COPS-IDR: a protocol for Intrusion Detection & Response

SEUNG-YONG YOON, GAE-IL AHN, KI-YOUNG KIM, JONG-SOO JANG
Electronics and Telecommunications Research Institute
161 Gajeong-dong, Yuseong-gu, Daejeon, 305-350
REPUBLIC OF KOREA
syyoon@etri.re.kr http://www.etri.re.kr

Abstract: The IETF Resource Allocation Protocol(RAP) WG has defined the COPS protocol as a scalable protocol that allow policy servers(PDPs) to communicate policy decisions to network devices(PEPs) in a Policy-Based Networking environment. So far most of the studies focused on QoS provisioning in this area.

Applying security policy, especially related to Intrusion Detections and Response, to Policy-Based Networking has been already discussed and developed. A lot of proposals are used existing SNMP or vender-specific methods to convey security policy information. But COPS is proposed for this situation, there is no definition of the extensions to the COPS protocol for security policies. In this paper a new client type for the COPS protocol is proposed to support security policies. The new client type is called "COPS-IDR"(COPS- Intrusion Detection and Response).

The proposed protocol has been implemented in a test-bed, where both the control plane and the data plane are realized according to the specification.

Key-Words: COPS, Intrusion Detection & Response, Policy-Based Networking, Network Security, Security Policies

1 Introduction

PBN(Policy-based networking) offers a solution to many of the pressing network management by offering a system-wide view of the network and its services, and shifting the emphasis of network management away from devices and interfaces to users and applications, abstracting the details of device configuration, and centralizing the creation and storage of network policies [1]. The representative application of PBN is a QoS(Quality of Service) provisioning in IP networks. Two architectural models for IP QoS, the Integrated Service(Intserv) and the Differentiated Service(Diffserv) architecture, have been proposed and the extensions to COPS to support for their QoS policy have been defined, called COPS-RSVP and COPS-PR [2][3].

Now, let us consider the security policies. As for security policies, especially related to Intrusion Detection and Response, the architecture and framework in a PBN environment have been already discussed and proposed[4]. Although COPS is a transport protocol for exchange of policy information between PDP and PEPs, there is no definition of the extensions to the COPS protocol for security policies. In this paper a new client type for the COPS protocol(COPS-IDR: COPS for Intrusion Detection and Response) is proposed to support security policies.

2 Security Policies & Alerts

2.1. Security Policies

The policy architecture by defined the IETF/DMTF is shown in Fig.1[5].

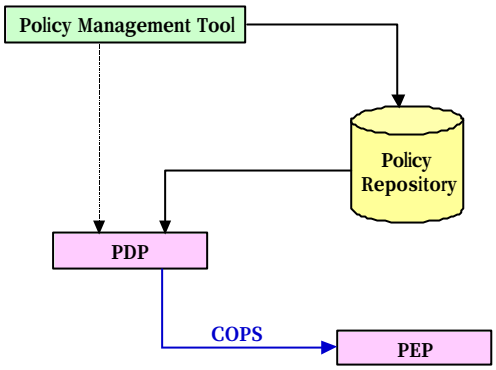

Fig. 1 The Policy Architecture

Administrator use the policy management tool to input the security policies and the policy repository is used to store the security policies generated by the management tool. The PDP is responsible for interpreting the security policies stored in the repository and communicating them to the PEP. The PEP applies and executes the security policies. The proposed COPS-IDR is used to convey the security policies between PDP and PEP. Herein IDRS(Intrusion Detection and Response System) acts as PEP and CPS(Centralized Policy Server) acts as PDP.

A Policy is a set of rules and instructions that determine the network's operation. Security Policies generated by the policy management tool and stored in policy repository classify the following categories:

(1) Detection Rules: A rule set of attack pattern(signature) for intrusion detection. When a new severe-impact attack is discovered, a new detection rule is created by administer and stored repository. At the same time, the PDP installs new security policy to PEP automatically and in real time. The PEP utilizes the latest detection rules without requiring reset or reboot, for uninterrupted attack protection.

(2) Alert Control Policies: Alerts are generated by PEP as intrusion detection reports. Alerts can be controlled by removing false positive alerts, reducing repetitive alerts, and aggregation of alerts.

(3) Response Policies: Cost model can be used to determine basic response. This cost model uses attack severity and detection certainty, along with administrator-specified thresholds to determine which response should be taken: take no response, log and alert, trace the attack, increase auditing, and block the attack.

(4) Filtering Policies: Filtering rules can be applied according to Access Control List or Black List. Response Policies are directly mapped and dynamically changed corresponding to the attack, while Filtering Policies based on long term statistics are static. So, these policies can be included Response Policies but herein classify separate category.

As for Alert Control Policies and Response Policies, the detailed specification can be found in [6]. COPS-IDR is used to convey above security policies.

2.2. Alerts - Detection Reports

IAP(Intrusion Alert Protocol) and IDXP(Intrusion Detection Exchange Protocol) have been already proposed for exchanging alert data between intrusion detection entities. Although there exist alert message transport protocols, we included alert into extent of message conveyable through COPS-IDR. To do so gives us several benefits: increasing efficiency and decreasing complexity. So the contents of COPS-IDR's payload contain not only security policies described in previous section but also alerts reported as execution result in

security policies. Alerts are not security policies, but closely related to them. After Detection Rules and Alert Control Policies install and execute in the PEP, alerts are reported to the PDP as a result of that. In order to send alerts from the PEP to the PDP, the Report State(RPT) message is used in COPS-IDR.

3 COPS-IDR

3.1. Provisioning Model for IDR

The COPS(Common Open Policy Service) is a simple query and response protocol that can be used to exchange policy information between a policy server(Policy Decision Point or PDP) and its clients(Policy Enforcement Points or PEPs)[7]. In order to be extensible, the COPS protocol has been designed to support multiple types of policy clients. Each client-type is described in a different usage document. The protocol employs a client/server model and uses TCP as its transport protocol for reliable exchange of message. Two main models are supported by the COPS protocol: Outsourcing model and Provision model[3]. Under the Outsourcing model, trigger events in the PEP must be handled with a policy decision. The PEP delegate this decision to the PDP with an explicit Request message. The PDP takes the policy decision and answer with a Decision message. RSVP client type used this model. Under the Provisioning(also known as Configuration) model, the PDP proactively sends Decision message to configure the resource handling mechanisms in the PEP. The mechanisms to exchange the configuration information and to store this information are based on the definition of a "Policy Information Base". This model makes no assumption of such direct 1:1 correlation between PEP events and PDP decisions. So Decisions are not necessarily mapped directly to requests, and are issued mostly when the PDP respond to external events. Outsourcing and Provisioning models are shown in Fig.2[8].

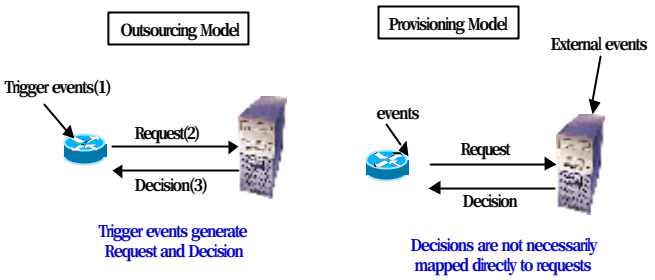

Fig. 2 Outsourcing and Provisioning Models in COPS

Let us now consider security policies for Intrusion detection and response. Detection Rules, Alert Control Policies, Response Policies, and Filtering Policies must be proactively provisioned to the PEP. The PDP installs Security Configuration decisions so that the client is able to detect and response Intrusion locally. Security Policies generated at the PDP download to the PEP infrequently and asynchronously. Because the provisioning model is very well suited such a situation, the proposed COPS-IDR client type operates under the Provisioning Model.

3.2. COPS-IDR protocol operations

In this section we describe an overview on the operation of the proposed COPS-IDR client-type.

3.2.1 Initialization

In order to initialize communication, the PEP must open the client session on the TCP connection with its PDP. First, a TCP connection is established between the client and server and the PEP sends a Client-Open message specifying a COPS-IDR Client-Type. This Client-Open message also contains system information

of the PEP and the PDP used it for client management.

If the PDP supports this specified Client-Type, the PDP responds with a Client-Accept(CAT) message. If the Client-Type is not supported, a Client-Close(CC) message is returned by the PDP to the PEP with appropriate reason for the close. After receiving the CAT message, the PEP can send requests to the server.

3.2.2 Common Operations

The PEP send "Security Configuration Request" to the PDP once the connection is established. This request message is discriminated among four categories of the security policies described in the section 2.1 according to M-Type field in the Context object. In response to request, the PDP downloads all security policies that currently relevant to requested category. The request is a demand not only to install suitable security configuration data to the PEP but also to send them from the PDP to the PEP asynchronously. If the security policies generated by PMT are needed to apply and execute to the PEP, they can be downloaded at any time asynchronously. This asynchronous data may be new policy data or an update to policy data sent previously. After the PDP received a REQ message from PEP, DEC message is returned to the client in response to the receipt of REQ message. DEC message contain the security policy data within the COPS Named Decision Data object and specify an "Install" Command-Code in the Decision Flags object. If there are no security policies, Command-Code in the Decision Flags object will be "NULL Decision".

As the PEP must specify a Client Handle in the request message, the PDP must process the Client Handle and copy it in the corresponding decision message.

The PDP add new security policy data or update/delete existing security configurations by sending subsequent unsolicited DEC message to the PEP.

If the previous security configurations change installed on the PEP, then the PDP update by simply re-installing the same instance of security configuration information again. And if the security policies are not more necessary, then the PDP delete by specifying a "Remove" Command-Code in the Decision Flags object.

The PEP must acknowledge a DEC message and specify what action was taken by sending a RPT message with a "Success" or "Failure" Report-Type object. When the PEP detected an intrusion, RPT message is used to transport detection reports, called alerts. This RPT message must include an "Accounting" Report-Type object and ClientSI object contained alert information.

3.3. Message Content

This section describes the basic message exchanged between a PEP and a remote PDP as well as their contents. These contents are contained client-specific data objects in each message.

3.3.1 Client-Open(OPN) PEP-> PDP

The Client-Open message is used to open COPS-IDR client-type session. The PEPID uniquely identifies the specific client to the PDP. A named ClientSI object included system information of PEP , that is OS, Network, H/W, S/W, Sensor, Analyzer, and so on.

The OPN message has the following format:

<Client-Open> ::= <Common Header>
<PEPID>
<ClientSI>
[<Integrity>]

3.3.2 Request(REQ) PEP-> PDP

The REQ message is sent by COPS-IDR clients to issue a Security configuration request to the PDP on TCP connection establishment. The R-Type field of context object will be Security configuration Request(0x10) and M-Type field of this object will be one of following security policies: Detection Rules, Alert Control Policies, Response Policies, Filtering Policies. ClientSI, the client specific information object, holds the security configuration specific data which a decision needs to be made. The REQ message is used to synchronize Request/Decision state shared between PEP and PDP. At this time, ClientSI object contain the security configuration state information installed in the PEP.

The REQ message has the following format:

<Request> ::= <Common Header>
<Client Handle>
<Context = security configuration >
<ClientSI: state data>
[<Integrity>]

Note that the COPS objects IN-Int, OUT-Int and LPDPDecisions are not included in a COPS-IDR REQ message.

3.3.3 Decision(DEC) PDP-> PEP

But the DEC message is sent from the PDP to a COPS-IDR client in response to the REQ message received from the PEP, the most of them is unsolicited message generated by external events - creation, update, and deletion of security policies by administrator - in the PDP.

The Client Handle and Context object will be the same as contained in the REQ message. The Decision: Flags object will contain "NULL Decision" or "Install" or "Remove" in the Command-Code field. And the Decision: ClientSI Data object will contain actual security configuration policies.

The DEC message has the following format:

< Decision Message> ::= <Common Header>
<Client Handle>
<Decision> | <Error>
[<Integrity>]

<Decision> ::= <Context>
<Decision: Flags>
<Decision: ClientSI Data>

3.3.4 Report State(RPT) PEP-> PDP

The Report State message is sent from the PEP to the PDP as a result to apply and execute security polices in the DEC message. In this case, the Report-Type field of Report-Type object will contain "Success" or "Failure".

Note that this message is used to communicate the alert information of intrusion detection. In this case, the Report-Type field of Report-Type object will contain "Accounting" and the ClientSI object will contain specific alert information.

The RPT message has the following format:

<Report State>::= <Common Header>
<Client Handle>
<Report-Type>
[<ClientSI>]
[<Integrity>]

3.4. COPS-IDR Scenario

In this section, we describe interaction between the PEP and the PDP through COPS-IDR protocol. Here is a transcript of a scenario in which the PEP acted as IDRS(Intrusion Detection and Response System) wishes to communicate security policies to PDP acted as CPS(centralized policy server). Such a possible sequence of scenario is depicted in Fig.3.

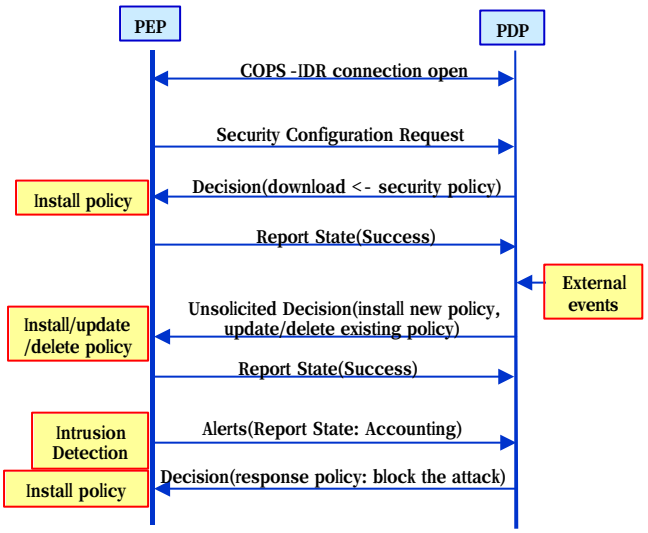

Fig. 3 A simple scenario

When an attack is detected in the PEP, alerts as detection reports are sent to PEP. In response to alerts, PDP download the response policies(e.g. block the attack) to install in the PEP as close to the attack source as possible. Because CPS(PDP) has already known the network domain managed by IDRS(PEP), that is possible. According to this scenario Ingress filtering can be realized through COPS-IDR. Fig.4 depict how it is accomplished.

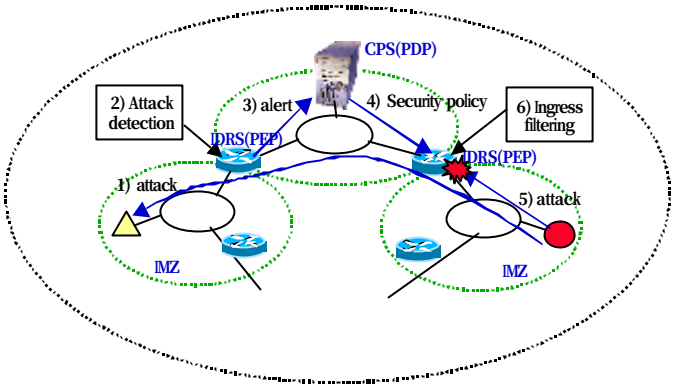

Fig. 4 Interaction between the PEP and PDP

4 Implementation

This section describes the prototype implementation realized within test-bed. The COPS-IDR protocol implemented with C language on Linux platform. The implementation consists of two main modules: COPS server and COPS client. Of course, it contains Intrusion detection and response modules at the COPS client side and policy management tool, policy repository, and alert/system manager at the COPS server side for complete operation. The modules that have been developed are shown in Fig.5. Intrusion detection and response modules implemented in one system physically.

Alert/System manager deals with alert information and client management information respectively. This information affects security policies.

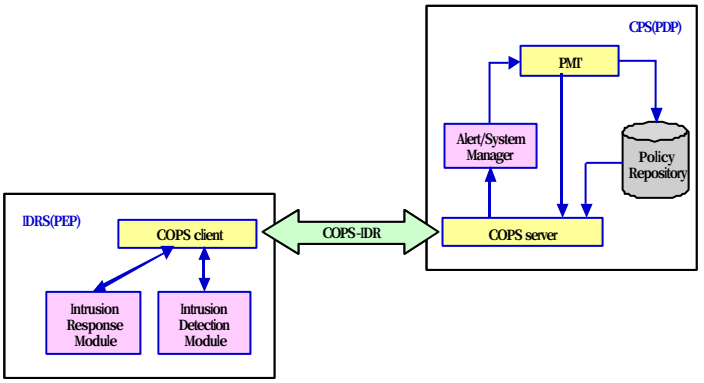

Fig. 5 COPS client/server modules

5 Conclusion and Future Work

In this paper, we proposed COPS-IDR protocol. COPS-IDR is a new client type for the COPS protocol is proposed to support security policies, especially related to Intrusion Detection and Response. The information contained COPS-IDR's message content are alerts as well as security policies. The proposed protocol is based on Provisioning model. We describe operations, each message content, and simple scenario in section 3.

Most of the existing IDRSs have a limitation on interoperability and prompt response capability. Policy-Based Intrusion and Response Architecture using COPS-IDR protocol solves this problem. As all of IDRS(PEP) update detection rules for newly discovered attacks at a time and response more quickly. The PDP collect and analysis alerts from all PEP overall and can apply new security policies considering whole network situation. As we can see in Fig. 4, the response can be also performed more efficiently.

So far we assume that the COPS-IDR is applied in single domain policy framework. When network grows larger and larger, managing all the devices in a single domain is impossible[10]. And a failure of the single centralized policy server may cause the failure of the whole network. To solve these problem, scalability and reliability, the studies to apply COPS-IDR under multi-domain policy based network architecture are required. But Inter domain issues are difficult to handle, constitute important topic to be covered in further studies.

Reference:

[1] Dave Kosiur, "Understanding policy-based networking", John Wiley & Sons, Inc., 2001

[2] J. Boyle, R. Cohen, D. Durham, S. Herzog, R. Rajan, A. Sastry, "COPS usage for RSVP", IETF RFC2749, January 2000

[3] 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)", IETF RFC3084, March 2001

[4] D. Schnackenberg, K. Djahandari, D. sterne, "Infrastructure for Intrusion Detection and Response", Preceedings of the DARPA Information Survivability Conference and Exposition, Hilton Head, SC, January 2000

[5] Dinesh C. Verma, "Policy-Based Networking: Architecture and Algorithms", New Riders Publishing, 2001

[6] D. schnackenberg, H. Holliday, R. Smith, K. Djahandari, and D. Sterne, "Cooperative Intrusion Traceback and Response Architecture(CITRA)", DISCEX'01, Anaheim, California, June 2001

[7] D. Durham, Ed., J. Boyle, R. Cohen, S. Herzog, R. Rajan, A. Stastry, "The COPS(Common Open Policy Service) Protocol", IETF RFC2748, January 2000

[8] S. Salsano, E. Sangregorio, M. Listanti, "COPS DRA: a protocol for dynamic Diffserv Resource Allocation", Planet-IP, Courmayeur 2002

[9] R. Mameli, S. Salsano "Usage of COPS for Intserv operations over Diffserv: Architectural issues, Protocol design and Test-bed implementation", ICC2001, Helsinki

[10] Yi Pan, Weilin Zeng, Yan Huang, "Multi-domain Policy Based Network Architecture", http://www.ics.uci.edu/~ypan/QoSppt/WebPPT/PS/multidomain/MDPS.ppt