A CONTRACT-BASED EVENT DRIVEN MODEL FOR COLLABORATIVE SECURITY IN FINANCIAL INFORMATION SYSTEMS

Roberto Baldoni, Giorgia Lodi
Department of Computer Science, University of Rome "La Sapienza", Rome, Italy

Gregory Chockler, Eliezer Dekel
Haifa IBM Research Lab, Haifa, Israel

Barry P. Mulcahy
WIT, Waterford, Ireland

Giuseppe Martufi
ElsagDatamat, Genova, Italy

Keywords: Financial information systems, Collaborative systems, Contracts, P2P systems, Complex event processing.

Abstract: This paper introduces a new collaboration abstraction, called *Semantic Room (SR)*, specifically targeted to facilitating sharing and processing large volumes of data produced and consumed in real time by a collection of networked participants. The model enables constructing flexible collaborative event-driven distributed systems with well-defined and contractually regulated properties and behavior. The contract determines the set of services provided by SR, the software and hardware resources required for its operation along with a collection of non-functional requirements, such as, data protection, isolation, trust, security, availability, fault-tolerance, and performance. We show how the SR model can be leveraged for creating trusted information processing systems for the sake of protecting financial institutions against coordinated security threats (e.g., stealthy scans, worm outbreaks, Distributed Denial of Service). To this end, we present several use-cases demonstrating a variety of the SR administration task flows, and briefly discuss possible ways of implementing the SR abstraction using the collaborative intrusion detection as an example.

1 INTRODUCTION

Context and Motivation. Financial institutions increasingly rely on Information and Communication Technology (ICT) systems and networks, which may also consist of publicly accessible communication mediums such as the Internet, in order to manage their internal processes and offer their services to financial actors, businesses and ordinary people worldwide.

Financial institutions have exploited these infrastructural changes in order to deliver innovative and high quality services, and increase their revenues, thus preserving efficiency and cost-effectiveness. The benefits have been enormous for both public institutions and private organizations. However, increased reliance on networked systems has exposed financial institutions and their infrastructure to a variety of security related risks, such as increasingly sophisticated cyber attacks aiming at capturing high value (or, otherwise, sensitive) information, or disrupting the service operation for various purposes.

Today, attacks are distributed in space by being coordinated on a large scale basis and originating from multiple geographically dispersed locations. They are also distributed in time often consisting of a preparation phase spanning over several days or weeks, and involving multiple preparatory steps aiming at identifying vulnerabilities and attack vectors

Baldoni R., Lodi G., Chockler G., Dekel E., P. Mulcahy B. and Martufi G. (2010).
A CONTRACT-BASED EVENT DRIVEN MODEL FOR COLLABORATIVE SECURITY IN FINANCIAL INFORMATION SYSTEMS.
In *Proceedings of the 12th International Conference on Enterprise Information Systems - Software Agents and Internet Computing*, pages 147-153
DOI: 10.5220/0002900001470153

(such as accidentally open ports).

For instance, on 25 January 2003 a memory-resident worm called Slammer began propagating itself from East Asia throughout the entire Internet (Moore et al., 2003a). In the U.S. and Canada approximately 13,000 Bank of America ATMs, which use the Internet for sending encrypted information, had to be shut down, due to their inability to correctly complete transactions.

Such security attacks result in both short and long term economic losses due to the lack of service availability and infrastructural resilience, and the decreased level of trust on behalf of the customers. Financial institutions are sensitive to such attacks and related damages. They are categorized as an operational risk in the first pillar of the Basel II accord (bas, 2009) saying that financial institutions that address the specific requirements developed for each risk category can potentially lower their risk capital requirements.

However, due to massive scale of the attacks, individual financial institutions would often lack the necessary infrastructure and resources to effectively handle the vast amounts of information that should be collected and analyzed in real time to enable effective protection. It is therefore, desirable to provide financial institutions with the necessary tools and abstractions to enable them to consolidate their physical resources and collaborate on sharing and processing information in a trusted and controllable fashion. In this paper, we introduce one such abstraction, called *Semantic Room (SR)*.

Each SR has an objective and is associated with a contract that specifies the set of services provided by that Semantic Room, rights and obligations of the semantic room members (including hardware and software requirements) along with the data protection, isolation, trust, security, availability, fault-tolerance, and performance requirements. Using this abstraction, contractually regulated cooperation environments can be created in a structured and controlled way.

We show the UML use case that describes the steps to be performed for creating and instantiating SR and we highlight the flexibility of the SR model discussing different instantiations of the specific SR that realizes a collaborative intrusion detection system.

Related Work. The need for collaborative systems for coping with current generation of threats and security attacks is highlighted in a number of works that can be found in the literature (e.g., (Krügel et al., 2001)(Xie et al., 2006)(Locasto et al., 2005)) and specific systems have been built. However, these systems do not address organizational issues, thus limiting their effectiveness as employed by federations of organizations, this is noted in (Moore et al., 2003b). The semantic room model includes both technical (e.g., complex event processing and data privacy) and organizational aspects (e.g., contract management). We expect the latter can foster security collaboration among financial institutions.

From the point of view of contract-based cooperation, the semantic rooms model is similar to the one proposed in the MEDUSA system introduced in (Balakrshnan and Stonebraker, 2004). MEDUSA is a distributed framework for managing the load in federated systems. It is based on pairwise contracts negotiated between on-line participants. Contracts set tightly bounded prices for migrating each unit of load between two participants and they specify the set of tasks that each is willing to execute on behalf of the other. The federated systems described in (Balakrshnan and Stonebraker, 2004) are similar to our SRs; however, the federation model regulated by contracts of (Balakrshnan and Stonebraker, 2004) is not used for complex event processing purposes as in our case.

The rest of this paper is organized as follows. Section 2 describes the Semantic Room abstraction, the roles of Semantic Room members and the contracts that are used to regulate Semantic Rooms. Section 3 describes the UML use case we have designed for SR creation, instantiation and management purposes. An example of a possible usage of an SR is also discussed. Finally Section 4 concludes the paper and outlines some future work.

2 THE SEMANTIC ROOM ABSTRACTION

A Semantic room is a federation of financial institutions formed for the sake of information processing and sharing. The partners participating in a specific SR are referred to as the *members* of the SR.

Each SR is associated with a contract that defines the set of processing and data sharing services provided by that SR along with the data protection, isolation, trust, security, dependability, and performance requirements. The contract also contains the hardware and software requirements a member has to provision in order to be admitted into the semantic room.

The SR abstraction embodies the *Event-Driven Architecture (EDA)* paradigm (Chandy, 2006), which applies a loosely coupled communication pattern among application components. EDAs typically consist of a sensing module which gathers data from various sources. Data is then correlated and analyzed in

order to determine whether appropriate actions have to be taken (responding building block) in a timely response to what has been sensed (Chandy, 2006) (e.g, sending alerts, invoking applications). The sensing module employs an event dissemination layer (e.g. publish/subscribe systems, group communication etc) which communicates events in an intelligent way from one entity to another. The collected sensor data is supplied to the processing module which might encapsulate various types of data processing services (such as on-line complex event processing, or long-running analytics and intelligence extraction).

In Figure 1 SR members provide raw data to the SR they are part of. The raw data are then processed in order to produce processed data. Raw data may include real-time data, inputs from human beings, stored data (e.g., historical data), queries, and other types of dynamic and/or static content.

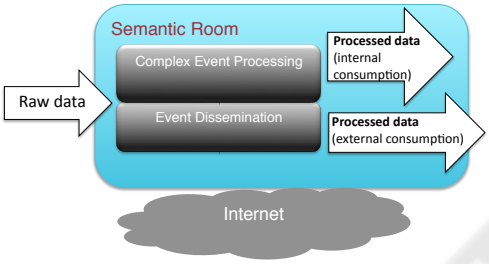

Figure 1: The Semantic Room model based on EDA.

Processed data can be used for internal consumption within the SR: in this case, derived events, models, profiles, blacklists, alerts and query results can be fed back into the SR so that the members can take advantage of the intelligence provided by the processing. SR members can use this data to properly instruct their local security management software in order to trigger timely reactions to threats, for example. In addition, a (possibly post-processed) subset of data can be offered for external consumption. The processed data can be rendered by means of Graphical User Interfaces (GUIs) or Dashboard applications.

Semantic Room members have full access to both the raw data that the members agreed to contribute to by contract, and the data being processed and thus output by the SR. The members can also be in charge of performing processing and dissemination.

In addition to the SR members, there can exist clients of the SR. These clients cannot contribute raw data directly to the SR; however they can be the consumers of the processed data the SR is willing to make available for external consumption (see Figure 1). One of the SR members is designated as the *owner* of that SR. The SR owner is in charge of the management and administration of that SR (note that the SR owner can delegate the administration to a third party). This might include the following:

- creating the SR;
- defining the contract that regulates the SR;
- registering the contract (so that the contract can be publicly available);
- terminating the contract (and thus disbanding the SR)
- managing the SR membership.

SRs can communicate with each other. In particular, processed data produced by an SR can be injected into another SR (e.g., through the SR client role above regulated by the contract). The interested readers can refer to (Lodi et al., 2010b) for further information about SRs communications.

2.1 Semantic Room Contract

An SR contract is used to define the set of rules for accessing and using data processed within the SR. There is a contract for each SR. Partners willing to participate in the SR must sign the contract. The contract content is defined during the SR creation phase. It includes four main parts: (1) details of the parties involved in the SR contract, (2) a collection of contractual clauses, (3) a collection of Service Level Specifications (SLS), and (4) the signatures of the involved parties. For the sake of brevity, in the following, we only describe the main components of the SR contract. (In the actual implementation, the SR contracts are specified in XML and are generated using an SLA definition and specification language named SLAng (Lamanna et al., 2003)).

The contract of an SR is a binding contract for the SR members. To this end, it includes:

1. Owner: unique identifier of the SR owner;
2. Members of the SR: list of SR members. This allows SRs to guarantee membership transparency. Nevertheless, this field can be optional or properly masked in order to accommodate cases in which complete SR anonymity has to be met;
3. Other business entities: unique identifiers of other business entities that can be involved in a specific SR (e.g., software provider, SR clients);
4. Signatures: signatures of the members, and other business entities if included.

The remaining content of the SR contract can vary depending on the services offered by the SR, and on the elements and requirements specified by the members. In particular, the contractual clauses included in the contract specify:

1. Name of the SR: a descriptive name in natural language (it represents the objective of the SR);
2. SRID: an identifier that uniquely identifies the SR;
3. Services: a description of the services offered by the SR and the rules for accessing the services (note that in case of an SR client, the SR can be paid by the client for the service offering. The rights and obligations that regulate this type of relationship can be included in this contract clause);
4. Penalties: the actions to be performed in case members, clients, or other business entities violate the contract.

The Service Level Specifications section of the contract includes all the technical requirements and the associated metrics. This section can be thought of as consisting of the following two main parts: (1) the requirements related to the quality of the information being processed and disseminated by the SR, and (2) the requirements related to the SR performance, as described below:

1. Data Format: the raw and processed data format required by the SR (e.g. attributes, types, etc.);
2. Data Sharing, Processing, and Storage restrictions within the SR and for Outgoing/incoming Data: the data exchange and usage must be protected against unauthorized accesses. Therefore, requirements for data exchange and usage can be defined as follows:
   - Information Security Rules: define the policies required for data encryption within the SR and for data exiting the SR (data disclosure policies can be included in these rules);
   - Anonymization rules: define the level of anonymity required for data that enter and exit the SR;
   - Processing Rules: define the rules for processing data exchanged in the SR or received from outside;
   - Dissemination Rules: define the rules for data exchange and sharing within and outside the SR, and across SRs;
   - Data storage rules: define the rules for recording data and logs within the SR.
3. Resource Sharing Relationships: specify the requirements for resource sharing with other SRs;
4. Minimum Requirements for Joining the SR: amount of resources and data that a joining partner must provide to be part of the SR. This could be a minimum set of raw data that the partner commits to provide to the SR and a minimum set of hardware and software resources (peer-to-peer approach);
5. QoS Requirements for Processing: specify the minimum level of quality of service expected for the processing activity within the SR with the related metrics;
6. QoS Level for Dissemination: specify the minimum level of quality of service expected for the dissemination activity within and outside the SR with the related metrics;
7. QoS Level for Data Storage: specify the minimum level of quality of service expected for data storage within the SR with the related metrics.

3 USE CASE: SR CREATION AND INSTANTIATION

The UML use case of Figure 2 depicts the steps that are performed for creating, instantiating and managing Semantic Rooms.

In our system there exists a service provider responsible for deploying the middleware solution that supports the construction of what we refer to as *Semantic Room-enabled environment*; that is, an environment consisting of SRs possibly communicating with one another. An example of a service provider could be SWIFT (Society for Worldwide Interbank Financial Telecommunication) or any other third-party service provider. The service provider is also in charge of creating the *Basic Semantic Room*, thus becoming its owner and administering it.

The basic Semantic Room is a special type of Semantic Room that can be created, instantiated and administered as any other kind of Semantic Room. It provides basic services to financial institutions that wish to be part of the Semantic Room-enabled environment and is regulated by a basic Semantic Room contract. Basic services include among others the capabilities to (i) create other semantic rooms, (ii) instantiate SRs that have been created, (iii) join existing SRs that have been instantiated and have a SR owner, and (iv) monitoring activities carried out within some SRs. Finally, the service provider can create, instantiate and administer any type of Semantic Room.

A financial institution that wishes to be part of the SR-enabled environment has to first join the basic SR. In order to join the basic SR a financial institution has to register as a basic SR member with the Service Provider responsible for administering the basic SR. Once the registration is completed and approved, the financial institution becomes a semantic room member (i.e., a basic SR member): from that moment on

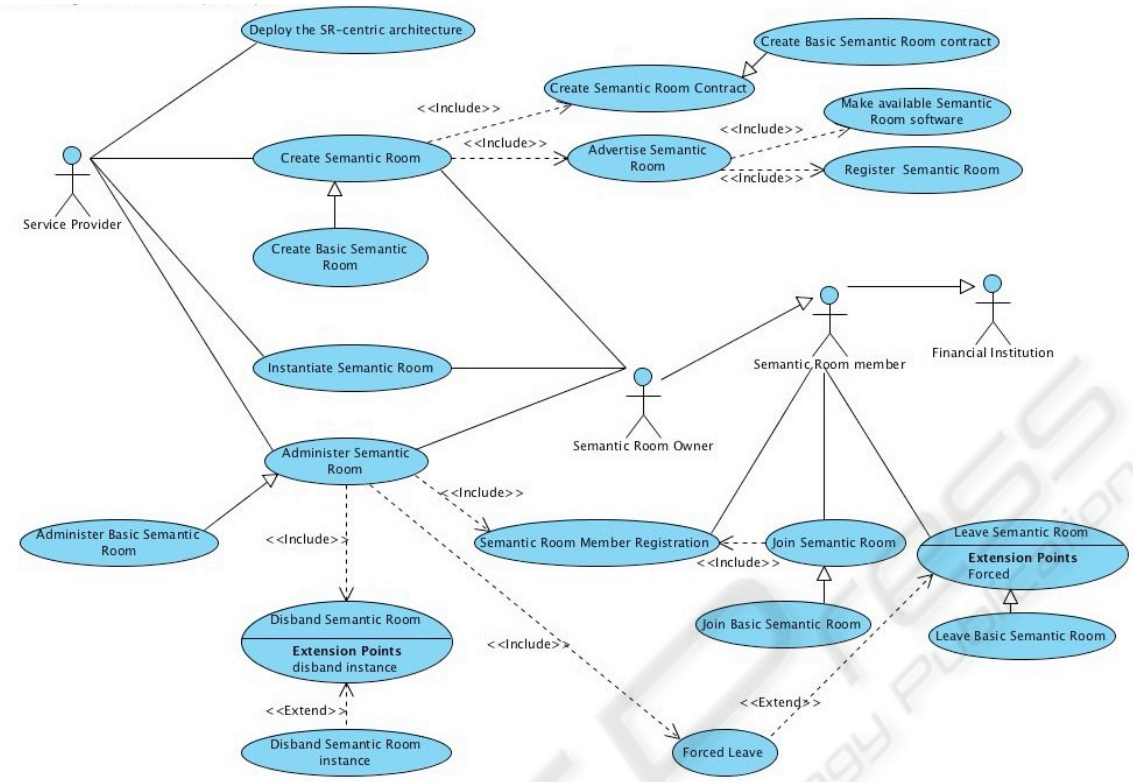

Figure 2: SR creation and management use case.

the member can take advantage of the earlier mentioned basic services, creating, instantiating, administering, joining, leaving and monitoring other semantic rooms. In particular, the basic SR member can:

- *create* new (specialized) SRs. The creation phase consists of the following two main steps: (1) the creation of the Semantic Room contract, where the precise objective for which the SR has been created and all the contract requirements are defined; and (2) the SR advertising. The advertising step is being further broken down as follows: the Semantic Room is registered using a discovery service, and the SR software is made available through the same service. The SR software is the software necessary to realize the specific SR logic and achieve the objective defined in the SR contract.

  A member of the basic SR that creates a specialized SR might become the owner of that SR provided it wishes to assume that specific role;

- *instantiate* an SR that has been previously created (for example by the service provider or other SR basic members). In this specific case, it is mandatory for the member that instantiates the SR to become the SR owner.

  The SR administration responsibilities lay with the SR owner, and include the following: managing member registration (see below), disbanding the SR, forcing some member(s) to leave the SR (this specific case can occur when some member is behaving maliciously) and monitoring whether the members comply with the SR contract clauses.

  The disbanding of a Semantic Room can involve an instance of a specific SR. In this case, disbanding that SR implies closing the SR and deleting its membership. In addition, the disbanding of a Semantic Room can include deleting the created SR, (i.e., removing it from the discovery service previously mentioned).

- *join* existing SRs that have been instantiated by the SR owner. In order to execute the join operation, a basic SR member shall register with the SR owner. The SR owner will check if the member has the right credentials to be part of the SR and, if so, the member is included in the membership of that SR. At that point, the member can download the specific SR software required for gaining the intelligence from the SR.

- *voluntary leave* SRs the member is part of (in this case, the SR owner deletes the member from the membership), or voluntary leave the entire SR-enabled environment by revoking its participation in the basic Semantic Room.

3.1 SR Usage Example

In this section we present an applied example of an SR formed by financial institutions that wish to use and exploit the SR model in order to implement a basic intrusion detection mechanism. The output of this mechanism is a blacklist contains IP addresses or range of IP addresses that may have attempted to break into various financial institution sites.

The scenario we consider here is that of an intruder that executes stealthy scans across multiple financial sites (attack subjects can be external web servers in the demilitarized zone of the SR financial members). The goal of the intruder is to identify TCP and UDP ports of financial institutions that may have been left open and to use those ports as intrusion vectors to access the services listening on those ports.

From a technical viewpoint, detecting these types of attacks requires the ability to analyze and process a large number events per second, and correlate them in a very large time window. The added value of the SR abstraction is manifested through the ability to correlate the sensor data coming from multiple sites which enables identifying intrusion attempts where the number of ports tried within each individual SR participant is below the local threshold. The specific heuristic being used looks for the pattern of unusually high total number of requests originating from a specific source IP address and directed to distinct the pairs [host, port]. The SR then disseminates the processed information to the SR members and raises an alert that a specific IP address is performing malicious activities. It will be the responsibility of each individual financial institution to undertake the proper countermeasures to protect itself.

There can be different approaches to be used in order to achieve these objectives. One approach can be fully centralized (esp, 2009): every financial institution sends its own raw data to a central location responsible for carrying out the correlation and analysis of the data. This approach has the advantage to increase the detection accurancy; however, it exhibits the inherent drawbacks of a central system. In contrast to the above approach, another one can be to apply a fully decentralized system architecture where the processing and storage load is spread over the entire population of the SR members thus avoiding performance hot-spots and parallelizing the processing. In our current implementation we have favored a fully decentralized approach. For the sake of brevity we do not describe the system here. The interested readers can refer to (Lodi et al., 2010a) for a detailed description and preliminary evaluation of this system. However, it is worth noting that the SR abstraction described in this paper can be used to deploy all the various aforementioned system architecture schemes: a created SR can be associated with potentially more than one instance according to the various deployed technologies, used in turn to realize the SR complex event processing.

4 CONCLUDING REMARKS

In this paper we have presented a new architectural abstraction, called Semantic Room, which can be used to construct collaborative security systems. We have applied the SR abstraction in the context of the financial infrastructure protection. In particular, we demonstrated how the SR model can be used to assist financial institutions in forming federations aimed to protect against security threats, such as intrusion.

The SR functionality is determined by their objectives and is regulated by contracts. The contracts define the set of rules that govern the participation in the SRs along with all the security, isolation, trust and performance requirements that are to be met in order to perform the information processing and sharing.

SRs can be customized; that is, the SR logic that allows it to achieve the objective specified in the contract is to be made available for the use by the SR members. Being customizable, SRs may deploy various system architecture schemes (e.g., a centralized scheme, a fully decentralized one). Our current effort is centered on the experimental evaluation of the centralized and decentralized approaches. Although the results of this evaluation are not shown in this paper, it seems that when high volumes of data are to be correlated and analyzed and no complex event correlation patterns are to be detected, a fully decentralized approach would be favorable as it provides a coarse-grained detection and copes with scalability requirements crucial to accommodate high rates of the incoming events. In contrast, when complex event correlation in time and space is required and the volumes of data are not significantly high, a centralized approach can be effectively exploited.

ACKNOWLEDGEMENTS

This research is partially funded by the EU project CoMiFin (Communication Middleware for Monitoring Financial Critical Infrastructure). The authors wish to thank their project colleagues for their valuable comments and suggestions on the described Semantic Room model.

REFERENCES

(2009). Basel II Accord. http://www.bis.org/bcbs/ bcbscp3.htm.

(2009). Where Complex Event Processing meets Open Source: Esper and NEsper. http://esper.codehaus.org/.

Balakrshnan, B. M. and Stonebraker, H. M. (2004). Contract-based load management in federated distributed systems. In *1st Symposium on Networked Systems Design and Implementation*, San Francisco, CA, USA.

Chandy, M. K. (2006). Event-Driven Applications: Costs, Benefits and Design Approaches. Presented at the Gartner Application Integration and Web Services Summit, http://www.infospheres.caltech.edu/node/ 38.

Krügel, C., Toth, T., and Kerer, C. (2001). Decentralized event correlation for intrusion detection. In *ICISC*, pages 114–131.

Lamanna, D., Skene, J., and Emmerich, W. (2003). Slang: A language for defining service level agreements. In *FTDCS '03: Proceedings of the The Ninth IEEE Workshop on Future Trends of Distributed Computing Systems*, page 100, Washington, DC, USA. IEEE Computer Society.

Locasto, M. E., Parekh, J. J., Keromytis, A. D., and Stolfo, S. J. (2005). Towards collaborative security and p2p intrusion detection. In *IEEE Workshop on Information Assurance and Security*, United States Military Academy, West Point, NY.

Lodi, G., Baldoni, R., Bortnikov, V., Chockler, G., Dekel, E., Laventman, G., and Angori, E. G. (2010a). A Collaborative Environment for Customizable Complex Event Processing in Financial Information Systems. Technical Report MIDLAB 5/2010.

Lodi, G., Baldoni, R., Elshaafi, H., Mulcahy, B., Csertain, G., and Gonczy, L. (2010b). Trust Management in Monitoring Financial Critical Information Infrastructures. In *The 2nd International Conference on Mobile Lightweight Wireless Systems - Critical Information Infrastructure Protection Track*.

Moore, D., Paxson, V., Savage, S., Shannon, C., Staniford, S., and Weaver, N. (2003a). Inside the Slammer Worm. *IEEE Security and Privacy*, 1(4):33–39.

Moore, D., Shannon, C., Voelker, G. M., and Savage, S. (2003b). Internet quarantine: Requirements for containing self-propagating code. In *INFOCOM*.

Xie, Y., Sekar, V., Reiter, M. K., and Zhang, H. (2006). Forensic analysis for epidemic attacks in federated networks. In *ICNP*, pages 43–53.