Log-based Distributed Intrusion Detection for Hybrid Networks

Francoise Sailhan, Julien Bourgeois

Log-based Distributed Intrusion Detection for Hybrid Networks

Francoise Sailhan
LIFC, University of Franche-Comté,
Centre de Développement Multimédia,
1 cours Leprince-Ringuet 25201 Montbéliard,
France.
sailhan@ieee.org

Julien Bourgeois
LIFC, University of Franche-Comté
Centre de Développement Multimédia
1 cours Leprince-Ringuet 25201 Montbéliard,
France.
julien.bourgeois@univ-fcomte.fr

ABSTRACT

We propose a novel hybrid distributed security operation center which collects logs that are generated by any application, service, and protocol regardless of the layer of the protocol stack and the device (e.g., router); providing a global view of the supervised system based on which complex and distributed intrusions can be detected. Our HDSOC further (i) distributes its capabilities and (ii) provides extensive coordination capabilities for guarantying that both the network and the HDSOC components do not constitute isolated entities largely unaware of each others.

Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: General—*Security and protection*; C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Wireless communications*; K.6.5 [**Management of Computing and Information**]: Security and Protection—*Invasive software, unauthorised access*

General Terms

Security

1. INTRODUCTION

Spurred by the emergence of Wifi technologies and the advance of generalised eCommerce and pervasive applications (e.g., rescue and military application), interest has moved towards the provision of a global solution that interconnects in a secure manner changing sets of clients, services and networks. This construction of Internet-scale applications introduces new challenges consisting in securing large-scale networks, including Wifi-enabled *ad hoc* networks, which are spanning geographically dispersed sites and distinct administrative domains. To this end, intrusion detection mechanisms should be coupled with preventive measures so as to prevent and identify unauthorised uses and abuses. Existing Distributed Intrusion Detection Systems (DIDSs) are extremely diverse in the mechanisms they employ to gather data and analyse it. However, DIDSs share in common the fact that they glean intrusion data by intercepting and analysing the network communications. Consequently, while a DIDS is in a very convenient position wherein it has a complete access to all the traffic traversing the network, its perspicacity suffers from the cost (in term of processing usage) associated with the in depth analysis of the intercepted traffic. In addition, the absence of information owned by the DIDS on the resources (hosts, services, protocols and applications) that constitute the network, renders the DIDS impotent to detect a wide range of (host, service, protocol and application-specific) intrusions. This inefficiency of actual DIDSs engaged us to propose a novel approach to intrusion detection, which is based on a Distributed Security Operation Center (DSOC)[2]. Rather than relying exclusively on a resource-consuming monitoring system, our DSOC collects logs that are generated by any application/service, layer of the protocol stack or device. More particularly, for each managed network (e.g., Local Area Network or LAN) one device is responsible for collecting logs while one another aims to detect intrusion. Then, based on the logs and intrusion-related information provided by each network, a global analyser detects the distributed intrusions in the overall network. Based on this preliminary work, we propose a novel Hybrid DSOC (HDSOC) which is dedicated to provide a fully distributed intrusion detection in a large-scale network including dispersed wifi-based *ad hoc* networks. Such network requires to support a global low-overhead and highly-distributed HDSOC that is adapted to the network topology and characteristics (e.g., its dynamics, organisation) as well as the medium of communication which may be e.g., unreliable. In order to increase the resilience of the HDSOC, we propose to (i) distribute its capabilities to the supervised nodes and (ii) dynamically reassign both correlation and intrusion detection functionalities as the topology evolves. The proposed security operation center further provides extensive log-parsing and event filtering capabilities, reducing the bandwidth usage while addressing conflicting, bogus, and overlapping data.

The remainder of this paper is organised as follows. We introduce the proposed Distributed Security Operation Center (§ 2). Then, we conclude this paper with a summary of our results (§ 3).

CSIIRW '08, May 12-14, Oak Ridge, Tennessee, USA.

2. DISTRIBUTED OPERATION CENTER FOR HYBRID NETWORKS

The main challenge in collecting logs in *ad hoc* networks stems from the need to minimise the generated traffic and the computational load on resource-constrained devices. This calls for parsing logs (i.e., extract only relevant data) rather than collecting raw logs that are characterised by a large size and prone to overload our system. Note that logs should also be parsed as close as possible from the device that generates it so as to diminish the number of long distant communications. In addition, resource-constrained devices that are not able to parse locally their logs, should delegate their parsing activity to a remote device which offers sufficient capabilities. To this end, we collect and parse logs locally whenever possible (§2.1), a lightweight collector and parsing agent (hereafter referred as Embedded Collection Box or simply ECBox) being embedded on the devices that show sufficient memory and computing resources. Alternatively, for resource-constrained nodes, we rely on a service discovery protocol [4], which discovers dynamically ECBoxes. Among potential candidates, the closest ECBox is selected so as to keep to a minimum long distance communications. The selected ECBox is further assigned the task of collecting and parsing logs on behalf of resource-constrained devices. After parsing and extracting relevant data from logs, (local/remote) ECBoxes generate event notifications that are further disseminated over the network, causing a slight increase of the traffic thank to the notification compact size. The dissemination of events is performed by an event notification service which aims to ensure that each device is delivered the information (i.e., events) relating to the distributed intrusion to which it participates. These event notifications give to each device a global view of the intrusion (i.e., intrusion state, intrusion development and its level of implication), helping it in reporting in early stages any intrusion furtherance (§2.2). In order to prevent each device from un-necessarily flooding all the network while providing to each device a global view of the intrusion attempts to which it takes part, we rely on a publish/subscribe distributed event notification service whereby consumers (e.g., devices taking part in the intrusion, security administrator's computer) express their demands to event producers (e.g., devices taking part in the intrusion) during a subscription process. This event systems faces the HDSOC requirements (namely scalability, autonomy and timeliness) by disseminating in a distributed way the events over a self-configured delivery structure organised as a cluster-based hierarchy (§ 2.3); such structure providing convenient aggregation and correlation points while rending our HDSOC less vulnerable to attacks due to its distributed nature.

2.1 Local Event Collection

Any resource (application/service, layer of the protocol stack and device) generate logs expressed in different formats including standard formats (e.g., syslog, MIB, HTML), proprietary or application-specific formats. These logs can be easily collected relying on standard Xmit protocols (e.g., SNMP, SMTP, HTTP) as it is the case in traditional monitoring architectures. The pervasiveness of these protocols ensures a significant level of interoperability to our HDSOC despite the hardware and software platform heterogeneity. We therefore consider a HDSOC in which each device generates logs whose collection is enabled thanks to a protocol agent which corresponds to a collection of clients implementing Xmit protocols. The collected information is then expressed in a format which is henceforth understandable by any HDSOC component. For this purpose, a dispatcher determines the source-type of incoming information and then forward it to the appropriate agent which formats it in a common format in the form of an event notification; an event notification being composed of a set of typed attributes, each attribute consisting of a name-value pair.

2.2 Distributed Intrusion detection

After translating collected logs into event notifications that are understandable by any DSOC component, events are correlated so as to avoid transmitting all the events across the network. The main objective of correlation lies in producing a succinct overview of security-related activities by filtering and aggregating events. Event filtering consists in eliminating events that are not relevant. These latter include 3 kinds of events : (i) events that match policy criteria (e.g., administrator login), (ii) duplicate events that do not provide additional information while consuming significant bandwidth, and finally (iii) events that are not critical to the supervised system, excluding events that relate to some vulnerabilities whose system is not exposed to. The events that went through the filtering process are further aggregated so as to provide a more concise view of what is happening in the system. This actual system view called *context* is stored locally with the previously generated contexts, before being transmitted over the network by the event notification service. Based on the collection of contexts it owns, a device intrusion detection. Intrusion detection consists in analysing a sequence of events so as to identify event sequence patterns characterising intrusion attempt. In practice, such intrusion detection consists in matching a sequence of events (a context) against a set of attack signatures. An attack signature characterises all the successive steps that are successfully completed by a conquering attack. In practice, an attack signature is defined by the security administrator as a labelled tree rooted by a node representing the goal, and intermediate nodes representing an attack step (i.e., an observable event) with a succession of children defining a way of achieving it. Such representation renders intrusion easily identifiable by matching attack signatures against a context, i.e., against a succession of (possibly distributed) events occurring on a specific set of systems (e.g. devices, collection of devices, network segments). Whenever an intrusion attempt is detected, an alarm (i.e., an event characterised by a critical level) is issued and transmitted. Central to the intrusion identification efficiency is therefore the context accuracy. This accuracy is maintained by the event notification service, which updates the context of each device with the most up-to-date events arising in the network.

2.3 Distributed Event Notification

The event notification service aims to deliver event notifications to devices so as to enable them to update their context, providing them a global view of the intrusions and hence rending intrusion detection more accurate. In order to prevent devices from blindly flooding the network whenever an intrusion is reported, our event model derives from the well known asynchronous publish/subscribe paradigm, in which *consumers* (e.g., devices taking part in the intrusion and se-

curity administrator's computer) express their demands to *event producers* (e.g., devices taking part in the intrusion) that in return transfer to these subscribers the description of any relevant event triggered locally. From a communication perspective, our distributed event notification consists in exchanging notifications and subscriptions/un-subscriptions between producers and subscribers through a collection of intermediate event agents. Note that a potential *event agent* (hereafter simply called agent) designates a device which holds our notification service. In practice, this collection of intermediate agents is organised into a cluster-based hierarchical structure wherein each cluster leader[1] maintains information and connectivity with its cluster members and its clusterhead. This underlying structure is then used for delivering event notifications to consumers. When delivering a notification, the main objective pursued by agents lies in forwarding that notification to an agent only if, toward this direction, there exists a consumer interested in receiving it. For the purpose of forwarding selectively notifications, each agent holds a subscription repository that includes each received subscription along with the respective neighbouring agent which forwarded it. Note that a neighbouring agent constitutes the potential candidate for forwarding notifications. This repository is used to define if there exists a consumer along the direction of the considered agent that subscribed for this notification. Based on this event notification, security information can be efficiently disseminated to the HDSOC.

Event notification Delivery Structure. In order to distribute the management of events, we based our event notification system on a distributed grouping communication which organise nodes into a self-organised delivery structure deployed of the hybrid network. This structure corresponds to a cluster-based hierarchy of n_L layers (n_L=$log(n_n)$, with n_n designating the number of nodes that are expected to join the event system), each layer being portioned into a set of bounded-size clusters controlled by a cluster head. The reason for setting bounds on the number of layers and on the cluster size is twofold. First, it ensures a control overhead ranging about $log(n_n)$ at each node. Second, the length of the path used for delivering notifications, and hence the related delay, is bounded ($o\ log(n_n)$). In practice, to warrant a loop-free structure, each node belongs to the lowest layer (L_0) and only the leader of cluster located in a layer L_i belongs to the upper layer L_{i+1}. This delivery structure is created and maintained by a grouping solution. Considering the fact that there exists no unique grouping protocol that is optimal for any kind of network, we rather use a specialised grouping solution for each type of network along with a mechanism for integrating them. We rely on the Nice protocol [1] which has been specifically designed for operating in infrastructure-based large-scale network (e.g., the Internet) and the Madeira protocol that comes from our previous research on network management [3] and is customised to infrastructure-less networks. The reason that motivates our choice for the Nice protocol, is twofold. First, this application-level protocol can operate over a large-scale network spanning different administrative domains. In addition, it was originally developed to support video streaming and hence meets the requirements driven by real-time delivery.

[1]The root of the delivery structure maintains information restricted to its cluster member.

3. CONCLUSION

In this paper, we propose a Hybrid Distributed Security Operation Center (HDSOC) which collects logs that are generated by any application/service, layer of the protocol stack or resource, providing a global view of the supervised system based on which it complex and distributed intrusions can be detected. Rather than directly transmitting these logs over the network, causing its overload, logs are parsed so as to easily extract intrusion-related information and distribute it by the mean of compact event notifications and alarms. This HDSOC couples a lightweight distributed intrusion detection component with a distributed event system and a distributed service discovery protocol for an efficient delegation of resource-consuming tasks and a bandwidth-saving cluster-based collection of the events across the hybrid network. Our Hybrid Distributed Security Operation Center further addresses the main commitments of hybrid networks: scalability and autonomy. More precisely, scalability comes from the distribution of the load resulting from log parsing or intrusion detection, to the supervised devices. In the meanwhile, autonomy is the consequence of a loosely-distributed event delivery that adapts dynamically to any permanent or transient network failure and a service discovery protocol that permits to discover dynamically an alternative to a faulty service.

4. ACKNOWLEDGEMENTS

Authors would like to acknowledge the implementation work carried by Renaud Bidou (University of Franche Comté), Epifanio Cuadrado-Salamanca(EIRC, LM Ericsson Ltd), Paddy Farrell (EIRC, LM Ericsson Ltd) and Abdoul Karim Ganame (University of Franche Comté).

5. REFERENCES

[1] S. Banerjee, B. Bhattacharje, and C. Kommareddy. Scalable application layer multicast. *ACM SIGCOMM*, 2002.

[2] A. Ganame, J. Bourgeois, R. Bidou, and F. Spies. A global security architecture for intrusion detection on computer networks. *ACM/IEEE PDPS*, 2007.

[3] F. Sailhan, L. Fallon, K. Quinn, and al. Wireless mesh network monitoring: Design, implementation and experiments. *IEEE DANMS workshop*, 2007.

[4] F. Sailhan and V. Issarny. Scalable service discovery for manets. In *IEEE PERCOM*, 2005.