A New Approach for IDS Composition

José Eduardo M. S. Brandão
Instituto de Pesquisa Econômica Aplicada - IPEA
Brasília, Brazil
eduardo@ipea.gov.br / jemsb@das.ufsc.br

Joni da Silva Fraga, Paulo Manoel Mafra
Departamento de Automação e Sistemas (DAS)
Universidade Federal de Santa Catarina (UFSC)
Florianópolis, Brazil
{fraga, mafra}@das.ufsc.br

Abstract—This paper presents a new approach for dynamic composition of Intrusion Detection Systems (IDSs). In this approach, the dynamic composition is used for building more flexible and adequate IDSs to open distributed systems. Intrusion detection systems are built on components or monolithic systems for creating more complex and cooperative intrusion detection systems that work as unified structures. Our approach can be applied in medium and large sized systems, for either closed or open networks. In order to provide the necessary scalability and flexibility, this approach is based on a service-oriented architecture (SOA). The IDSs are composed by using services founded on the XML language and Web Services technology. These technologies are being used with their security extensions on the data communications and the security events notifications. This paper presents the basic framework features, emphasizing the components search, selection, composition and communication.

Keywords - Intrusion detection; security; web services; XML.

I. INTRODUCTION

The security monitoring systems for virtual companies and large-scale networks demands a new approach. The systems that execute these tasks must cross the physical limits to create new logical limits for virtual organization. Hardly a unique IDS (Intrusion Detection System) model may take care of the diversity of systems and applications. To solve this problem, it is necessary to use interoperable wide scale IDSs and, if possible, a dynamic configuration to take care of temporary conditions.

In this paper we are proposing and describing a new approach for dynamic composition of intrusion detection systems. This model can be applied in medium and large sized systems, for either closed or open networks. These composed systems are created from the application of standards efforts to make possible the interoperability and communication among IDSs and its components.

The proposed model is also materialized by an infrastructure of services that allows the composition of intrusion detection systems. This infrastructure follows a Service Oriented Architecture (SOA) supplied by the Web Services technology [1]. The proposed model makes extensive use of XML (Extensible Markup Language) [2], Web Services technologies and its extensions for security [3][4][5][6].

The next section presents a brief summary of the related literature. The section III presents the proposed model for IDS composition and its requirements. The composition of IDSs in a service-oriented architecture is described in section IV. Section V presents the Registry and Search Service used in the services infrastructure that gives support to our composition model. In section VI we introduce the management model for components and compositions. The section VII presents the prototype, the implementation results and general considerations of the model. The conclusion of the work and future works are presented in section VIII.

II. RELATED WORKS

The computer security monitoring in widespread networks is a recent and an open issue research. The exchange of security information among organizations differs from that applied by the traditional intrusion detection systems, which keeps the information restricted to the scope of the organization where the data was collected. The Intrusion Detection Force (IDF) [7] is one of the few works in this area. It proposes mechanisms for security incident information exchange over the Internet.

The use of computational grid environment to collect and analyze intrusion detection data is a new approach. The GIDA [8] and GIDS [9] projects distribute the data analysis task through intrusion detection servers located at grid nodes. While the GIDA uses a predefined set of nodes to distribute the data to be analyzed, the GIDS use load balance. On this approach, our framework can be useful to standardize the discovery, access and communication among the IDS nodes.

The new proposals for distributed IDSs [7][8][9], in general, do not use standardized formats and protocols for communication among its components. However, the necessity of common formats is present in the literature. The Bass proposal [10] is an example of this. It transforms information from one format/structure to another format/structure without standardization concern.

Recent standardization efforts for security management are being developed, mainly, by the Internet Engineering Task Force (IETF). The IDWG¹ and INCH² groups are examples of this. The IDWG is specifying the Intrusion Detection Message Exchange Format (IDMEF)[11] and the Intrusion Detection Exchange Protocol (IDXP)[12]. These proposed standards can be used for information exchange among intrusion detection systems and components. The IDWG also defines a general intrusion detection model whose components are the basis of our framework. The INCH group is working on the exchange of incident information and statistics among involved parties [13].

¹ http://www.ietf.org/html.charters/idwg-charter.html
² http://www.ietf.org/html.charters/inch-charter.html

Paper sponsored by CNPq Project Nº 550114/2005-0
and Computer Security Incident Response Teams (CSIRTs). The requirements [13] and the Incident Object Description Exchange Format (IODEF) [14] data model and implementation are being specified. All of these specifications are based in XML [2]. The IDWG and INCH integration was foreseen in the IODEF model.

One of the most popular monolithic IDS, the SNORT\(^3\) can use one plug-in that allows it to send notifications in an old IDMEF format. The DOMINO [15] and the STAT family [16] extend the IDMEF format for their specific requirements. However, in these cases, the use of standards is limited and the standards extensions are not completely compatible with the original specification. In contrast to this, in our proposal, the architecture is based on original standards.

One work that is associated to ours is described in [9]. In that work, a Web Services based IDS was developed using the IDWG model [11]. Network sensors are distributed and generate alerts to a central server who correlates the information and generates new alerts. Unfortunately, the use of standards is restricted to the messages exchange. The model is centralized and it uses only its specific sensors. The Web Services architecture is used as part of the messages exchange mechanism.

The IDS that is closer to our proposal is the Prelude-ids\(^4\). The Prelude-ids is a hybrid IDS that aggregates and correlates alert messages generated by distributed sensors from diverse types and manufacturers. The messages are sent using adapted IDMEF messages over an SSL connection. It uses a hierarchical model of analysis, reporting the events to one or more central managers. Unfortunately, the control and the configuration of these components are manual. The Prelude-ids, as other IDSs, makes the data interchange through firewalls difficult to implement. The use of Web Services, as considered in our model, allows the transport standardization, facilitating the communication and the network control. The prelude-ids can be used in our model as a sensor and as an analyzer who integrates diverse components.

III. FRAMEWORK OBJECTIVES

The proposed model is based on five main objectives: the distributed intrusion detection; the use of heterogeneous elements in the intrusion detection; the dynamic composition of intrusion detection systems; the use of standards for interoperability and; the manipulations and interactions security of the distributed intrusion detection system. These objectives are described in sequence.

A. Distributed Intrusion Detection

Our model prioritizes the distribution of the intrusion detection elements. This elements must change information with others intruder detection elements, without the necessity of centralized elements. The distribution should not determine a lost of performance in the model, comparing with the most conventional IDS systems. However, the bottlenecks or unique points of fail, that are common in the centralization of functions or components of IDSs, will be prevented.

\(^3\) http://www.snort.org/
\(^4\) http://www.prelude-ids.org/

B. Heterogeneous Elements

Sometimes an IDS is not capable to collect, analyze and manage information coming from diverse computational levels and different networks. The considered model must allow the integration of any previously existing tools. These intrusion detection tools may receive and correlate data from other monitoring tools. Our model is based on a support integrator that does not define hierarch levels. The same support is used for the integration of IDSs or among components of an IDS. Components of IDSs could be, for example, tools for perimeter security, commercial detection systems (monolithic or distributed), systems based on agents (mobile or fixed), etc.

C. Dynamic Composition

The IDSs composition and the components integration can create a significant computational base, allowing a greater efficiency. It can also allow the possibility of adaptation in anomalies detection functions and the establishment of correlations among these anomalies. The composition of intrusion detection systems can be made dynamically, involving components sharing among IDSs. A composition can be permanent or temporary, defined with the purpose, for example, to collect data from sensors to search diverse events databases or to share information of an attack in progress. The components associations can still be extended for different organizations, allowing, for example, the security alerts sharing. This exchange of information can be subjects of politics that limit the communication flow among organizations. The model that we propose is a way to make these flexible compositions and the proper IDSs configuration control.

The Fig. 1 illustrates some possibilities of configuration supplied by our model from an infrastructure of services (support integrator). One IDS can be composed, for example, with independent sensors that supply events generation services and elements that execute analysis, reduction or correlation of these events. The IDS composition "B" shown in Fig. 1 is an example of this possibility. Sensors and analyzers, therefore, can be dependents or autonomous, assuming roles of requesters and services suppliers. The roles depend on the type of interaction among these components. Another aspect that we are exploring is the possibility of elementary components sharing among IDSs. In Fig. 1, the compositions "A" and "B" shares an independent analyzer. The "CE" composition congregates comple-

![Figure 1. Composition of Intrusion Detection Systems](image-url)
monolithic IDSs, supplying a more trustworthy intrusion detection service. Managers use sensors and analyzers and can also interact with other managers.

D. Standards for Interoperability

Our model of intrusion detection systems composition must take care of the necessities of both open and closed networks in medium and large-sized companies. The interoperability is important since we assume the use of monolithic intrusion detection systems and independent components supplied by different manufacturers. In these compositions, it is necessary that IDS elements “speak the same language”, through standardized formats of communication and integration. To make such systems possible, we propose the application of standards that allow the interoperability in these distributed compositions. In order to do this, we concentrate efforts to use the IETF standardization, the XML format and the Web Services technology of distributed service-oriented programming.

E. Security

The IDS security is obviously critical for any monitoring system. The use of security mechanisms is necessary to guarantee the security properties, mainly for the information exchange and manipulation in these distributed IDS compositions. The use of new security standards, associated with service-oriented architecture, is a way to achieve this.

IV. IDS COMPOSITION IN A SERVICE-ORIENTED ARCHITECTURE

The proposed model is developed based on the concepts of a service-oriented architecture (SOA) supplied by the Web Services technology [1] and its extensions. Therefore, the IDS composition components and the associated services of the support integrator (an infrastructure of services) are presented as Web Services. A component can be a service requester, a service provider or both. The Fig. 2 presents the necessary service stratification for IDS compositions.

Each element of a composition in the IDS service-oriented architecture is provided as a service. The referring services information, necessary for the services interactions, is made available through a Register and Search Service presented in our support infrastructure. This information is also important for the IDS composition. The description of an intrusion detection component in Web Services format must contain its service identification and the information of its location. The service access aspects and which policies govern it are also part of this information. Its interfaces are described in a machine-processable format, supplied by a service description language. In the Web Services technology, the WSDL (Web Services Description Language) [17] fills these requirements.

The composition services make the exchange of information among themselves through messages. These messages follow standardized formats and are carried by standards protocols, keeping security properties. Some IDS components in the service format make use of standards as IDMEF [11] and Syslog [18], codified in SOAP [19] messages. The IDS components can communicate with other components by two ways. The first one (Fig 3.a) is executed using the proposed framework and it is performed in SOAP [19] messages using the Web Service that controls it. The original message is formatted/translated to a standardized format, as the IDMEF, and transmitted to the Web Service endpoint that controls the other component. In the second communication option (Fig 3.b), the component is configured to send directly the messages to the other component, using a protocol supported by both components.

The IDS compositions management is made following the infrastructure proposed in Fig. 2. Through the Web Services Distributed Management (WSDM) [20], it is possible to manage any system service.

Security services are used in message communication, services authentication and access control. The XML security extensions (XML-Encryption [4][6] and XML-Signature [3][5]) are used to improve aspects of integrity and confidentiality for these communications. Only the registry and search security service will be treated in this paper.

V. REGISTRY AND SEARCH SERVICE

The Registry and Search Service of the proposed framework is based on the UDDI (Universal Description, Discovery and Integration specification) [21]. The UDDI uses the registry approach (SOA specification) with emphasis in the creation of domains for information storage. The UDDI defines mecha-
nisms that allow the association of several control servers, providing the necessary scalability to this service.

The UDDI uses XML data structures to registry business and services information, called "entities". The UDDI specification supplies a standard API to include, maintain and search these entities. Currently, the UDDI can registry six types of entities: businessEntity, businessService, bindingTemplate, tModel, publisherAssertion and subscription. BusinessEntity describes the business or organization that provide a collection of services. The businessService describes the collection of Web Services offered by the described organization in businessEntity. The BindingTemplate contains the technical information of one Web Service in particular. The tModel describes a "technical model" representing a reusable concept, such as a Web Service type, a protocol used by Web Services, or a category system. The publisherAssertion entity and subscription are used, mainly, for the information replication and for UDDIs associations.

A. Registry of Intrusion Detection Components

The IDS component must be registered and described in the UDDI to be offered as Web Service. The URL indicated in the accession tag, contained in one bindingTemplate structure, points to the Web Service access of this component.

In general, the localization and the implementation details are not important for Web Services. However, in the case of services related to the intrusion detection, such information can be indispensable. This occurs when it is necessary to locate a sensor in a specific network point. In order to do this, we use one bindingTemplate element. It includes the IDMEF Analyzer class and the AccessPoint of the original IP (or DNS) address of the component. One or more tModelInstanceInfo will indicate the native protocols and formats supported by the component.

The IDS component service interfaces are described in a WSDL document, whose URL is gotten from the UDDI registry. In our framework, for each type of intrusion detection component (sensors, analyzers and managers) there is a minimum WSDL API, in order to standardize the access and management operations.

B. Categories of Components

The categoryBag structure indicates the categories where the component can act. The general model of intrusion detection defined in [22], specifies three types of IDS components: sensors, analyzers and managers. In our work we also include the composition as a specific type of component. Other security components categories can also be included. The most recent and respected IDS classification models are in [23][24][25]. Joining these detection methods classifications and the IDS architecture characteristics, we produced a classification to be used for the analyzers categorization. For sensors taxonomy, we adopted the classification proposed in MAFTIA project [26], which details the origin of the data collected by the sensors. Those categories are represented by its specific tModels stored in the UDDI.

C. Registry of IDS Compositions

The IDS composition is a logical concept, not a service. The registry of the composed services represents it. A Web Service possessing a minimum IDS manager component API, represented in WSDL tModel “Composition API”, can manage it. To create a composition, the manager need to registry and maintains the composition information in the UDDI. The main differences between the composition registry and the IDS components registry are the composition representation details and the accessPoint. A specific XML structure was created to represent the composition. The URL in the accessPoint tag of the bindingTemplate points to a file instead of a Web Service. We call this file as “composition document”.

D. Discovering Components

A client (an administrator or autonomous system), who desires to locate one type of IDS component, can send its query to the UDDI. In order to do this, one find_service structure can be created using the component characteristics, including the categoryBag tags. Logical arguments can be used and the result can be ordered by using elements of the findQualifiers structure. The search function will return a serviceList with the description, the services identification keys (serviceKey) and the business key (businessKey) of the offered services. Having the serviceKey, the get_serviceDetail operation is sent to the UDDI. This operation will return the structure businessService with the necessary bindingTemplates to access the web service that manages the desired IDS component. Other ways of searching can be used, for example, localizing a component from its network address.

E. Discovering Compositions

There are two ways to locate the compositions. The first one is identical to the component discovery process. The registry is searched from its category. In the composition search, we use the keyValue="uddi:ids:composition" in the categoryBag. The second one is through any component of the composition, using the composition accessPoint at the component registry. This second mechanism permits to restore the composition structure if the composition document was lost, becoming the compositions location fault tolerant.

F. Security of the Registry

The registry operations and data exchange in the UDDI are always preceded by an authentication step. This authentication is part of the UDDI API and consists of sending a client credential and receiving one token that will be used from this moment in the UDDI transactions, while the session is still valid.

In our prototype, the messages are encapsulated in SOAP format and transported using the HTML protocol. The implementations of UDDI do not support the XML-Encryption directly. They must use the SSL to guarantee the data transport confidentiality. The optional dsig:Signature structure can be used in the UDDI to sign the service registry, preventing that adulterated registry points to malicious services.

In general, UDDI searches process do not demand authentication of users. An intruder can use the information collected in
the UDDI to identify, to locate and, consequently, to deceive the intrusion detection systems. Therefore, we also opt to restricting the access to the Web server who supports the UDDI. Using access control mechanisms native in the server, it is possible to control the user access.

VI. COMPONENTS AND COMPOSITION MANAGEMENT

The components management follows the OASIS Web Services Distributed Management (WSDM) specifications [20]. The intrusion detection components are resources managed by Web Services. The Management Using Web Services (MUWS) capabilities are applied on the components control. The Web Service that controls the component must only be compatible with the MUWS specification. Using the Management Of Web Services (MOWS), the “web service endpoint manageability capabilities” allows the monitoring of native Web Service IDS components and the Web Services that controls the IDS components.

The WSDM capabilities, the resources properties and the components APIs into the WSDL descriptions will be used to manage each IDS component. The management events differ from the security alerts exchanged among intrusion detection components. For management events we use the events services specified for the WSDM framework. These events will be sent from the Web Service components to the managers.

A specialized Web Service manager will exist to facilitate the composition management. The security operators and administrators using a browser can invoke this Web Service. This service will provide the graphical interface to the components relationships view and the necessary interface for the components and compositions registry into the UDDI. This Web Service will be registered like an IDS manager component. The register becomes independent of the manager service. A single composition can be managed by several manager services and a single service can be used to manage more than one composition.

VII. PROTOTYPE, RESULTS AND CONSIDERATIONS

In our work, we developed a prototype based on the Prelude-ids, adapted as Web Service. The sensors are based on the Snort IDS that also was adapted to work as Web Service. The sensors collect information and report them to the correlation in Prelude-ids, using IDMEF format. The composition was created initially at subnet “A” with a monolithic IDS based on the Snort (Snort WS1), which also acts as a sensor, sending alerts to the prelude-ids analyzer (PreludeWS1). The Manager component is responsible to make the composition registry and configuration.

A. Operational Tests

The tests on the prototype had been done at University Campus network, with components located in distinct subnets. An attack and possible responses to this was simulated. It was possible to activate new sensors, forming dynamically temporary compositions, as response of security incident alerts. Such compositions were used to collect detailed information. The tests obeyed the following sequence, illustrated in Fig. 4:

1. We have some intrusion detection components registered at the UDDI and distributed at campus network. The composition was created initially at subnet “A” with a monolithic IDS based on the Snort (Snort WS1), which also acts as a sensor, sending alerts to the prelude-ids analyzer (PreludeWS1). The Manager component is responsible to make the composition registry and configuration.

2. Under simulated attack, the component "Snort WS1" detected a possible attempt of attack proceeding from host "10.2.13.122", located at subnet "B".

3. The Manager queries the Registry and Search Service to locate new components that are available at subnet "B", using the netmask in search argument.

4. The component service Snort WS3 was located and activated as a new sensor to verify the traffic flow from the suspected host. New alert messages confirming the attempt of attack from suspect host are sent to analyzer.

5. The audit log system Loghost WS1 is activated in order to identify the users and network connections active during the attack at the suspect host.

In this simulation, a user in the console realized the attack. However, if the attack has its origin from other machines located at subnets under control (at the same security domain), more components could be activated to locate its origin and the possible damages.
B. Considerations

The tests demonstrate that the services-oriented architecture using our framework allows for a highly customizable combination of fast and simple IDSs at complex environments in virtual organizations. The use of IDSs dynamic compositions may be used to build adequate solutions for these wide scale systems. The use of Web Services permits that IDS components can be accessed across different or segmented networks.

The large use of standard formats, protocols and architectures, for the construction of the framework and the registry and research service, becomes the model interoperable in any environment. The use of the UDDI with a specific security components classification becomes the discovery of specialized services easier and more precise.

The use of computational grid-based IDS components [8][9] is possible. Using our infrastructure, these grid components can be registered and activated on-demand. This approach needs more investigation in future works.

VIII. CONCLUSION AND FUTURE WORKS

In this work we presented a model for dynamic composition of intrusion detection systems. This model allows the interoperability among intrusion detection elements and monolithic IDSs to create a unique large-scale intrusion detection system. To get flexibility and simplicity in these dynamic compositions we apply a service-based infrastructure, using Web Services technology.

This paper presented aspects of the necessary services stratification to support the IDS components services composition. We also presented details about the Registry and Search Service. This Registry and Search Service is based on the UDDI specifications. The composition of IDSs, from the available information into the UDDI, depends on the models and characteristics of the components that will be used. In order to do this, components classification were proposed.

In our work, we use consolidated and proposed standards. These standards include the formatting and transport of security alerts, the Web Services discovery standards and the Web Services management specifications. In our contribution, we demonstrate that several standards can be joined to produce an integrated framework for security management.

In future works, we will produce quantitative tests with the operational and computational costs of composed IDSs. We also want to detail the security mechanisms applied and risk analysis of our approach, including the interaction among different organizations.

REFERENCES