

Indicators and an Architecture for Dynamic SOA Security and Compliance

Stephan Neuhaus Bruno Crispo
Dipartimento di Scienze dell'Informazione
Università degli Studi di Trento
via Sommarive 14, 38100 Trento, Italy
First.Last@disi.unitn.it

1 Introduction

The Hospital San Raffaele in Milano, Italy (HSR), faces a number of difficult ethical and technical challenges: Do patients get the correct drugs in the correct amounts? Is the process of drug dispensation compliant with all the relevant laws and regulations? If it is compliant, is that because good controls were in place or was it a lucky coincidence in the functioning of the hospital's system? Is the hospital actually being reimbursed for the money that it extended, or is it making a loss? If the hospital decides to outsource drug dispensation, how can it make sure that its service provider is also compliant?

These questions are not exclusive to HSR; rather, they exemplify the major efforts from regulatory bodies to protect the privacy, the security, and the assets of citizens from mismanagement. The number and the complexity of requirements placed on IT processes has increased (e.g., through regulations such as the Sarbanes-Oxley Act [9]). The pressure is higher for providers of critical infrastructural services (EU Directive on data retention [3] etc.) or holders of sensitive data like HSR.

These trends have a profound impact on the trust models, security policies, security procedures, and security infrastructure that HSR needs to develop and maintain. The problem is however that traditional security research has been concerned with the protection of data such as access control in its classical form [7] or in more sophisticated variants such as history-based [5], usage based [8], or purpose-based [1] access controls. Workflow security papers are no exception [10].

Yet, all this work does not apply to HSR: they are still set out in the classical Dolev-Yao security model where *a one-time failure of a security policy against an infinitely powerful attacker means that the security of the entire system is broken*. Nothing could be more irrelevant in the new technological and business context. In many cases, it is neither possible nor practical to fully control the overall IT infrastructure.

In this paper, we introduce MASTER (Managing Assurance, Security, and Trust for sERvices) [6], a NESSI strategic project that will deliver, among other things,

- measurable, evidence-based indicators for the compliant functioning of business

processes (called *Key Assurance Indicators*, or KAIs) and for the effectiveness and coverage of controls (called *Key Security Indicators*, or KSIs);

- run-time support that allows the translation of controls into *policies* for the various components of the MASTER run-time architecture and that allows the consistent dissemination of such policies; and
- a technical architecture that allows the effective enforcement and assessment of these policies.

One of the case studies for MASTER is based on a concrete process from Hospital San Raffaele (HSR) in Milano, Italy. (The other is from the banking environment.) Private Hospitals with a officially recognized public functions (such as HSR) are charged with administering drugs or with providing diagnostic services to patients that use their structure (e.g., because the corresponding public services are overbooked) and then are authorized to claim the cost of drug dispensation or diagnostic provisioning from the regional state health administration.

As a consequence of their public function, and because they treat sensitive data, the processes of HSR are highly regulated. To give an idea of the amount of regulation, the simple process of authorization and accounting for the dispensation and recompensation of drugs (called “File F”) is subject to at least seven clauses of one Legislative Decree, a decree that is successively amended by various notes and circulars.

Showing that the process is compliant with these objectives is important for HSR, first because of the moral obligations toward patients, but also because failure to be so would mean to lose the status of an institution with public functions, and thus also the related revenues.

In order to assess these processes for compliance and security in a quantitative way, the challenge for an organization like HSR is that existing metrics are often ad-hoc and not necessarily meaningful for the organization. In fact, the case has been made that bad metrics can actually harm an organization. Yet, metrics are mostly selected on the basis of what can be easily computed from existing evidence, not on the basis of what would be useful for the organization. For example, a typical security metric would be the number of computers with outdated virus signatures [4], but it is not at all clear what such a number would mean to HSR. If the number rises by 10%, is HSR then exposed to 10% more risk? What if the computers with outdated virus signatures are not connected to the Internet? This demonstrates the need for *indicators* that have an immediate and intuitive meaning for an organization such as HSR (Section 2).

In parallel with the indicators themselves, we will also need an *architecture* (Section 3) that can capture any needed evidence; compute indicators from evidence; and support indicator definitions that change over time.

2 Indicators

In assessing the compliance (or security) of a business, the usual approach is binary: either the business is judged to be compliant (or secure), or it is not. The reason is that most models of compliance and security are steeped in traditional models of security

where a single observed instance of a failure means the failure of the entire system. Like enforcement, this is beside the point for an organization like HSR, who want *graduated* models of security and especially compliance. This is because in an organization like HSR, failures of security policies will be inevitable; rigorously enforcing all security policies would be prohibitively expensive and would probably also squash any initiative on the part of HSR’s employees. Thus, while the goal remains being always secure and compliant, the emphasis is shifted towards *being in control*. In this case, it makes sense to say that “we are in control 95% of the time”, even when it may not make sense to say “our system is 95% secure” [2]

MASTER offers two kinds of indicators for this: *Key Assurance Indicators* (KAIs) measure the compliant functioning of business processes, and *Key Security Indicators* (KSIs) measure the the effectiveness and coverage of controls. In effect, KAIs measure how compliant a business is, and KSIs measure the extent to which the compliance is the result of good controls—and not just sheer good luck.

In order to arrive at these indicators, control objectives are decomposed until they have become very simple. For example, “comply with File F regulations” could be decomposed into “check whether the person requesting the drugs is a doctor”, among others. Once this decomposition is done, it is usually obvious what kinds of events are needed to assess the compliance and security of the sub-process. In this example, one meaningful indicator would be “number of unauthorized attempts to disburse drugs” in a certain evaluation period. This makes indicators immediately obvious to anyone working at this specific level of detail in the organization.

Indicators can also be found for higher levels. One assurance indicator for the entire File F process for example is “amount of money lost due to non-reimbursed drug disbursements”. Every level of abstraction gets its own indicators.

3 Architecture

The components of the MASTER run-time architecture are depicted in Figure 1. The components that are involved in computing indicators are:

- the graphical workbench, where indicators and their evidence are specified along with control processes, and which produces configuration files for the components in the second tier;
- the policy management component, whose job it is to disseminate changed configuration files (and hence also changed indicators) in a consistent manner;
- the business service that generates raw events;
- the monitoring component that takes raw events and turns them into evidence, for example by reformatting them in a MASTER-standardized way or by using complex event processing to aggregate them;
- the signaling component that disseminates evidence to interested components using a publish/subscribe mechanism; and

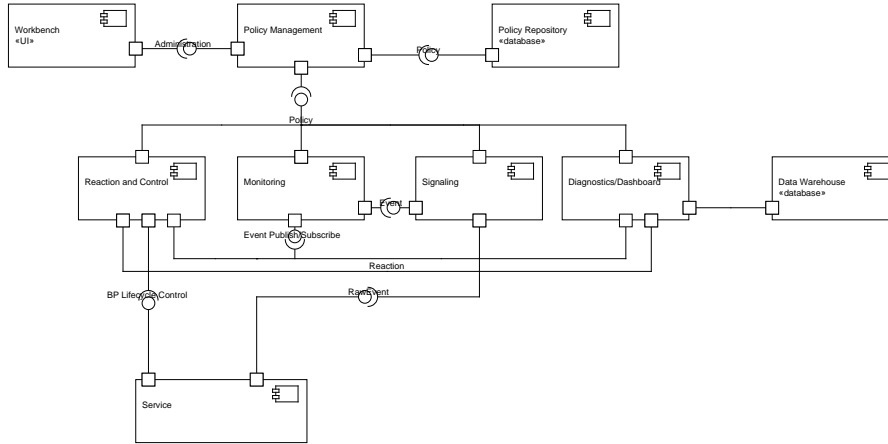


Figure 1: MASTER Run-Time Architecture.

- the assessment component that stores evidence in a database and computes indicators to display them on a dashboard.

4 Enforcement

One new aspect of the above architecture (and the explanation for the one component so far left undescribed) is that MASTER can react to indicators falling outside some agreed range by using *enforcement by reaction*. Traditionally, enforcement is done by *control*, where an action is allowed or disallowed, according to some security policy. This type of enforcement is of course also present in MASTER. However, it works only if it is clear *now* what taking a particular action *now* will have. If an action has a delayed effect, for example on an indicator, enforcement by control will not be possible.

What we can do instead is to specify *compensatory actions* that will be taken if for example an indicator leaves a previously agreed-upon interval. For example, if MASTER recognizes that the number of connections to the hospital system outside of working hours is higher than some threshold, a bandwidth-limiting rule could be activated in the firewall.

This kind of enforcement, which we call *enforcement by reaction*, is a cornerstone of one of MASTER's key innovations, *Protection Level Agreements*, which are analogous to Service-Level Agreements.

5 Conclusion

We have shown the need for quantifiable indicators of security and compliance, and also a component architecture that can deliver these indicators. At the moment, MAS-

TER makes the assumption that all components reside in a *single trust domain*, which means that all components are effectively under a single administrative authority. In the next two years, MASTER will explore the cases where the producers and consumers of events and evidence reside in different trust domains. We expect this to generate many results that will be interesting and useful for dynamic SOA security and compliance.

References

- [1] Rakesh Agrawal, Jerry Kiernan, Ramakrishnan Srikant, and Yirong Xu. Hippocratic databases. In *VLDB*, pages 143–154, 2002.
- [2] Steven M. Bellovin. On the brittleness of software and the infeasibility of security metrics. *IEEE Security & Privacy*, 4(4):96, July–August 2006.
- [3] European Commission. Directive 2006/24/ec of the european parliament. http://eur-lex.europa.eu/LexUriServ/site/en/oj/2006/l_105/l_1_10520060413en00540063.pdf, March 2006.
- [4] Andrew Jaquith. *Security Metrics: Replacing Fear, Uncertainty, and Doubt*. Addison-Wesley Professional, 2007.
- [5] K. Krukow, M. Nielsen, and V. Sassone. A framework for concrete reputation-systems with applications to historybased access control. In *Proc. of CCS'05*, 2005.
- [6] The MASTER Consortium. MASTER: Managing assurance, security and trust for services. <http://www.master-fp7.eu/>, February 2009.
- [7] Pierangela Samarati and Sabrina De Capitani di Vimercati. Access Control: Policies, Models, and Mechanisms. In *FOSAD 2001/2002*, volume 2946 of *LNCS*, pages 137–196. Springer-Verlag, 2001.
- [8] A. Schaad and J. Moffett. Delegation of Obligations. In *Proc. of POLICY'02*, pages 25–35. IEEE Press, 2002.
- [9] United States Code. Sarbanes-oxley act of 2002, pl 107-204, 116 stat 745. Codified in Sections 11, 15, 18, 28, and 29 USC, July 2002.
- [10] Jacques Wainer, Paulo Barthelmess, and Akhil Kumar. W-RBAC: A workflow security model incorporating controlled overriding of constraints. *International Journal of Cooperative Information Systems*, 12(4):455–485, 2003.