IJDCST @ Nov-Dec, Issue- V-3, I-1, SW-76
ISSN-2320-7884 (Online)
ISSN-2321-0257 (Print)

Reinforcement of Access Policies for Event Processing Systems

M.S.V. Praveen Kumar[1], Rehana Begum[2]

[1] M.Tech(CSE), Nimra College of Engineering and Technology, A.P., India.

[2]Asst. Professor, Dept. of Computer Science & Engineering, Nimra College of Engineering and Technology, A.P., India.

Abstract — There are lacking in methods of current event processing systems to preserve privacy constraints of incoming event streams in a chain of subsequently applied stream operations. This is a problem in large-scale distributed applications like a logistic chain where event processing operators may be spread over multiple security domains. An adversary can infer from legally received outgoing event streams confidential input streams of the event processing system. This paper presents a fine-grained access management for complex event processing. Each incoming event stream can be protected by the specification of an access policy and is enforced by algorithms for access consolidation. The utility of the event processing system is increased by providing and computing in a scalable manner a measure for the obfuscation of event streams. An obfuscation threshold as part of the access policy allows ignoring access requirements and delivering events which have achieved a sufficient high obfuscation level.

Keywords— Event Processing; Security; Access control;

I. INTRODUCTION

It is essential in business processes, to detect inconsistencies or failures early. For example, in manufacturing and logistic processes, items are tracked continuously to detect loss or to reroute them during transport. To answer this need complex event processing (CEP) systems have evolved as a key paradigm for business and industrial applications [1], [2]. CEP systems allow to detect situations by performing operations on event streams which emerge from sensors all over the world, e.g. from packet tracking devices.

While, traditionally event processing systems have applied powerful operators in a central way, the emerging increase of event sources and event consumers have raised the need to reduce the communication load by distributed in-network processing of stream operations [3]. In addition, the collaborative nature of today's economy results in large-scale networks, where different users, companies, or groups exchange events. As a result, event processing networks are heterogeneous in terms of processing capabilities and technologies, consist of differing participants, and are spread across multiple security domains [4], [5]. However, the increasing interoperability of CEP applications raises the question of security [2]. It is not feasible for a central instance to manage access control for the whole network. Instead, every producer of information should be able to control how its produced data can be accessed. For example, a company may restrict certain information to a subset of authorized users (i.e. that are registered in its domain). Current work in providing security for event-based systems covers already confidentiality of individual event streams and the authorization of network participants [6]. In CEP systems, however, the provider of an event loses control on the distribution of dependent event streams. This constitutes a major security problem, allowing an adversary to infer information on confidential ingoing event streams of the CEP system.

Figure 1 Access Control & Event Dependency

As an example consider the logistics process illustrated in Figure 1 where a manufacturer wants to deliver an item to a destination. The shipping company determines a warehouse close to the destination, where the item will be shipped to before it will be delivered to the customer. The logistic process is supported by an event processing system, where operators are hosted in the domain of each party and exchange events including potentially confidential information (e.g. the item's destination is transmitted to the shipping company). If now a third party receives events related to the warehouse, it may draw conclusions about the original event data (i.e. destination), in spite of the manufacturer declaring this information as highly confidential and only providing the shipping company with access rights to it.

The goal of this work is to establish access control that ensures the privacy of information even over multiple processing steps in a multi-domain, large scale CEP system. In particular, our contributions are i) an access policy inheritance mechanism to enforce access policies over a chain of dependent operators and ii) a scalable method to measure the obfuscation imposed by operators on information exchanged in event streams. This allows defining as part of the access policy an obfuscation threshold to indicate when the event processing systems can ignore access restrictions, thus increasing the number of events to which application components can react to and this way increasing also the utility of the CEP system.

Figure 2 Attributes in Shipping Scenario

II. Related Work

With the increasing popularity of event-driven systems, a lot of effort has been spent to make the systems secure. For example, a role-based access control is proposed in [3]. Pesonen et al. and Bacon et al. discuss how publish/subscribe systems can be secured by introducing access control policies in a multi-domain architecture [7], [8]. They describe how event communication between the domains can be supported. Opyrchal et al. present the concept of event owners that can be specified. These are used to provide access to their events [9]. Tariq et al. propose a solution to provide authentication and confidentiality in broker-less content-based publish/subscribe systems [6]. Our work is based on the previous work which make event communication secure among different entities in the system. We assume the presence of a system that can handle access control on events. Based on this, we use policy composition in order to derive the necessary access policies at any point during the event processing steps.

Access policy composition has found a lot of consideration
in distributed systems. Bonatti et al. defined a well-recognized algebra for composing access policies [10]. Especially in the area of web service composition, the composition of security policies plays an important

role, as different policies have to be combined for every combination of web services (e.g. [11]). We adopt some of these concepts into our distributed CEP system, which allows us to inherit access restrictions during the different processing steps in the operators of our system. To realize our concepts we make use of techniques from statistical inference. More specific, we calculate the Bayesian inference after creating a Bayesian network and learning the dependencies. Since Bayesian inference is a complex calculation, several Monte-Carlo algorithms have been proposed to estimate the inference value(s). They all have in common to arbitrarily pick samples from the Bayesian network probability distribution, and estimate the values based on the samples. The precision of the estimated inference values is dependent on the number of samples. A commonly used technique is the Gibbs sampler [12].

III. PROPOSED WORK

We assume a distributed correlation network, where dedicated hosts are interconnected. On these hosts we deploy operators, which are executed to collaboratively detect situations and form the distributed CEP system. The cooperative behavior of the operators is modeled by a directed operator graph $G = (\Omega, S)$ which consists of operators $\omega \in \Omega$ and event streams $(\omega i, \omega j) \in S \subseteq (\Omega \times \Omega)$ directed from ωi to ωj . Thus, we call ωi the event producer and ωj the consumer of these events. Each event contains one or more event attributes which have discrete values. Every operator ω implements a correlation function $f\omega : I\omega \rightarrow O\omega$ that maps incoming event streams $I\omega$ to outgoing event streams $O\omega$. In particular, $f\omega$ identifies which events of its incoming streams are selected, how event patterns are identified (correlated) between events, and finally how events for its outgoing streams are produced.

Figure 2 illustrates an operator graph of three operators according to the introduced logistics example, each operator hosted in a distinct domain. The correlation function fsc is applied to events received from and produced by ωm on produced items in the manufacturing domain. Events produced by fsc carry two event attributes, the warehouse location and estimated day of delivery for shipped items.

Our approach allows inheriting access requirements by assigning them to event attributes in form of an access policy. This allows preserving requirements through any chain of dependent correlation steps of operators in G. In addition, an obfuscation policy allows to specify an obfuscation threshold for event attributes. In each correlation step, the obfuscation of event attributes in produced events is determined by the proposed access policy consolidation protocol. Once the obfuscation threshold is reached for an event attribute, the attribute's access requirements can be ignored. In the following, we detail the concepts behind access policies and obfuscation policies, and formalize the security goal.

A. Access Policies

Access control allows to specify access rights of subjects (operators) for the set of available objects (event attributes).

These access rights are provided by the owner of an object (e.g. the producer of an event stream) and may be granted to operators based on an access requirement. Such a requirement may be a role, a location or a domain affiliation. Requirements are usually not direct properties of the operators, but of the hosts where the operators are deployed. Formally, we specify the access rights within an access policy

AP for an operator ω as a set of (attribute, access requirement) pairs:

$$AP\omega = \{(att1, ar1), ..., (attn, arn)\}$$

If there is no requirement specified for an attribute, any consumer in the network will be able to access it. Note that we consider attributes to be distinct even if they use the same name, but are produced at two distinct operators. An access requirement is a tuple of a property p, a mathematical operator op and a value set val: ar = (p, op, val), where op ∈ {=,<,>,≤,≥, ∈}. val can be specified by a range or a set of values. For the sake of simplicity, in this paper access requirements are only referring to domain affiliation and have a structure like this:

$$ar1 = (domain, \in, \{domainA, domainB\}).$$

In our example scenario, the manufacturer's event attributes have different access requirements. While the information about the item's destination is accessible by the customer, information about where the item is produced and when it can be picked up is restricted to the shipping company. Therefore, the attached AP is defined as follows:

APmanufacturer =
{(destination,(domain,∈,*{*shippingComp,customer*}*)),
(pickup time, (domain,=,shippingComp)),
(production place, (domain,=,shippingComp))*}*

With the enforcement and assurance of access policies at each producer, a consumer will be eligible to access (receive) an attribute only if the consumer's properties match the access requirements defined for the particular attribute. In this case the consumer is trusted to use the attribute in its correlation function and adopt the requirements specified for the attribute in its own access policy for all produced events.

B. Obfuscation of Event Information

While access policies allow a producer to specify access requirements in a fine-grained manner, the inheritance of requirements in a chain of succeeding operators is at times very restrictive and can limit the efficiency and applicability of the CEP system: in each correlation step of this chain, the number of access requirements may increase by the consolidation of requirements from multiple producers. Each consolidation step can therefore increase the number of interested consumers which are prevented from access to the event attributes of produced event streams. This does not reflect the nature of event processing systems where basic events like single sensor readings may have only little influence on the outcome contained in a complex event representing a specific situation.

In our logistics example, fsc uses destination, production place and pickup time to determine the estimated day of delivery. As a consequence, the customer has no access to the estimated day of delivery of the ordered item, since she does not fulfill the access requirements for production place and pickup time. Yet she has a reasonable interest in this information. And one may claim, that knowledge of the day of delivery does not necessarily allow to draw a relevant conclusion on the production place and pickup time attribute values. We say, the attribute values get obfuscated during the correlation process and depending on the achieved level of obfuscation, the access requirements of an attribute may no longer be needed. In our approach, the level of obfuscation is

a measure, to which extent a consumer of the produced attribute (estimated day of delivery) can infer the value of the original attribute (production place). It can be easily seen in the example, that obfuscation is not only dependent on the values of the attributes, but also on the knowledge of the consumer. Since the destination value has led to the day of delivery as well, knowledge of the destination would be of great help when trying to infer the restricted attribute production place because the delivery time of the item is probably related to the distance between destination and production place. In this work, we will use obf(att_{old}, $attn_{ew}$, ω) to refer to the obfuscation achieved by attnew for attold given the knowledge available at a consumer ω ∈ Ω

C. Security Goal

Let att_{old} →ω att_{new} denote that

1) at some operator ω ∈ Ω, attold is taken as input to the correlation function fω, and
2) fω produces attnew in dependence of attold.

Furthermore, let attold →∗ attnew denote the transitive closure of the dependency relation. For any pair of attributes with att_{old} →*att_{new} we say that attnew is dependent on attold. Our main goal is to preserve the privacy of event attributes over multiple correlation steps by respecting the dependency relationship between the attributes produced by the CEP system. In particular, access requirements must not be applied solely to the attribute attold, but have to be inherited to all dependent attributes (attnew) unless a sufficient obfuscation threshold for attnew has been reached.

More formally, given for each attribute att an initial set of access requirements denoted by ARinit(att). We require for any policy consolidation algorithm:

Algorithm 1 Local Obfuscation Calculation

```
procedure INITIALIZE(ω)
    for all operator ω do
        D_ω ← FINDMULTIPATHOPERATORS(ω)
    end for
    for all ω ∈ D_ω do
        relAtts ←FINDRELATEDATTRIBUTES
        for all (att_new, att_old) ∈ relAtts do
            TRANSMIT P(att_new|att_old) TO ω
        end for
    end for
end procedure
procedure UPONRECEIVEEVENT(e)
    for all att ∈ e do
        if ∃ multPathDependency(att) then
            CALCULATEWORSTCASEOBFUSCATION(ATT)
        else
            CALCULATELOCALOBFUSCATION(ATT)
        end if
    end for
end procedure
```

IV. CONCLUSION

This paper addressed the inheritance and consolidation of access policies in heterogeneous CEP systems. We identified a lack of security in multi-hop event processing networks and proposed a solution to close this gap. More specific, we presented an approach that allows the inheritance of access requirements, when events are correlated to complex events. Our algorithm includes the obfuscation of information, which can happen during the correlation process, and uses the obfuscation value as a decision-making basis whether inheritance is needed. We presented an implementation of our approach, based on Bayesian Network calculations. The analysis and evaluations show that the approach is computation-intensive, once the Bayesian Network grows, hence raising the processing time of an event. To deal with the

calculation cost, we introduced a local approach, where every participant calculates local obfuscation achieved during the correlation process. We use a variable elimination optimization to further reduce the computational effort for calculating obfuscation. Future work will concentrate on enhancing the obfuscation calculation and methods to increase the Bayesian Network size so we are able to measure obfuscation over more than one correlation steps.

V. REFERENCES

[1] A. Buchmann and B. Koldehofe, "Complex event processing," it - Information Technology, vol. 51:5, pp. 241–242, 2009.

[2] A. Hinze, K. Sachs, and A. Buchmann, "Event-based applications and enabling technologies," in Proceedings of the Third ACM International Conference on Distributed Event-Based Systems, ser. DEBS '09. New York, NY, USA: ACM, 2009, pp. 1:1–1:15.

[3] P. Pietzuch, "Hermes: A scalable event-based middleware," Ph.D. dissertation, University of Cambridge, 2004.

[4] B. Schilling, B. Koldehofe, U. Pletat, and K. Rothermel "Distributed heterogeneous event processing: Enhancing scalability and interoperability of CEP in an industrial context," in Proc. of the 4th ACM International Conference on Distributed Event-Based Systems (DEBS), 2010, pp. 150–159.

[5] B. Schilling, B. Koldehofe, and K. Rothermel, "Efficient and distributed rule placement in heavy constraint-driven event systems," in Proc. of the 10th IEEE International Conference on High Performance Computing and Communications (HPCC), 2011, pp. 355–364.

[6] M. A. Tariq, B. Koldehofe, A. Altaweel, and K. Rothermel, "Providing basic security mechanisms in broker-less publish/ subscribe systems," in Proceedings of the 4th ACM Int. Conf. on Distributed Event-Based Systems (DEBS), 2010, pp. 38–49.

[7] L. I. W. Pesonen, D. M. Eyers, and J. Bacon, "Encryption-enforced access control in dynamic multidomain publish/subscribe networks," in Proc. of the 2007 ACM International Conference on Distributed Event-Based Systems (DEBS), 2007, pp. 104–115.

[8] J. Bacon, D. M. Eyers, J. Singh, and P. R. Pietzuch, "Access control in publish/subscribe systems," in Proc. of the 2nd ACM International Conference on Distributed Event-Based Systems (DEBS), 2008, pp. 23–34.

[9] L. Opyrchal and A. Prakash, "Secure distribution of events in content-based publish subscribe systems," in In Proceedings of the 10th USENIX Security Symposium, 2001, pp. 281–295.

[10] P. Bonatti, S. De Capitani di Vimercati, and P. Samarati, "An algebra for composing access control policies," ACM Trans. Inf. Syst. Secur., vol. 5, pp. 1–35, February 2002.

[11] F. Satoh and T. Tokuda, "Security policy composition for composite web services," Services Computing, IEEE Transactions on, vol. 4, pp. 314 – 327, 2011.

[12] S. Geman and D. Geman, "Stochastic relaxation, gibbs distributions, and the bayesian restoration of images," Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. PAMI-6, pp. 721 –741, 1984.