Analysis and Extension of the IDWG Group Intrusion Detection Model

Paulo Fernando da Silva, Carlos Becker Westphall
Network and Management Laboratory, Post-Graduate Program in Computer Science Federal
University of Santa Catarina, Florianópolis – SC, Brazil
paulof.silva@ig.com.br, westphal@lrg.ufsc.br

Abstract
This paper proposes an intrusion detection environment model which allows for the interoperability among IDSs. The proposed model extends the intrusion detection architecture of the IDWG group, so as to support the alert response treatment. It uses the IDXP protocol and the IDMEF data model for the exchange of alert information. It was necessary to create new components in the architecture and to develop a new data model in order to support the responses. The proposed model has been implemented and validated through tests. The result is a response and alert management environment for different kinds of IDSs.

1. Introduction

Intrusion Detection Systems (IDSs) are tools used in the attempt to identify and track attacks to computer networks [4]. Not only the existing IDSs do not have a standard as for its implementation, but also the amount of modules may change from an IDS to another.

An important step of standardization of the IDSs is currently being developed by the IDWG group (Intrusion Detection Working Group) of IETF (Internet Engineering Task Force). The group’s objective is to define data formats and procedures for the sharing of information between IDSs. IDWG efforts have resulted in the specification of a format for message exchange, the IDMEF (Intrusion Detection Message Exchange Protocol). IDWG has also specified a protocol for the transportation of IDMEF messages, the IDXP (Intrusion Detection Exchange Protocol).

The IDS model proposed by the IDWG tries to define and to standardize the communication alerts only. This model does not define the communication or the architecture of the answers to the alerts.

The general objective of this paper is to propose an extension to the environment model of intrusion detection of the IDWG group. This extension must provide support to the sending of answers to alerts generated by IDSs.

The specific objectives are: to extend the IDSs architecture of the IDWG group, the data model IDMEF and the IDXP protocol, in a way that support the treatment of answers to the alerts; to develop a IDSs manager that uses the communication architecture presented by IDWG along with the developed extensions.

2. Related Papers

The IDMEF model is used in [1] for the development of a module to correlate alerts of different types of IDSs. In [3] new methods of alert grouping are proposed as well as possible future uses of the IDMEF model. One possible suggestion of future research for these projects is the possibility of sending answers through the same tool that correlates and groups the alerts, where the model proposed in this paper could be applied.

In [7] the IDMEF is used in mechanisms of contribution between agents of a IDS for mobile networks. In [8] the STAT model is presented. There, the IDMEF is used in the alerts generation and the communication between the modules because they can use different methods of intrusion detection. In the STAT reply modules can also be created, where the answers model developed in the present paper could be applied.

Several papers present research and implementations related to the capacity of IDSs answer to identified attacks. [2] presents the FIDRAN project, in which the author features specific mechanisms for execution of answers capable to reject packages and to execute helpful services. [9] presents an architecture for detection of intrusion in mobile networks, which provides reply mechanisms.
3. The Proposed Model

This chapter will present the IDREF intrusion detection environment model (Intrusion Detection Response Exchange Format), which will allow the IDSs to send answers to the detected alerts. IDREF Model is based on the works developed by IDWG related to the communication and interoperability between IDSs.

The implementation of a similar response model to the IDMEF alert model will bring advantages in the management of the IDSs, such as: response centered management, even with the use of IDSs from different manufacturers; categorization of answers from different IDSs in a single format (this way the operator does not need to know all the types of IDSs, it is enough to know the model); interoperability between the IDSs of the environment (the alert from one IDS can be answered by a component of another IDS); configuration of automatic answers in the manager for all analyzers connected to it, regardless of the kind of IDS.

Figure 1 presents the new architecture of IDS proposed by this paper. By comparing this architecture to the original architecture one can notice the inclusion of three new elements: Countermeasures, Action and Resource. Besides, there is the alteration in the Reply element, becoming now related to the Operator, the Manager and the new component of Countermeasures.

In this new architecture, when the operator receives a notification from the manager it has the option to send a reply in return to the manager. This reply informs the measures that must be implemented in the environment in order to contain an attack. When the manager receives a reply it codifies in accordance with the IDREF model and transmits it through the IDXP protocol to the countermeasures component, which interprets the reply and applies actions to resources of the environment.

A resource is any element of the environment that needs to suffer action so that an is controlled. Examples of resources can be accounts of user, router, firewalls, operational system processes or files.

![Figure 1. Architecture of IDS with support to the answers.](image)

An action is something that needs to be done in the environment so that an attack can be controlled. Examples of actions can be the blockage or closing of some resource, or the sending of a package through the network to contain the origin of the attack.

A reply always specifies one or more actions that must be applied to resources. When the component of countermeasures receives a reply, it must interpret the information contained in this reply and apply the actions in its resources.

The IDREF data model was developed based on the information that a manager gets when it receives a formatted alert inside the IDMEF model. This fact highlights the strong existing relationship between the two models.

Figure 2 presents the main class of the IDREF data model. The complete description of this model and the DTD that specifies the model for XML format can be found in [6].

The base class of IDREF model is the IDREF-Message class. In Figure 2 we can observe that this class has three derived classes: Response, React and Config. These three classes correspond to the three types of replies supported by IDREF model.

The first type of reply is represented by the Response class. This class contains information that must be sent to control or to inform about an attack, having three derived classes: TCP, ICMP and notify.

The second type of reply is represented by the React class. This class represents an environment reaction against the attack; in other words, some attitude will be taken in the environment to contain the attack. The React class has two aggregate classes: Block and Shutdown. The Block and Shutdown classes represent respectively the blockage and the closing of some resource. These two
classes have aggregated in themselves the Resource class, which represents a resource of the environment.

As examples of answers of the React type we have the blockade of an operational system file, the blockade of network equipment, the closing of a user session or the closing of an operational system process.

The third type of reply is sent through the Config class. This class represents an alteration in the configuration of some environment resource, having two aggregated classes: Command and Resource. The Command class contains a command to be executed by the resource that will be configured, and the Resource class represents the resource to be configured.

![Diagram of IDREF model classes](image)

**Figure. 2. Main classes of IDREF model.**

As examples of configuration answers we have the alteration of users or files permissions, the reconfiguration of firewalls or services and the activation of security auxiliary devices.

Due to the strong relationship between IDMEN and IDREF models, many of the existing information in IDREF model can have as origin information of the IDMEN alert received. For example, IDREF model will always allow the conversion of IDMEN information of origin or destination of the attack into a resource that will suffer some type of alteration so that the attack is contained.

### 4. Development and Validation of the Model

In order to make possible the validation of the proposed model the necessary components of the IDSs architecture for the formation of an environment with support to the sending of alerts were developed, as presented in Figure 1. The components developed for the validation of the model are:

- **IDSMan**: is a manager of alerts of IDSs which has the capacity to receive IDMEN messages and to send IDREF messages through an IDXP protocol;
- **IDSAAna**: makes the connection between the analyzer and the manager; it has the capacity to read IDMEN messages from a text file and to send them to the manager;
- **IDSRes**: is a component of countermeasures that has the capacity to receive IDREF messages, to apply them in resources and to store them in a log.

It was also necessary to develop a library that implemented the IDREF model, which was used by IDSMan and IDSRes. The IDREF library implements functions for generation and reading of IDREF messages in XML.

Four libraries were used in order to allow IDXP communication between the components. The Beeepcore-java and IDXP-java libraries are available at [http://sourceforge.net](http://sourceforge.net) and they implement the BEEP and IDXP protocols. The JavaIDMEF library is available at [http://www.silicondefense.com](http://www.silicondefense.com) and implements the IDMEN data model. And the Xerces2 library implements the document manipulation XML and it is available at [http://xml.apache.org/xerces2-x/](http://xml.apache.org/xerces2-x/).

The validation consists of executing an environment of real functioning, where an IDS will generate alerts in a text file in IDMEN format and the IDSAAna will send these alerts to the IDSMan. In the IDSMan the configuration of IDREF reply for the received alert and the sending of this reply to the IDSRes will be performed, which will apply in the environment the defined actions in IDREF reply, thus interrupting the attack.

The validation environment is formed by the IDS Snort, by a text file and the IDSAAna, IDSRes and IDSMan components. The IDS Snort was chosen both because it is an IDS frequently used and also because there is already a plugin that makes it capable of generating an alert in IDMEN format.

To make the Snort generate alerts in IDMEN format in a text file, the “Snort IDMEN XML plugin”, available at [http://www.silicondefense.com](http://www.silicondefense.com), was used. During the installation of the plugin other four libraries were necessary so that it could compile. These libraries are: LibIDMEF, LibXML2, LibNtp and LibIsc. Diverse problems of compilation and compatibilities between the libraries were also solved, until obtaining a new executable of the Snort with an installed plugin.

For validation test effect, a rule in the Snort was created. It causes alerts to be generated upon occurrence of any type of communication through ICMP protocol. To generate an alert the command “Ping 192.168.1.1” was executed in the local network. After the ping execution the Snort generated several alerts, which were captured by
the IDSA and sent to the manager by the previously established IDXP channel.

The manager, when receiving IDMEF messages from the IDSA, presents those to the operator in its alerts panel. With the initiated IDSRs, the operator can configure and send, via IDSMan, answers to received alerts.

In the example presented here the operator considers that host 192.168.1.1 is being attacked and configures in the IDSM a reply that specifies that this host must be blocked for 30 minutes. To achieve that, an IDREF reply of the React type is generated; it contains a class of the Block type with a resource of the Node type. The attributes of the Block class define a blockade of 30 minutes, and the information contained in the Node class specify the host that is being attacked.

When receiving the IDREF reply, the IDSRs applies it in the environment and records it in the log file. Through recorded XML text in the log file we verified that all the information that had been configured in IDREF reply from the manager arrived at the IDSRs component correctly and were also correctly applied in the IDS domain environment.

5. Conclusion

As verified in the research of correlated works, several works related with IDS deal with the sending of alert and its respective answers. With regard to alerts, the work of IDWG group joins them in a model in a way to allow the interoperability between the IDSs. However, with relation to the answers, we have no knowledge of a work that aims at joining them in a single model in order to allow a greater interoperability between different IDSs. So, the model proposed in this paper is presented as a new option to be incorporated in the development of intrusion detection systems.

By using the proposed model it is possible to manage alerts originated in IDSs of diverse types and manufacturers, sending answers to a component of countermeasures of a manufacturer about an alert generated by an IDS of another manufacturer, thus getting the total cooperation between the IDS.

The presentation of the functionalities contained in the developed components and its posterior validation demonstrate that the model specified and implemented in this paper acts as a mechanism of security for efficient and fast sending of answers to attacks. In such case, the present work contributed for the development of the security of computer networks, in the sense of providing an intrusion detection environment model that makes possible the interoperability, of both information of alerts and answers, between different types and manufacturers of IDSs.

As suggestions for future works we have: the analysis of IDREF model in comparison to other IDSs, as a way to propose improvements or new functionalities; the analysis and implementation of the other types of reply in the countermeasures component; and the development of a plugin structure in the countermeasures component.

References