policy application to policy server by using Report message.

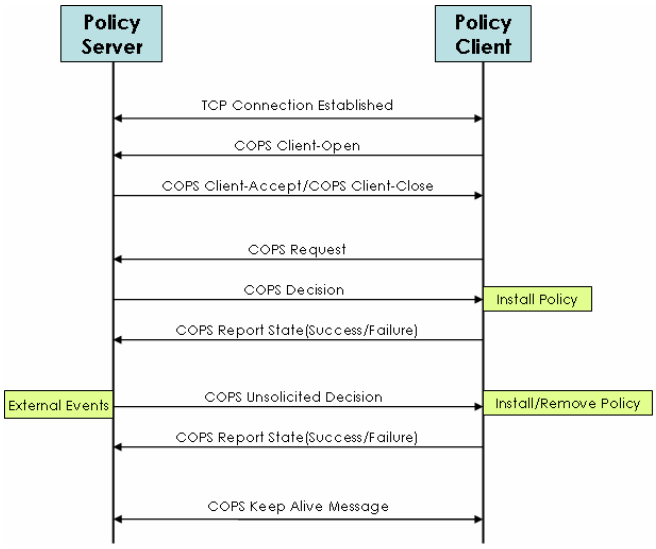


Fig. 2. The messages of COPS protocol

After policy client receives PIB from policy server, policy client keeps the connection with the policy server and exchanges Keep-Alive message periodically. If the request for applying a policy is received from the administrator some time later, policy server transforms the policy into PIB, transmits it to policy clients by using Decision message, and

checks success or failure of policy application through Report message received from policy clients.

## 3 Policy Proxy Agent

The Policy Proxy Agent(PPA) translates a policy transferred from policy server into several commands that legacy device is acceptable, applies them to legacy device, and notifies the result of policy application to policy server. PPA consists of several modules as shown in Figure 3.

COPSAgent communicates with a policy server by exchanging messages of COPS protocol. ProxyManager has interfaces with COPSAgent, and executes the function serving role and capability information, and processing policy data after decoding received PIB according to the request from COPSAgent. LPManager manages policy data transferred from policy server. TVManager manages time data of policy transferred from policy server. DBManager has interfaces with database, manages policy data, command data, resource information, and so on. PEPAgent translates policy received from policy server into several commands that legacy device is acceptable.

Messenger transfers translated commands to legacy device through an acceptable interface(e.g. Command Line Interface). RPManager installs policy to legacy device, or remove policy from legacy device according to time condition of policy.

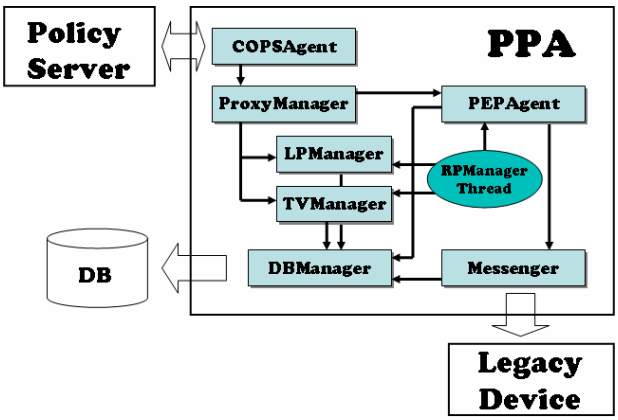


Fig. 3. PPA Architecture

The scenario of applying policy for legacy device is as shown in Figure 4. When PPA receives policy from the policy server through Decision message, PPA translates it into one or more commands, and transfers them to legacy device by using protocol that can communicate with it such as telnet. After PPA receives the result applying policy from legacy device, it transfers Report message with the policy application result to policy server.

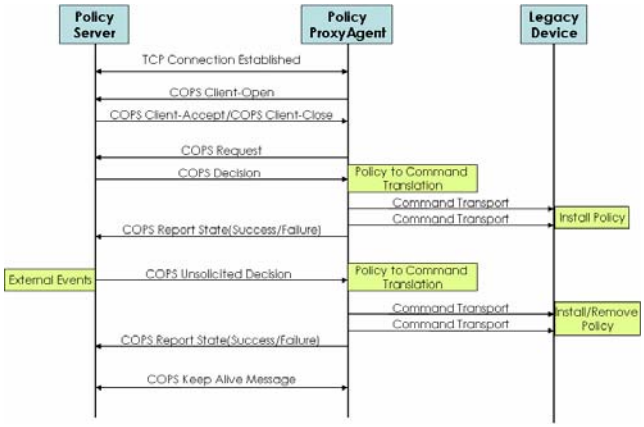


Fig. 4. The Policy Application Scenario for Legacy Device

When policy is translated to command, a policy can be transformed to several commands. Even if only one of them is failed, atomicity of the policy isn't satisfied. So, PPA provides rollback process as shown in Figure 5. The operation of policy requested from policy server is installing or removing. If PPA receives the request installing policy that consists of several commands from policy server, it installs them to legacy device one by one. If one of them is failed, PPA retries to install it. If the installing is failed again, PPA executes rollback process that removes the installed commands of the policy in legacy device. And if PPA receives the request removing a policy that consists of several commands from policy server, it removes them from legacy device one by one. If one of them is failed, PPA retries to remove it. Even if removing is failed again, PPA doesn't execute rollback process because it is needless policy, and reports the removal failure result to an administrator.

For a policy is applied to legacy device, the policy is transferred to legacy device through PPA. PPA translates the policy into several commands, and transfers them through an acceptable interface. Therefore, the policy has to be transferred to legacy device through PPA, and policy server can check the policy application result from only PPA. The policy application is completed with this procedure. So, PPA and the policy server can have the policy inconsistency problem at the exception situation such as the down of PPA, the down of legacy device, and the communication obstacle between PPA and legacy

device. The next section describes how PPA solves the policy inconsistency problem.

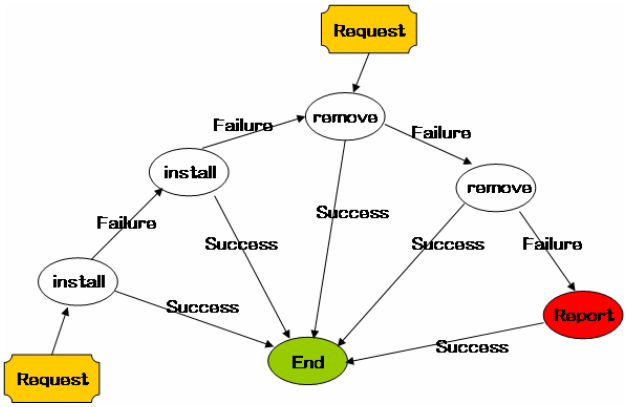


Fig. 5. The Rollback Procedure

When a policy translated into several commands is applied, even if only one of them is failed, PPA has to execute rollback process for keeping the policy consistency. PPA uses the Flag for keeping the policy consistency. The meaning of Flag value is as shown in Table 1.

Tab 1. The meaning of Flag value.

| Flag value | The meaning of Flag value |
|---|---|
| **0** | This command was installed in legacy device. |
| **1** | There was the request for installing of this command. |
| **2** | There was the request for removing of this command. |
| **-1** | Installing of this command was failed. |
| **-2** | Removing of this command was succeeded. |

PPA receives the request for policy installing or removing from policy server. When PPA receives the request for policy installing from policy server, PPA translates a policy into one and more commands, saves the commands and Flag value "1" meaning the request for installing to database, and installs the commands one by one to legacy device. If the installing of a command succeeds, PPA changes the Flag value of the command to "0" meaning that policy was installed to legacy device. PPA processes all commands by using this same procedure. If the installing of a command fails, PPA retries to install the command. If it fails again, PPA executes rollback process for the commands installed previously, and transfers Report message to policy server with failure information. If PPA fails the rollback process, it reports the result to an administrator.

When PPA receives the request for policy removal from policy server, PPA translates a policy into several commands that legacy device is acceptable, and tries to remove the commands from legacy device. If the removal of a command succeeds, PPA changes the Flag value of the command to "-2" meaning the success of policy removal. PPA processes all commands of the policy one by one by using this same procedure. If the removal of a command fails, PPA retries to remove the command. If it fails again, transfers Report message to policy server with failure information, and reports the result to an administrator.

When the exception situation such as the down of PPA, the down of a legacy device, and the communication obstacle between PPA and legacy device occurs, PPA has to reconnect to policy server with COPS protocol for providing the service with the policy consistency continually.

When PPA reconnects to policy server, it executes the policy consistency process by using the commands and the Flag value of commands in database. The policy consistency process is executed before connection of COPS protocol. PPA removes commands that the Flag value is "0", "1", and "2" from legacy device. The command that the flag value is "1" have to be removed because PPA may have installed the command to legacy device and may haven't changed the Flag value of the command to "0", and the command that the Flag value is "2" have to be removed because PPA may haven't removed the command from legacy device.

In conclusion, PPA keeps the policy consistency with legacy device and policy server through the policy consistency process with the commands and the Flag value of commands in database before connection of the COPS protocol.

## 4 Conclusion

This paper describes PPA that is designed for using legacy device in policy-based network security management system. Unlike policy client in policy framework, if PPA receives a policy from policy server, it translates a policy into one and more commands that legacy device is acceptable, transfers them to legacy device through protocol that legacy device is acceptable, and report the policy application result of legacy device to policy server. PPA has to consider the atomicity of policy because a policy is

translated into one and more commands. Therefore, the inconsistency problem of the policy can be occurred at the exceptional situation such as the down of PPA, the down of legacy device, and the communication obstacle between PPA and legacy device. This paper describes the method using legacy device in policy framework with the method solving the policy inconsistency problem between PPA and legacy device.

In conclusion, this paper describes PPA as the method using legacy devices that already is arranged and used at network in policy-based network security management system. PPA allows you to manage network security devices with small cost and effort centrally and efficiently in Policy Framework.

Finally, PPA has to consider the following occasions. The resource that the policy administrator wants to define is already being used in legacy device by the network administrator. So, when PPA intends to control the legacy device, PPA has to consider how many the resource of the legacy device can be used, and how to process the case that a policy has to be defined by using the resource that is already defined in the legacy device.

*References:*

[1] B. Moore, E. Ellesson, and J. Strassner, *"Policy Core Information Model – Version 1 Specification"*, RFC 3060, Feb. 2001.

[2] B. Moore, Ed. IBM, *"Policy Core Information Model (PCIM) Extensions"*, RFC 3460, Jan. 2003.

[3] Distributed Management Task Force, Inc., *"Common Information Model (CIM) Specification, version 2.2"*, Jun. 1999.

[4] D. Durham, J. Boyle, R. Cohen, S. Herzog, R. Rajan, A. Sastry, *"The COPS (Common Open Policy Service) Protocol"*, RFC2748

[5] K. Chan, J.Seligson, D. Durham, S. Gai, K. McCloghrie, S. Herzog, F. Reichmeyer, R. Yavatkar, A. Smith, *"COPS Usage for Policy Provisioning (COPS-PR)"*, RFC 3084

[6] J. Schnizlein, J. Strassner, M. Scherling, B. Quinn, S. Herzog, A. Huynh, M. Carlson, J. Perry, S. Waldbusser, *"Terminology for Policy-Based Management"*, November 2001, RFC3198

[7] R. Sahita, S. Hahn, K. Chan, K. McCloghrie, *"Framework Policy Information Base"*, RFC3318