Predictive Security Analysis for Event-driven Processes

Roland Rieke and Zaharina Stoynova

Fraunhofer Institute for Secure Information Technology SIT, Darmstadt, Germany
{roland.rieke,zaharina.stoynova}@sit.fraunhofer.de

Abstract. This paper presents an approach for predictive security analysis in a business process execution environment. It is based on operational formal models and leverages process and threat analysis and simulation techniques in order to be able to dynamically relate events from different processes and architectural layers and evaluate them with respect to security requirements. Based on this, we present a blueprint of an architecture which can provide decision support by performing dynamic simulation and analysis while considering real-time process changes. It allows for the identification of close-future security-threatening process states and will output a predictive alert for the corresponding violation.

Keywords: predictive security analysis, analysis of business process behaviour, security modelling and simulation, complex event processing

1 Introduction

With the increased adoption of service oriented infrastructures and architectures, organisations are starting to face the need for an accurate management of cross-process and cross-layer security information and events. The main constraint of current systems is the restriction of Security Information and Event Management (SIEM) [8] to network infrastructure, and the inability to interpret events and incidents from other layers such as the service view, or the business impact view, or on a viewpoint of the service itself. Conversely, specific service or process oriented security mechanisms are usually not aware of attacks that exploit complex interrelations between events on different layers such as physical events (e.g. access to buildings), application level events (e.g. financial transactions), business application monitoring, events in service oriented architectures or events on interfaces to cloud computing applications. Nevertheless, next generation systems should be able to interpret such security-related events with respect to specific security properties required in different processes. On the base of these events, the system should be able to analyse upcoming security threats and violations in order to trigger remediation actions even before the occurrence of possible security incidences.

In this paper we propose to combine process models with security policies and a security model in order to identify potential cross-cutting security issues. We furthermore suggest a blueprint of an architecture for predictive security analysis that leverages process and threat analysis and simulation techniques in order to be able to dynamically relate events from different execution levels, define specific level abstractions and evaluate them with respect to security issues.

2 Related Work

Our work combines aspects of process monitoring, simulation, and analysis. Some of the most relevant contributions from these broad areas are reviewed below.

Business Activity Monitoring (BAM). The goal of BAM applications, as defined by Gartner Inc., is to process events, which are generated from multiple application systems, enterprise service busses or other inter-enterprise sources in real time in order to identify critical business key performance indicators and get a better insight into the business activities and thereby improve the effectiveness of business operations [6]. Recently, runtime monitoring of concurrent distributed systems based on LTL, state-charts, and related formalisms has also received a lot of attention [5, 3]. However these works are mainly focused on error detection, e.g. concurrency related bugs. In the context of BAM applications, in addition to these features we propose a *close-future* security analysis which provides information about possible security risks and threats reinforcing the security-related decision support system components.

Complex Event Processing (CEP). CEP provides a powerful analytic computing engine for BAM applications which monitor raw events as well as the real-time decisions made by event scenarios. David Luckham [4] provides us with a framework for thinking about complex events and for designing systems that use such events. A framework for detecting complex event patterns can be found e.g. in [10]. However such frameworks concentrate on detecting events important for statistical aspects, redesign and commercial optimisation of the business process. Here we want to broaden the scope of the analysed event types by introducing *complex security events* in the CEP alphabet.

Simulation. Different categories of tools that are applicable for simulation of event-driven processes including process modelling tools based on different semi-formal or formal methods such as Petri Nets [2] or Event-driven Process Chains (EPC) [1]. Some process managements tools, such as FileNet [7] offer a simulation tool to support the design phase. Also some general purpose simulation tools such as CPNTools [11] were proven to be suitable for simulating business processes. However, independently from the tools and methods used, such simulation tools concentrate on statistical aspects, redesign and commercial optimization of the business process. On the contrary, we propose an approach for *on-the-fly* intensive dynamic simulation and analysis considering the current process state and the event information combined with the corresponding steps in the process model.

Security Information Management (SIM). SIM systems generally represent a centralized server acting as a "security console", sending it information about security-related events, which displays reports, charts, and graphs of that information, often in real time. Commercial SIEM products include

Cisco Security Monitoring Analysis and Response System (`http://www.cisco.com/en/US/products/ps6241/index.html`), EventTracker by Prism Microsystems (`http://www.prismmicrosys.com/EventTrackerSIEM/index.php`), SenSage (`http://www.sensage.com/products/sensage-40.php`) and others. All these products monitor the low-level events (such as network events) and perform event correlation only on the base of event patterns and rules. Our approach additionally considers the business process level events combined with the current process state and business process information provided by a process specification.

3 Blueprint of Architecture for Security Event Processing and Predictive Security Monitoring

In this section we introduce our approach for security evaluation of event-driven processes. Figure 1 depicts the core components which we consider necessary in order to be able to perform a security event processing and monitoring analysis in the context of a running event-driven business process.

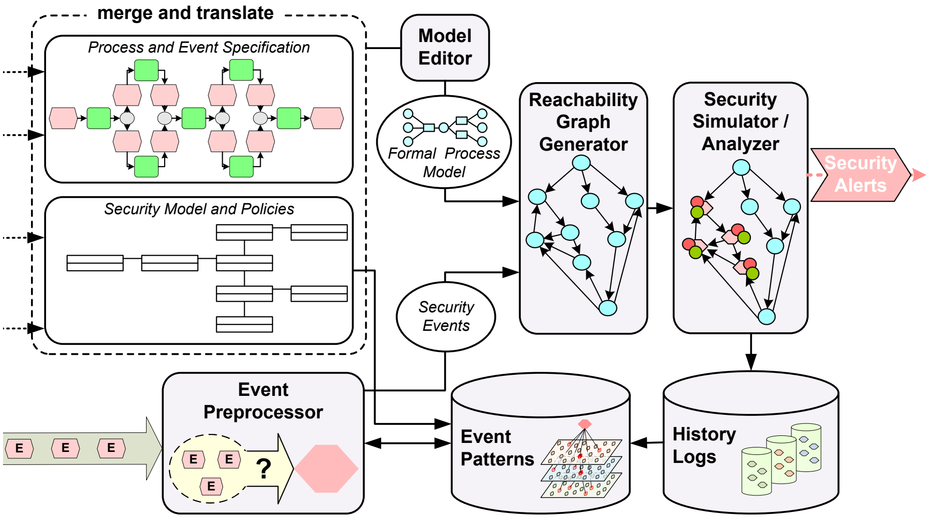


Fig. 1. Predictive Security Analyser

The input elements which we need comprise, (1) a *process model* given in a notation such as EPC, BPEL, YAWL or BPMN that contains a specification of the events which can be triggered during runtime, (2) *security policies* which contain information about the relations between the users involved in the process, their roles and the relations between the roles and resources deployed by the process, (3) a *security model* that should provide information about the process's predefined security requirements which will be used to construct the security events patterns, and, (4) *real-time events* which will be triggered during runtime.

Model Editor. In order to analyse the system behaviour with tool support, an appropriate formal representation has to be chosen because semi-formal languages such as BPMN allow to create models with semantic errors [2]. In our approach, we use an operational finite state model based on *Asynchronous Product Automata (APA)* [9]. An APA consists of a family of so called *elementary automata* communicating by common components of their state (shared memory). The process model, the organisational model and the security model should be imported and merged in a high-level model of the process and then this model is translated into an APA, which will enable the computation of the possible system behaviour. In general, we could also use other descriptions of processes with unambiguous formal semantics here such as the approaches in [2] for BPMN or [1] for EPC that allow for computation of possible system's behaviour.

Reachability Graph Generator. Formally, the behaviour of an APA can be given by a reachability graph which represents all possible coherent sequences of state transitions starting with the initial state. In the context of on-the-fly security analysis the reachability graph will represent the path given by the already triggered events, forwarded by the Event Preprocessor. The computation will be automatically paused each time when the current state (according to the triggered events) of the process is reached. In the context of predictive simulation analysis the Reachability Graph Generator computes all possible near-future paths according to the given process specification, (e.g. sequences of at most 2-3 plausible events). This will allow exhaustive analysis of all near-future states to be performed in order to compute whether there exist possible security-threatening states of the process which can compromise the process security and match some of the event patterns saved in the Event Patterns database.

Security Simulator/Analyser. During the computation of the graph this component will check for each state, whether the specified security properties are fulfilled and trigger security alarms when possible security violations are found. Furthermore, it is possible to detect new security violations that were not predicted by the available security patterns. In order to include them in the analysis of future process instances, they will be logged in the History Logs database and then they will be transformed into security event patterns and saved in the Event Patterns database. The simulator will also enable security analysis by performing intensive simulation which inspects the behaviour of complex/parallel processes under given hypotheses (*what-if analysis*) concerning changes in the organisational model/security policies or the process model.

Security Event Patterns. These patterns which are relevant for the corresponding process are kept in the Event Patterns database and they should be extracted from the provided security model. In order to be able to reason about potential security problems, based on real life events, specific abstractions are included in this extraction process so that the abstraction levels for the various types of security-related events can be interrelated. Solutions for these kind of security analysis are already available but usually limited to a narrow field of application such as IDS where e.g. the detection of a number of abnormal connections could lead to a “worm detection” alarm. We propose a generic approach leveraging these ideas and incorporating other types of security related events.

Event Preprocessor. In the context of on-the-fly security analysis the Event Preprocessor is responsible for receiving the real-life events triggered during runtime, matching them against the available security event patterns and forwarding them to the Reachability Graph Generator. During predictive security analysis the Event Preprocessor will generate all possible events according to the process specification and will match them against the event patterns. Then it will forward them to the Reachability Graph Generator in oder to enable the computation of the process graph.

History Logs. In the History Logs database newly detected security-violating sequences of events will be logged. These will be used to create new security event patterns.

4 An Application Scenario

For illustrating how our architecture components, described in the previous section, collaborate we will refer to a common example scenario for online credit application.

4.1 Process Model

In an EPC graph events are represented as hexagons and functions that describe state transitions are represented as rounded rectangles. Now consider the online credit application process expressed in EPC notation in Fig. 2. The process starts when an applicant submits an application form. Upon receiving a new application form a credit clerk performs checks in order to validate the applicant’s income and other relevant information. Depending on the requested loan amount different checks are performed. Then the validated application passed on to a manager to decide whether to accept or reject it. In both cases the applicant is notified of the decision and the process ends.

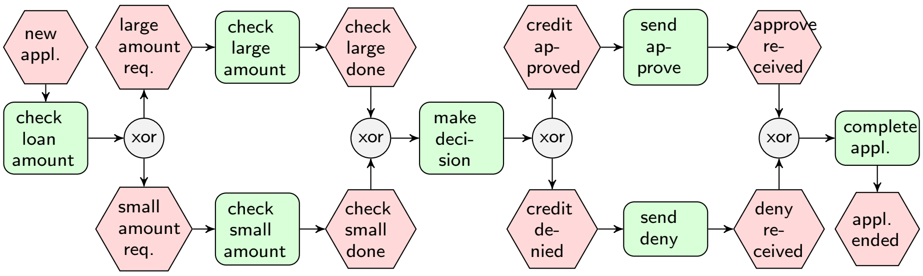
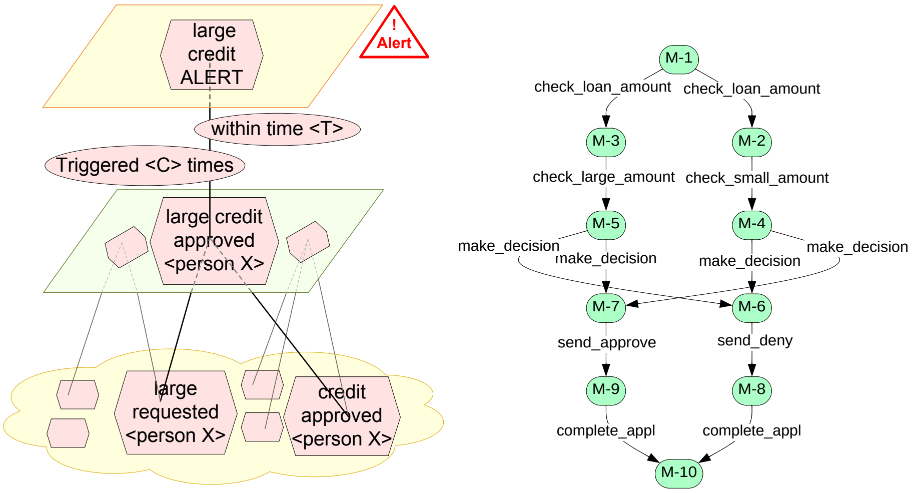


Fig. 2. Business Process Model

4.2 Predicting Security Events

In our example scenario we consider the security event "*large credit ALERT*" which is raised when too many large credits are approved for one customer (see Fig. 3(a)). This is an example of an event abstraction or complex event generated by a certain sequence of simple events, triggered in the process. Such complex events are generated by CEP engines whenever certain predefined sequences of events have been triggered.

Additionally, we apply such complex event patterns in a predictive way. This means that whenever an event pattern is *probably* going to match by taking into account a current partial match and a possible continuation of the current state, these abstractions can be generated prior to the real-time triggering of the simple events. In our example we generate an abstraction of the atomic events "*large amount requested*" and "*credit approved*" triggered by the same customer, namely the complex event "*large credit approved*". Then if this complex event is generated e.g. two times within a certain time and according to security regulations only two large credits can be given to one customer we can generate the alert "*large credit ALERT*" in the upper abstraction level prior to the next approval in order to ensure that the security regulations will not be overseen by taking the credit decision.



(a) On-the-fly security violations (b) Reachability graph of business process

Fig. 3. Predict near future security violations

4.3 Operational Model for Security Event Prediction

A computation of the possible system behaviour of a formal APA model of the business process in Fig. 2 results in the reachability-graph depicted in Fig. 3(b).

The state M-3 e.g. represents the situation where an event of type "*large amount requested*" is available and can be processed by the action "*check_large_amount*" which in turn will trigger an event of type "*check large done*". After this, the process is in state M-5, where the action "*make_decision*" can be executed and lead to one of the two possible followup states M-6 or M-7. M-7 is reached iff the decision results in an event "*credit approved*".

From this we now conclude that a predictive alert "*large credit ALERT*" can be generated if, (1) the system is in a state where the number of large credits allowed for one customer is exhausted, (2) an event "*large amount requested*" for the same customer is received, and, (3) an evaluation of possible continuations of the process's behaviour based on the operational model shows that an additional event of type "*large credit approved*" is possible within the forecast window.

The method described in this paper addresses security properties that can be stated as safety properties. Possible violations of these properties are identified by reachable states in the predicted system behaviour. Some examples of security related event types that can be analysed by the method given in this paper are:

Confidentiality. Consider an event sending a cleartext password. Predict that in one possible continuation of a process, an event about processing a cleartext password locally may lead to an event sending that password.

Authenticity. Consider the physical presentation of a token which is known to be unique such as a credit card or passport as parameter of two different events with very close time and very different location.

Authorisation. Consider two events with persons with the same biometric parameters in different locations at the same time.

Integrity/Product counterfeiting. Consider RFIDs being scanned in places where they are not expected.

Integrity/Safety. Consider two trains on the same railtrack. Predict that a specific constellation of switches leads to a crash in one possible continuation.

5 Conclusions and Further Work

In this paper we proposed a blueprint of an architecture for predictive security analysis of event-driven processes that enables exhaustive process analysis during runtime based on the triggered real-life events. Our approach is based on the specification of an operational finite state model of the process behaviour We have demonstrated how our methods can be applied in order to ensure certain security regulations in the process of online credit application and how we can construct event abstractions on different levels in order to detect current and near-future threats.

Currently our components are prototypically implemented without automated merging and translation mechanisms for the input models and specifications, automated event pattern extraction and new event pattern composition. We used the *SH verification tool* [9] to analyse an exemplary business process model for different concrete instantiations (numbers of clients, and time-horizon) of the model. In the future, we will further develop such techniques in order to automate the security analysis and simulation and extend the method to cover liveness properties.

Furthermore, alerts in today's monitoring systems by themselves bring little value in the process security management if they cannot be acted upon. Therefore, we have to provide additionally to the alerts alternative counter-measure scenarios that can be quantifiable evaluated thanks to simulation. In this way our analysis can be extended to provide feedback to the operators on feasibility and impacts of both attacks and counter-measures.

Acknowledgments. The work presented in this paper was developed in the context of the project Alliance Digital Product Flow (ADiWa) that is funded by the German Federal Ministry of Education and Research. Support code: 01IA08006F.

References

1. Dijkman, R.M.: Diagnosing Differences Between Business Process Models. In: Dumas, M., Reichert, M., Shan, M.C. (eds.) BPM. Lecture Notes in Computer Science, vol. 5240, pp. 261–277. Springer (2008)
2. Dijkman, R.M., Dumas, M., Ouyang, C.: Semantics and analysis of business process models in BPMN. Inf. Softw. Technol. 50(12), 1281–1294 (2008)
3. Kazhamiakin, R., Pistore, M., Santuari, L.: Analysis of communication models in web service compositions. In: WWW'06: Proc. of the 15th international conference on World Wide Web. pp. 267–276. ACM, New York (2006)
4. Luckham, D.: The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems. Addison-Wesley (2002)
5. Massart, T., Meuter, C.: Efficient online monitoring of LTL properties for asynchronous distributed systems. Tech. rep., Université Libre de Bruxelles (2006)
6. McCoy, D.W.: Business Activity Monitoring: Calm Before the Storm. Gartner Research (2002)
7. Netjes, M., Reijers, H., Aalst, W.P.v.d.: Supporting the BPM life-cycle with FileNet. In: Proceedings of the Workshop on Exploring Modeling Methods for Systems Analysis and Design (EMMSAD'06), held in conjunction with the 18th Conference on Advanced Information Systems (CAiSE'06), Luxembourg, Luxembourg, EU. pp. 497–508. Namur University Press, Namur, Belgium, EU (2006)
8. Nicolett, M., Kavanagh, K.M.: Magic Quadrant for Security Information and Event Management. Gartner RAS Core Reasearch Note (May 2009)
9. Ochsenschläger, P., Repp, J., Rieke, R., Nitsche, U.: The SH-Verification Tool Abstraction-Based Verification of Co-operating Systems. Formal Aspects of Computing, The International Journal of Formal Method 11, 1–24 (1999)
10. Pietzuch, P.R., Shand, B., Bacon, J.: A framework for event composition in distributed systems. In: Middleware '03: Proceedings of the ACM/IFIP/USENIX 2003 International Conference on Middleware. pp. 62–82. Springer-Verlag New York, Inc., New York, NY, USA (2003)
11. Rozinat, A., Wynn, M.T., van der Aalst, W.M.P., ter Hofstede, A.H.M., Fidge, C.J.: Workflow simulation for operational decision support. Data Knowl. Eng. 68(9), 834–850 (2009)