available at www.sciencedirect.com

ScienceDirect

www.compseconline.com/publications/prodinf.htm

Information Security Technical Report

Analysis of recommended cloud security controls to validate OpenPMF "policy as a service"

Ulrich Lang [a,b,*], *Rudolf Schreiner* [a,b]

[a] *ObjectSecurity, Cambridge, UK*
[b] *ObjectSecurity, Palo Alto, CA, USA*

Keywords:
Cloud
Security
Policy
Authorization management
Access policy
Compliance
Model-driven security
Accreditation
Audit policy
Application security
XACML
OpenPMF
NIST 800-53
NIST 800-147
NIST IR 7628
PCI-DSS
HIPAA

ABSTRACT

This paper describes some of the findings of a cloud research project the authors carried out in Q2/2011. As part of the project, the authors first identified security concerns related to cloud computing, and gaps in cloud-related standards/regulations. The authors then identified several hard-to-implement, but highly cloud-relevant, security requirements in numerous cloud (and non-cloud) regulations and guidance documents, especially related to "least privilege", "information flow control", and "incident monitoring/auditing/analysis". Further study revealed that there are significant cloud technology gaps in cloud (and non-cloud) platforms, which make it difficult to effectively implement those security policy requirements. The project concluded that model-driven security policy automation offered as a cloud service and tied into the protected cloud platform is ideally suited to achieve correct, consistent, low-effort/cost policy implementation for cloud applications.

1. Introduction

A current trend in IT is a shift toward cloud computing as the platform for next generation IT infrastructure & applications, with a forecasted very large and rapidly growing market opportunity (Gartner, 2010). However, security and compliance concerns are typically the top factor for slowing down cloud adoption (Portio Research for Colt Telecom Group, 2009), and there is significant debate in the IT industry about whether traditional security is sufficient for cloud or not.

Implementing business/people-centric security policies for clouds can be complex for a number of reasons. It is often particularly complex (Lang, 2011a) for interconnected, dynamically changing application landscapes with cross-organization information flows, such as cloud application mashups, Service Oriented Architectures (SOAs), and other agile application architectures. Such application environments are adopted for various business reasons, and security needs to support those at the lowest overall cost possible. Automation is therefore key.

Furthermore, security solutions must meet the specific security needs for cloud computing, without impacting the Return on Investment (ROI) business rationale for cloud adoption (esp. increased operational expenditure and minimized

* *Corresponding author*. ObjectSecurity, Cambridge, UK.
E-mail addresses: ulrich.lang@objectsecurity.com (U. Lang), rudolf.schreiner@objectsecurity.com (R. Schreiner).
1363-4127/$ – see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.istr.2011.08.001

upfront capital expenditure). An automation approach security policy implementation is particularly important for cloud computing because cloud subscribers will demand support for regulatory compliance policy management from cloud providers, but will at the same time judge the financial ROI by the same measures as they do for cloud computing in general, i.e. by how much it cuts their upfront capital expenditure and their in-house manual maintenance cost.

The paper is based on the results of the authors' research project "Feasibility Analysis: Cloud Computing Application Security Policy Automation as a Service: Gaps and Solutions" (Lang and Schreiner, unpublished, 2011). The goal of the project is to validate that the market potential is significant for a suitable structured cloud-based security policy automation service that helps implement advanced security policy requirements for clouds (and non-cloud environments) correctly, effectively, and manageable, and in line with the cloud computing business model. Such a cloud "policy as a service" can be implemented using the authors' OpenPMF model-driven security (MDS) policy automation technology (esp. for authorization and incident monitoring). The project involved:

1. *Problem analysis*: Analyzing security and compliance concerns and the need for a novel cloud service that automates security policy and compliance implementation;
2. *Peer review & advocacy*: Involving the community to ensure the accuracy of assumptions and results, as well as to educate and advocate the wider community about the results;
3. *Solutions design*: Establishing the feasibility of such a novel cloud service that automates security policy and compliance implementation for cloud applications ("policy as a service"). A design study also showed that ObjectSecurity's OpenPMF product (ObjectSecurity, 2011a, Ritter et al., 2006) and its model-driven security approach (Lang, 2011a,b) are well suited for automating technical security policy implementation.

This paper mainly focuses on first task, the results of the problem analysis and the validation that the solution is required. For the second task, the authors are providing ongoing feedback and contributions about the identified recommendation to the various organizations (e.g. NIST 800-146, NIST Cloud Computing Security Working Group – NCC-SWG, ENISA, FedRAMP, UK G-Cloud Cabinet Office, CSA), attended numerous related meetings and online discussion groups, produced several publicly available scientific publications (such as this one), and blogged (Lang, 2011a,b) about the results. The third task, the actual solution design and demonstrator, is not the main focus of this paper and described in detail in (Lang, 2011b).

2. Analysis objectives

The first part of this project involved understanding the exact challenges around cloud security and compliance, in particular to answer the following questions:

- *Unique concerns*: Are there any unique cloud security and compliance concerns, vulnerabilities and weaknesses that are specific to cloud computing (i.e. significantly different from non-cloud computing architectures), and what are the causes of those unique cloud security concerns? And how hard is it to implement solutions for those concerns?
- *General concerns*: Which general (i.e. not unique to cloud) security and compliance concerns apply to cloud computing? And how hard is it to implement solutions for those concerns?
- *Guidance gaps*: Are there any gaps (e.g. missing, irrelevant, and not-quite-right controls) in current cloud-related security and compliance regulations, guidelines, policies, standards[1], e.g. by National Institute of Standards and Technology (NIST), International Standards Organization (ISO), European Network and Information Security Agency (ENISA), and Cloud Security Alliance (CSA)?
- *Technology gaps*: Are there any gaps in current technologies implementing the recommendations for cloud?
- *Required features and drivers*: Which specific cloud business, technical, and market issues does a security policy automation cloud service have to address in order to be successful, as compared to issues found in traditional deployments (e.g. for Service Oriented Architectures, SOA)?

To analyze whether the findings of this project are of a cloud specific nature or of a general nature, additional general (i.e. not cloud-specific) cyber security guidance was analyzed for comparison purposes, and the conclusion was that the same problem occurs wide-spread across many guidance documents. In particular, the uses reference examples from the following analyzed guidance[2]:

- NIST 800-53 (general cyber security) (NIST, 2009);
- NIST 800-146 (cloud) (NIST, 2011a);
- ENISA (cloud security) (ENISA, 2009);
- PCI-DSS (payment card security) (PCI Standards Council, 2010);
- NIST IR-7628 (smart grid security) (NIST, 2010);
- HIPAA (healthcare security regulation) (US department of Health and Human Service, 2011a,b)
- FedRAMP (US Government CIO Council, 2010)
- CSA Guidance Document v2.1 (CSA, 2009) and CSA Cloud Controls Matrix (CSA, 2010)

FedRAMP and CSA Cloud Control Matrix are not specifically mentioned in any of the examples throughout this paper because both do not actually provide new guidance, but rather select and structure existing other guidance (esp. NIST 800.53).

Interestingly, some cloud architectures do not reveal their security architectures (e.g. the UK government G-Cloud (UK Cabinet Office, 2009) and could therefore not be analyzed). In the authors' view, the concept of "security through obscurity" (Wikipedia, 2011) is generally not advisable[3] in an emerging

[1] For the sake of simplicity, we will subsume all these different kinds of documents into the term "guidance" unless there is a reason to differentiate between their normative/de-facto/legislative etc. natures.

[2] The government relevant Common Criteria standard has been analyzed in Lang and Schreiner (2009).

[3] The UK G-Cloud information assurance document is classified. The UK government responded to the author's blog post on www.modeldrivensecurity.org and subsequent contact request that this "security through obscurity" approach was chosen due to national security reasons.

technology space where the contributions of the wider community should be used as valuable input.

Due to space restrictions this article of course cannot discuss every analyzed control in all covered guidance. Instead, the remainder of this subsection will discuss the concrete findings of several exemplary guidance and particularly relevant controls.

3. Key project findings

3.1. *More details in cloud-related guidance is needed in order to push for effective implementation, esp. of authorization and access control*

All analyzed cloud relevant guidance lacks sufficient controls or control detail related to authorization policy creation, update, management, enforcement, and monitoring. This is not only due to the general lack of technical implementation guidance in most documents, as shown by the fact that most documents for example specify controls in much more depth (e.g. identity and authentication related – see discussion below).

That lack of concrete guidance causes the real-world problem that cloud providers and subscribers today choose to not effectively implement any controls to mitigate these vulnerabilities, and often get away with it from a regulatory perspective until breaches occur.

The analyzed guidance around controls related to IT security compliance and auditing is also specified in insufficient depth to be effective, e.g. what exactly to monitor and log, and how to identify non-compliance in a manageable way. The guidance also does not state that policy-driven analysis is needed.

All cloud security guidance should explicitly state that typically recommended identity and authorization mechanisms are only useful if subscribers can actually author and maintain policies and get meaningful visibility, which in a cloud environment is unclear for most of the mechanisms you list. It is important that clouds will likely evolve to interconnected, business process workflow orchestrated application mashups, and authorization policy automation for those mashups ("system of systems") must also be controllable by subscribers, and not just by providers. In particular, all standards and guidance should call for (1) preventive, dynamic least privilege enforcement, (2) information flow control enforcement, and (3) policy-driven monitoring, analysis and auditing (each is discussed below).

Technologies such as authorization management and model-driven security (MDS) policy automation are available today, but standards and guidance documents do not sufficiently push providers and subscribers to implement effective technology solutions.

3.1.1. *Analyzed examples*

(1) NIST SP 800-53 "Recommended Security Controls for Federal Information Systems and Organizations" (NIST, 2009) laudably covers access controls as the first set of controls. It mentions application level access enforcement (AC3), but misses implementation guidance or examples of what features to actually concretely implement. It also does not explain in sufficient depth that access policies should be implemented to be manageable (and especially does not mention ZBAC or model-driven security anywhere throughout the document). In addition to the least privilege and information flow control sections covered individually below, the "separation of duties" control (AC5) omits the most complex use case of dynamic policies (e.g. "Chinese Wall"), and gives no implementation advice or examples.

(2) NIST SP 800-146[4] "Cloud Computing Synopsis and Recommendations" (NIST, 2011a) mentions access policies in the Identity and Access Management (IAM) section, but the document does not specifically mention authorization policies, and does not mention that dynamic application interactions need to be controlled as well as the mentioned user interactions (see discussion below). Also, the list of access control standards and concepts/mechanisms is not fully correct and complete (in the draft version), and puts generic concepts and specific technical standards in the same list. Specifically, both model-driven security and ZBAC are missing.

(3) ENISA's "Cloud Computing – Benefits, risks and recommendations for information security" document (ENISA, 2009) does call for "scalable security management (policy and operating procedures) within cloud platforms: automatic enforcement of security and data protection policies", but leaves out any details. It misses an in-depth discussion of authorization throughout the document. Access control guidance is restricted to user accounts and roles and do not mention anything about how to implement those controls (e.g. using authorization management technologies), except once in the appendix (where XACML (OASIS, 2005) is mentioned). The document should state in the main part that fine-grained, contextual authorization management with a high degree of policy management automation (through model-driven security) is required. Also does not mention ZBAC. It will be absolutely critical for clouds that policies can be determined by the subscriber, not only by the provider. The mentioning of least privilege is covered individually below. Finally, since policy management can be viewed as part of the wider area of configuration management, the covered vulnerability "misconfiguration" is too vague and should specifically include incorrect authentication, authorization, and accounting policies.

3.2. *Proportionally weighted focus in guidance is needed; instead, guidance is overly focused on identity & authentication and neglegts on policy and authorization*

Inherently, identities and authentication are worthless without authorization and access control. For cloud computing to work at large scale and interconnectedness ("cloud mashups"), fine-grained contextual authorization management with a high degree of policy management automation (e.g. through model-driven security) is required.

[4] The authors have provided feedback to NIST accordingly as part of the gap analysis project.

But, as already briefly mentioned, most of the analyzed standards and guidance documents discuss identity & authentication in depth, while only peripherally touching policy and authorization.

Some guidance even interchangeably refers to the concepts of authorization and policy and the concepts of authentication and identity – the authors see this problem frequently in today's information security field. In line with that identity-centric view to security policy management, virtually all guidance around authorization also overly focuses on role-based access control (RBAC) (Ferraiolo and Kuhn, 1992), and only peripherally discusses attribute based access control (ABAC) (Karp et al., 2009), which enables the enforcement of fine-grained, contextual policies, and authorization based access control (ZBAC) (Karp et al., 2009), which simplifies federated authorization management and controlled service chaining with delegation. These more state-of-the-art authorization mechanisms are especially critical for clouds where applications are composed from interconnected cloud services and cloud-hosted application modules.

3.2.1. *Analyzed examples*

(1) NIST SP 800-53 is the only notable exception analyzed, where the discrepancy of depth of coverage between the two areas is still in favor of identity & authentication, but less than in other guidance. In stark contrast to too-vague and incomplete authorization controls, NIST 800-53 covers very specific technical control related to user account authentication (AC7-15) – for example, controls for unsuccessful login attempts (AC7). This stark contrast illustrates that the level of depth is not deliberately chosen but differs between the covered controls – which can be misleading. Also it would be useful to have concrete implementation examples for all covered controls.

(2) NIST SP 800-146 focuses significantly more on identity than on access. General statements such as "the subscriber's organization is often fully responsible for managing the accounts …" imply that policy management needs to be done. However, the document does not mention that policy management has to be done both for users and applications, not just for user accounts. Account management is only one part of policy management, and the document does not give equal weight to authorization and access policies for applications and users. The document also specifically mentions identity management as a technical control, but omits the numerous above-mentioned access control methods.

(3) ENISA's cloud computing risk assessment, along the same lines, focuses (throughout the document) more on the management of identity and authentication systems, but not on the management and maintenance of authorization management systems. For example, the document states that "besides authentication, accounting and audit, what other controls are in place to prevent (or minimize the impact of) malicious activities by insiders? " – it is unclear why authorization was omitted here. Another example is that the section on "access control and audit trails" talks about authentication, but not about authorization. Yet another example is the described vulnerability "authentication, authorization, accountability", which overly focuses on authentication, roles, and passwords, and not enough about authorization/access control policy enforcement. Each point should be discussed separately (authentication, authorization, accounting), and specifically ensure that the authorization part is correct (i.e. not talk very much about role assignments, passwords etc., but much more about how the access policies are authored, distributed, enforced, monitored, and maintained across the cloud). It is also somewhat misleading that the document singles out the use of a detailed technical control ("efficient role-based access control and federated identity management solutions by default"), but does neither discuss more state-of the art alternative mechanisms (esp. ABAC and ZBAC with model-driven security), nor mention real-world implementation and scalability issues around RBAC and federated identity management solutions. It also does not mention the need for efficient and effective security policy automation (e.g. using model-driven security).

3.3. *Implementing "least privilege" control requirements widely recommended but typically not implemented effectively*

All analyzed guidance recommends the implementation of least privilege ("need-to-know", "minimum necessary"), i.e. that access policies need to be enforced contextual by the job (not job role!) – so for example, if someone ("Alice") needs access to some customer's ("Bob") payment information for the job of charging Bob, the technical access policy implementation needs to make sure that Alice is actually involved in a sales transaction related to Bob, and that Alice is at the "charge the customer" step in the sales business process. This is called "context". It is important to understand that Alice should not have blanket access to all customers' payment data because she might potentially have a transaction with any customer when they call at some unknown point in the future and buy something. In that case, least privilege would not be fully implemented.

This example makes clear that RBAC with user account management are an insufficient technical mechanism to implement the guidance, because they are not contextual enough to only grant access when needed for the particular use.

Instead, to technically implement least privilege access based on the minimum necessary for the particular granted use, disclosure, or request, technical access control must be fine-grained and contextual (e.g. based on the context of the access, the business process the requester of information is in, the way information is aggregated across interconnected IT systems etc.). Therefore, fine-grained, contextual authorization management is needed to enforce such complex policies.

What this specifically means is that a dynamic access control "whitelist" (i.e. stating what is allowed, vs. "blacklists" that state what is not allowed) needs to be available that enforces the that policy requirement. Static access control models such as identity-based access control (IBAC) or role-

based access control (RBAC) (Ferraiolo and Kuhn, 1992) are not sufficient access mechanisms because they do not capture such context in the policy (Lang, 2011a,b, Lang, 2010a). As a result, virtually all IBAC/RBAC implementations, including traditional Identity and Access Management (IAM) technologies, are insufficient on their own. Attribute-based access control (ABAC), as for example standardized in XACML (OASIS, 2005), help add this missing context and other additional expressions to the policy. The challenge with authorization management is that policies are hard to author and maintain – there are simply too many technical rules, and maintaining those is too time-consuming, expensive, difficult, and error-prone. Also these technical rules will often not directly translate from the human thinking about business security policies. To solve that policy authoring and maintenance show-stopper, model-driven security (MDS) policy automation is also needed, which automatically generates technical security rules from generic security policy requirements (models) – for example captured in models close to the understanding of the recommended guidance controls. This paper will describe further below how model-driven security can solve the unmanageability problem of ABAC.

One of the most prominent examples where least privilege policy enforcement would have had the potential to avert disaster is the recent Wikileaks leak (Guardian, 2010).

3.3.1. *Analyzed examples*

(1) NIST 800-53: The "least privilege" control (AC6) misses dynamic contextual policy complexities: as explained above, to be really effective, least privilege control implementations should only grant access based on the privilege required in a certain context (e.g. at a specific step in a business process, or at a specific time, or if something else has happened before).
(2) ENISA: Similar to the lack of detail in NIST's coverage of "least privilege", the identified vulnerability "need-to-know principle is not applied" is too generic. It should specifically state that technical mechanisms need to be in place to enforce need-to-know policies, esp. through fine-grained, contextual authorization management with a high degree of policy management automation through model-driven security.
(3) PCI-DSS states that "restricting access is crucial!", and more concretely: "Restrict access to cardholder data by business need to know: To ensure critical data can only be accessed by authorized personnel, systems and processes must be in place to limit access to system components and cardholder data based on need to know and according to job responsibilities. It explicitly explains that need to know is when access rights are granted to only the least amount of data and privileges needed to perform a job, i.e. only to those individuals whose job requires such access, and is set to "deny all" unless specifically allowed.
(4) HIPAA: The Health Insurance Portability and Accountability Act (HIPAA) "Privacy Rule" (US Department of Health and Human Service, 2011a) establishes regulations for the use and disclosure of Protected Health Information (PHI), in particular it requests the implementation of least privilege: "A central aspect of the Privacy Rule is the principle of "minimum necessary" use and disclosure". A covered entity must make reasonable efforts to use, disclose, and request only the minimum amount of protected health information needed to accomplish the intended purpose of the use, disclosure, or request. A covered entity must develop and implement policies and procedures to reasonably limit uses and disclosures to the minimum necessary, i.e. a covered entity may not use, disclose, or request the entire medical record for a particular purpose, unless it can specifically justify the whole record as the amount reasonably needed for the purpose." The "HIPAA Security Rule" (US Department of Health and Human Service, 2011b) also limits uses and disclosures of PHI to the "minimum necessary, "the Security Rule's administrative safeguards section requires a "covered entity" to implement and periodically assess policies and procedures for authorizing access to e-PHI only when such access is appropriate. Interestingly this administrative (i.e. non-technical) section specifically states that this should be implemented "based on the user or recipient's role (role-based access)", which is insufficient. The technical safeguards section mandates access control "A covered entity must implement technical policies and procedures that allow only authorized persons to access electronic protected health information (e-PHI), and must "record and examine access and other activity in information systems that contain or use e-PHI."
(5) NIST IR 7628: The recommended control "Least Privilege" requires that "the organization assigns the most restrictive set of rights and privileges or access needed by users for the performance of specified tasks", and that "the organization configures the Smart Grid information system to enforce the most restrictive set of rights and privileges or access needed by users"[5]. In other words, a caller should only be granted access to a resource if that caller has a need to do so in the specific context, for example a particular step in a business process, or a particular system situation such as emergency level.

3.4. *Implementing "information flow enforcement" access control requirements widely recommended but not effectively implemented in practice*

Information flow enforcement is a critical cloud security requirement because, as mentioned earlier, interconnected, agile application mashups (potentially across trust domains) will play a big role for clouds, and authorizations/access policies for those mashups must be controllable by subscribers.

[5] The section of the document also offers the additional considerations relevant to the discussion of this paper, in particular that "the organization authorizes network access to organization-defined privileged commands only for compelling operational needs and documents the rationale for such access in the security plan for the Smart Grid information system", and "the organization authorizes access to organization-defined list of security functions (deployed in hardware, software, and firmware) and security-relevant information".

As already mentioned in the previous subsection, IBAC and RBAC are insufficient on their own, and due to the dynamic changing ("agile") nature of today's interconnected IT landscapes ("system of systems"), ABAC policies would need to be constantly manually updated to be correct after "system of systems" changes, resulting in a policy management nightmare (Lang, 2010b). There are a number of other problems with ABAC, e.g. challenges around authorization delegation across service chains and impersonation, which can be solved using ZBAC (Karp et al., 2009), which uses authorization tokens and federated authorization token servers.

This paper will describe further below how model-driven security can solve the unmanageability problem of ABAC and ZBAC.

3.4.1. *Analyzed examples*

(1) NIST 800-53: The mentioning of "flow control" (AC4) is overly simplistic in scope and does not mention policies based on workflows (e.g. application interactions, BPM orchestration or cloud mashups). It also omits the challenges related to dynamic (agile) changes.
(2) HIPAA: Mandated technical safeguards state that "a covered entity must implement technical security measures that guard against unauthorized access to electronic personal health information (e-PHI) that is being transmitted over an electronic network" (US Department of Health and Human Service, 2011b). This statement is too vague, for example does it not even state to what level of assurance these measures have to be implemented. But since the ultimate goal is to prevent breaches and the associated damage and cost associated with them, it is clear that implementing this as best as possible is desirable (e.g. using model-driven security policy automation).
(3) NIST IR 7628: The recommended control "Information Flow Enforcement" requires that "the Smart Grid information system enforces assigned authorizations for controlling the flow of information within the Smart Grid information system and between interconnected Smart Grid information systems in accordance with applicable policy. Information flow control regulates where information is allowed to travel within a Smart Grid information system and between Smart Grid information systems. As example implementations, the document mentions boundary protection devices that restrict Smart Grid information system services or provide a packet-filtering capability. This section of the document also offers a number of supplemental considerations. Particularly interesting for the discussion in this paper, the guidance recommends "dynamic information flow control allowing or disallowing information flows based on changing conditions or operational considerations"[6].

[6] Other considerations not directly related to the discussion in this paper include enforcement based on explicit labels on information, source, and destination objects, organization-defined security policy filters, human policy filter review, privileged administrator policy filter configuration.

3.5. *"Security incident monitoring", "reporting", "auditing" implementations need to become more policy-driven and automated to improve near real-time visibility into security posture*

In the context of the fine-grained contextual authorization mentioned earlier, incident monitoring, reporting, and audit are intrinsically intertwined with authorization. Monitoring, reporting, and audit tools will need to know the specific authorization policies in order to decide whether behavior is in fact suspicious or not, and what the criticality is. This differs dramatically from traditional network monitoring approaches which mainly monitor for generic vulnerabilities (i.e. the same vulnerabilities occur for a particular technology, rather than for a particular business) and thus do not need to know any specifics about the organization's business processes in order to flag an incident. The authors call control and visibility for generic vulnerabilities "security hygiene" to distinguish them from organization-specific policy enforcement and monitoring. This paper will describe further below how model-driven compliance can solve the policy-driven monitoring challenge for authorization management.

3.5.1. *Analyzed examples*

(1) NIST IR 7628: numerous recommendations for incident monitoring, incident reporting, and auditing are spread throughout the document, recommending that "the organization monitors events … to detect attacks, unauthorized activities or conditions, and non-malicious errors" based on the organization's policy requirements ("monitoring objectives"). In addition to recommending such policy-driven monitoring, it suggests the use of automated incident analysis and reporting tools mechanisms. It also states that there needs to be a maintained list of auditable events, and again suggests automated, centralized analysis tools and near real-time analysis and after-the-fact investigations of security incidents, e.g. by automatically processing audit records for events of interest based on selectable event criteria. A more detailed analysis can be found in (Lang and Schreiner, 2011).
(2) None of the analyzed guidance documents mention that there is a need for visibility related technologies (including intrusion detection/prevention, monitoring, and audit analysis) to become more policy-driven. This is because without a policy-driven approach human security personnel are flooded with too much incident and audit log information to manually analyze fast enough to find actual security breaches (too many false positives, and a high chance of false negatives). For the same reasons, compliance audits (and assurance accreditation) become expensive, time-consuming, and error-prone. In contrast to that, if security is policy-driven and automated (using security policy automation and preventive whitelisting as described in the previous section), an automated system can automatically sieve through the logs and figure out automatically which policies have been violated (including unsuccessful attempts). For example, in the ENISA cloud computing risk assessment document section about "audit

reduction & report generation" (AU7) omits that this process will often have to be policy-driven.

3.6. *Cloud subscribers need more standardized control over internal cloud security*

The dominant security standards/guidelines perspective is that of the cloud subscriber. However, cloud subscribers are not sufficiently able to control their organization's particular security policy implementation at the higher cloud platform stacks (Platform and Software as a Service, PaaS, SaaS) in a standardized and manageable way. No standards for feeding user policy into cloud platforms are currently available or underway, which is a serious limitation of cloud[7].

Technical authorization policy standards such as XACML can play a part role as a machine-to-machine policy exchange format, but must be hidden from the user because they are unmanageable for humans (Lang, 2011a,b). To achieve this, model-driven security approaches (ideally based on standards such as Eclipse EMF or OMG MOF) let user organizations subscribe to (or capture themselves) policy in human-understandable terms and automatically turn them into specific technical implementation across cloud platforms. Both use cases (subscribe, capture) can be implemented using ObjectSecurity OpenPMF policy as a service.

Government cloud initiatives should use their clout and drive standardization of manageable cloud policy mechanisms, partly because some cloud systems (esp. for government use) could at some point be accredited by rigorous information assurance standards such as Common Criteria or NIAP, which will require a more standardized, formalized approach to policy (among other things). Model-driven security policy automation can help achieve this.

3.6.1. *Analyzed examples*

(1) NIST SP 800-146 mentions the concerns about control and visibility for subscribers over their cloud services and calls for standards and subscriber-understandable policy management mechanisms that "can be integrated with existing enterprise/agency security frameworks such as identification and authorization so that enterprise/agency security policies can be enforced."

3.7. *Cloud subscribers need more standardized visibility over internal cloud security*

Cloud providers only grudgingly provide insight into and control over internal cloud security. While some information is available on cloud provider websites (e.g. (Amazon, 2011)) there is not nearly enough to allow users to solve their organization/business specific security challenges. None of the analyzed guidance documents specifically mention the need for incident monitoring and audit logging standards. However, some standards are emerging that enable end users to request compliance reports from cloud providers (e.g. Cloudaudit.org (Cloud, 2011a)) and to process alert logs (e.g. SCAP (NIST, 2011b)) but it will take some time until these will be made widely available by cloud providers by default.

As also mentioned above, government cloud initiatives should use their clout and drive standardization of manageable automated cloud compliance and accreditation mechanisms, partly because some cloud systems could at some point be accredited by Common Criteria or NIAP, which will require a more standardized, formalized approach to compliance (among other things). Model-driven compliance automation in conjunction with model-driven security policy automation can help achieve this.

3.8. *Application layer security and business process layer security needed, but guidance does not cover this*

To be able to tackle today's security policy and compliance requirements, the emerging focus of acceptable and enforceable cloud security policy standards/guidelines has to include more application and business process security controls. For example, there is a critical need in many cases to enforce complex, contextual policies (e.g. least privilege) where an authorization decision depends on more than just the authenticated identity (or role) of the requester: the decision also depends on the business process and the context of the application whether access should be granted. For example, "doctors should only access the patient record of the patient they are currently treating", "billing should only be able to charge treatment to patients that have been treated by the doctor", "credit card data should only be collected if needed, only be used for the purpose collected, and deleted when the purpose is completed".

3.8.1. *Analyzed examples*

None of the guidance covers this explicitly.

3.9. *Cloud Platform as a Service (PaaS) application development tools that support secure development and policy/compliance automation are needed, but most guidance does not cover this enough*

While there is a school of thoughts that advocates more and better security training for developers, this is – judging from the authors' observations in numerous projects – not practical in most real-world situations because developers are usually under significant time pressure to produce the functional parts of the application. Security would just unnecessarily slow down the development process. The authors advocate a different school of thoughts where security is built into application development (and mashup) tools to guide developers toward more secure applications – and to prevent them from developing less secure software. The authors specifically advocate a model-driven approach for automating both the generation of technical security policies and the analysis of compliance/accreditation evidence.

3.9.1. *Analyzed examples*

(1) NIST 800-146 states that application development tools should mitigate security vulnerabilities, but does not state

[7] The authors are advocating the formation of an interest group within CSA to fill that gap using "model-driven security policy automation".

any details, or that the development tools also need to support the intuitive authoring and maintenance of security policies by subscribers. Model-driven security policy automation fits directly into such development frameworks (and mashup tools) to achieve this.
(2) NIST 800-53 does not itself cover this topic but instead peripherally refers to NIST SP 800-64, which is very generic and high-level.
(3) ENISA's cloud computing risk assessment does not specifically cover this topic either.

3.10. *"Policy as a Service" is very attractive for large, federated private/community clouds*

The described cloud-based policy service for cloud is widely applicable across many types of cloud deployments models and across many sectors, e.g. government, finance, utilities, healthcare, transportation, and commercial/internet. We identified that (1) the value proposition is most attractive when Policy as a Service is deployed internally for a private or hybrid cloud environment – this is mainly because those environments are generally more security-critical than public cloud deployments, and because most organizations would not be prepared to outsource their security policy to a third-party public cloud service; (2) manual fine-grained, contextual authorization management becomes more unmanageable the larger the IT landscape gets (3) compliance evidence creation for reporting/auditing becomes unmanageable in a cloud environment, and especially the larger and more interconnected the IT landscape gets. The studied cloud "policy as a service" is therefore particularly relevant for large, federated, highly regulated organizations such as governments (e.g. "G-Cloud" initiatives in the UK, US, and elsewhere). Governments should use their cloud initiatives to drive innovation in cloud security, and to lead by example by being an early adopter (G-Cloud, 2011).

3.11. *"Policy as a service" is easier to adopt for clouds, esp. Platform as a Service (PaaS), than for more traditional deployments, e.g. Service Oriented Architecture (SOA)*

PaaS is usually based on one (or a few different) technology platform (e.g. web app server, development & mashup tools) and configuration that is maintained by the provider and that scales homogeneously across all PaaS subscribers. Therefore the "policy as a service" as per the authors' OpenPMF implementation only needs to be integrated with that homogeneous platform. This is much more straightforward and cost-effective than the integration challenges around multi-platform SOAs (esp. with additional legacy integration).

3.12. *"Policy as a Service" will become increasingly attractive as clouds will evolve and become more interconnected, BPM mashup orchestrated*

As explained earlier, cloud computing will evolve to include flexible mashups (e.g. agile BPM workflow oriented integration) of individual application service modules ("system of systems"). These models can be installed onto a cloud platform from a cloud hosted "app store", as proposed for example by various government cloud initiatives. Clouds will also evolve to include legacy platform integration, support for vertical industry certifications, and flexible, more transparent (tiered) cost models. This is sometimes referred to as "Cloud 2.0" (Cloud, 2011b), to distinguish from currently widely used cloud "Cloud 1.0" services that include hosted email, CRM, office productivity applications, and IaaS.

For most security-critical organizations, the dynamic interactions between the various interconnected services and application modules will need to be controlled (esp. authorization, authentication, and auditing), because sensitive information flows across trust boundaries will occur. As mentioned above, manually authoring and maintaining technical authorization rules is too expensive, error-prone, repetitive, and time-consuming. Model-driven security policy automation can reliably automate much of this task so that authorizations are always up-to-date. It also shortens assurance accreditation time compared to manual ad-hoc policy authoring, because the model-driven security algorithms can be accredited once as correct and re-run after changes, rather than having to reaccredit a system of systems after changes (Lang and Schreiner, 2009).

3.13. *"Policy as a service" is more likely to be adopted when hosted as a private or dedicated cloud service*

Policy as a service is not likely to be a viable business model for high assurance cloud environments (e.g. for critical industries and government) if hosted on – or offered as – a public cloud service. This is mainly due to the inherent control, visibility, confidentiality, reliability, and availability concerns that go with using any public cloud service.

While subscribers with less high assurance requirements could make use of publicly hosted policy as a service as a cost-effective way to manage policy, those subscribers will naturally often not have a strong enough business driver for security to adopt security policy automation in the first place.

For subscribers with higher assurance requirements, policy as a service is likely to be hosted as a private or dedicated outsourced cloud service instead. The following section describes the authors' demonstrator environment for such a dedicated cloud service hosted at ObjectSecurity.

4. Solution design overview: OpenPMF security policy & compliance automation

The various parts of the described technical architecture, including policy automation (control) and incident analysis automation (visibility), are currently implemented as a "policy as a service" demonstrator by the authors for clouds using ObjectSecurity OpenPMF (ObjectSecurity, 2011a).

The OpenPMF policy as a cloud service business model (see (Lang, 2010b) for details) was designed to help cloud subscribers implement control and visibility for cloud with as little effort, cost and error-potential as possible (e.g. using model-driven security policy automation as a service). Policies are captured in models at the OpenPMF policy service by human experts (either in-house or by the external policy

service provider if hosted externally). The service is accessed whenever a set of policies need to be implemented, and is charged per use (e.g. monthly subscription fee).

The technical architecture was validated by implementing a demonstrator (described in more detail in (Lang, 2010c)) hosted in-house at ObjectSecurity, and using the following technology stack: Linux, KVM, Tomcat/AXIS2, OpenPMF within Eclipse, Intalio BPMS within Eclipse, SSL/TLS encryption. The OpenPMF policy automation features are embedded into the Intalio BPMS web service orchestration tool. When the context menu "OpenPMF->Generate Security Policy" is clicked, the detailed technical security rules for the specific application are automatically generated.

The technical architecture of OpenPMF policy as a cloud service involved the following two areas:

1) *Improving control*: Model-driven security (MDS) policy automation (Lang, 2011a,b; Lang, 2010a, 2010b, 2010c; Lang and Schreiner, 2009) (as implemented using OpenPMF) is a potential solution to the control challenge: MDS adds the required level of security policy automation by applying the reasoning behind model-driven software development approaches to security and compliance policy management. In essence, model-driven security can automatically generate technical authorization (and other) rules by analyzing the application with all its interactions, and applying generic security requirements to it. Model-driven security is a tool-supported process that involves modeling security requirements at a high level of abstraction (in models/meta-data), and using other information sources available about the system, especially the applications' functional models (produced by other stakeholders), to automatically generate fine-grained, contextual technical authorization (and other) rules. The inputs into model-driven security are expressed in Domain Specific Languages (DSL), using generic modeling languages (such as, Unified Modeling Language-UML). Model-driven security has a number of benefits: Compared to manual policy authoring/maintenance, it reduces manual administration overheads and saves cost/time through automation (policy generation, enforcement, monitoring, update) – especially for agile software applications. It also reduces security risks and increases assurance by minimizing human error potential, and by ensuring that the security implementation is always in line with business requirements and with the functional behavior of the system, thus improving both security and safety of the system. Furthermore, it unites policy consistently across security silos (e.g. different application runtime platforms). Finally, it forms part of a more automated model-driven approach to agile incident analysis and compliance.
2) *Improving visibility*: Model-driven incident monitoring, analysis, and auditing automation (based on the authors' "model-driven security accreditation" (MDSA) (Lang and Schreiner, 2009) concept, includes incident monitoring, analysis, and auditing, improves near real-time visibility by collecting application layer alerts and mapping those back to the human-understandable security requirements to be able to assess the security posture on an ongoing basis.

Model-driven security approaches sometimes still gets challenged because of its dependence on application specifications, which are not always readily available in traditional deployments. However, modeling aspects of the interconnected system (esp. interactions) is an important part of state-of-the-art Cloud PaaS and mashups, and is also part of robust systems design. Also, modeling systems at the right granularity (e.g. orchestration) does not actually add to the total cost of policy management. This is because if security administrators have to manually specify detailed technical security rules because their tools do not support model-driven security, they are effectively specifying the security related aspects of the application specification within their policy administration tool. In practice, this is impossible for non-trivial systems, esp. over the whole system life cycle. Model-driven security simply re-uses this information (which often make up the greater part of security policy rules) from models (e.g. BPM orchestration) specified by specialists (and/or tools) who understand applications and workflows better anyway (i.e. application developers/integrators, and process modelers). This argument is supported by the author's practical experiences that, even after only a short while in operation, model-driven security can greatly reduce costs of effort of protecting the system and improve security and safety compared to traditional, manual policy definition and management.

5. Related work

While "security as a service" (SecaaS) is a growing market, there are (to the authors' knowledge[8]) no directly related "security as a service" technologies for model-driven security policy automation, security compliance automation, or authorization management in the market today.

Identity and Access Management (IAM) is a task that is particularly closely related to policy management and application security[9].While frequently viewed as having more to do with user identity management than with application security, IAM is actually also often highly related to application layer security. This is because when users access applications and authenticate, an authorization policy needs to be enforced, which often depends heavily on the particular accessed application. At that point the hard questions "where does this authorization policy come from? who writes and maintains that policy? how is it enforced? how is it audited?" arise. These issues make the case for also using model-driven security policy automation alongside IAM to make policy management less time-consuming, repetitive, expensive, and error-prone. Unfortunately in many cases security policy automation is easier said than done, and many security tools today offer automation at the price of trading off relevance,

[8] This is based on one of the authors' work as a co-lead for the CSA Security as a Service (SecaaS) working group.

[9] In general, "application layer security" is a lot broader than the policy automation aspects covered in this article, and also includes tasks such as vulnerability scanning, application layer firewalls, configuration management, alert monitoring and analysis, and source code analysis.

correctness and automation (Lang, 2010b). Often the reason for implementing such complex and contextual policies is because end-user organizations must comply with industry specific guidance.

Conventional "authorization management", which is nowadays often categorized as part of Identity & Access Management (IAM), illustrates these challenges: Policies become unmanageable when systems and participants get numerous, when interconnected applications evolve dynamically ("agility"), and when policies become feature-rich, fine-grained and contextual. There are simply too many, too complex technical security rules to manage, so that authorization policies can become unspecifiable or unmanageable, and the confidence in the enforced policy can be undermined. In order to support the adoption rationale behind agile application environments such as cloud and SOA, authorization management itself needs to be at least equally agile, and also automated, manageable, fine-grained, contextual. Irrespective of those shortcomings, authorization management forms a critical technical building block to enforce and audit application authorization policies for all protected resources. It plays a critical part of cloud application security, and even more so for cloud mashups, because different actors (e.g. users or cloud applications) should only be able to invoke each others' services if they are authorized to do so in a specific situation based on security policies.

6. Conclusion

In this project, we gathered a valuable analysis of gaps in numerous current cloud-related (and other) security standards and guidance documents. There are major gaps in cloud-related standards and guidance documents. In particular:

- More details in cloud-related guidance is needed in order to push for effective implementation, esp. of authorization and access control
- Proportionally weighted focus in guidance is needed; instead, guidance is overly focused on identity & authentication and neglegts on policy and authorization
- Implementing "least privilege" control requirements widely recommended but typically not implemented effectively
- Implementing "information flow enforcement" access control requirements widely recommended but not effectively implemented in practice
- "Security incident monitoring", "reporting", "auditing" implementations need to become more policy-driven and automated to improve near real-time visibility into security posture
- Cloud subscribers need more standardized control over internal cloud security
- Cloud subscribers need more standardized visibility over internal cloud security
- Application layer security and business process layer security needed, but guidance does not cover this
- Cloud Platform as a Service (PaaS) application development tools that support secure development and policy/compliance automation are needed, but most guidance does not cover this enough
- "Policy as a Service" is very attractive for large, federated private/community clouds
- "Policy as a service" is easier to adopt for clouds, esp. Platform as a Service (PaaS), than for more traditional deployments, e.g. Service Oriented Architecture (SOA)
- "Policy as a Service" will become increasingly attractive as clouds will evolve and become more interconnected, BPM mashup orchestrated
- "Policy as a service" is more likely to be adopted when hosted as a private or dedicated cloud service.

As a result of this project, we were able to validate that "Policy as a Service" based on model-driven security policy automation and model-driven incident monitoring, analysis, and auditing automation is an important solution. This enabled us to design the future direction of OpenPMF "security policy & compliance automation as a service" to meet cloud security needs. Large enterprises and the numerous emerging government cloud initiatives worldwide (e.g. UK, US, Canada and elsewhere) should have a great interest in the described service, which is currently being implemented by the authors based on ObjectSecurity's OpenPMF product. The results will directly impact the cloud roadmap of ObjectSecurity's OpenPMF product.

In addition, the authors were able to impact emerging cloud-related security and compliance standards through active contribution to include application and business process security controls.

Acknowledgments

The project described in this paper was part funded by the UK Technology Strategy Board as part of the Feasibility Studies for Digital Services micro-SME call. The project was carried out by the ObjectSecurity authors, and involved informal collaboration with John Mullen and David Chizmadia from Promia, Inc. as part of other proposals and projects (including a joint project for US Navy SPAWAR (ObjectSecurity, 2011b)).

REFERENCES

Amazon. AWS security center website, http://aws.amazon.com/security/; 2011.

CloudAudit.org website, www.cloudaudit.org; 2011a.

CloudBestPractices. "Cloud 2.0 applications", in "cloud computing – best practices guide". blog (CloudBestPractices.info); 2011b.

Cloud Security Alliance (CSA). Security Guidance for Critical Areas of Focus in Cloud Computing V2.1; December 2009.

Cloud Security Alliance (CSA). CSA Cloud Controls Matrix V1.1; December 2010.

ENISA. Cloud computing – benefits, risks and recommendations for information security; November 2009.

Ferraiolo DF, Kuhn DR. Role-based access control; October 1992. 15th National Computer Security Conference. pp. 554–563.

G-Cloud innovation website, www.gcloudinnovation.com; 2011.

Gartner. Gartner Says Worldwide Cloud Services market to surpass $68 Billion in 2010, http://www.gartner.com/it/page.jsp?id=1389313; 22 June 2010. press release.
Guardian (UK). US embassy cables leak sparks global diplomatic crisis; 28 November 2010.
Karp AH, Haury H, Davis MH. From ABAC to ZBAC: the evolution of access control models. HP Laboratories; 2009. HPL-2009-30.
U. Lang, "Authorization as a Service for Cloud & SOA Applications", Proceedings of the international workshop on cloud privacy, Security, Risk & Trust (CPSRT 2010), Collocated with 2nd IEEE International Conference on cloud computing technology and Science (Cloudcom) CPSRT 2010, Indianapolis, Indiana, USA, December 2010a.
Lang U. Security policy automation: improve cloud application security ROI. ISSA Journal; October 2010b.
Lang U. Cloud & SOA application security as a service. In: Proceedings of ISSE 2010; 5–7 October 2010c [Berlin, Germany].
Lang Ul. "Security policy automation using model driven security". Blog, www.modeldrivensecurity.org; 2011a.
Lang U. Model-driven cloud security. IBM developerWorks online publication; 02 February 2011b.
Lang U, Schreiner R. Model Driven Security Accreditation (MDSA) for Agile, interconnected IT landscapes. In: Proceedings of WISG 2009 Conference; November 2009.
Lang U, Schreiner R. Security policy automation for smart grids: manageable security & compliance at large scale. In: Proceedings of the ISSE 2011 conference; November 2011. Prague, Czech Republic.
U. Lang and R. Schreiner, "Feasibility Analysis: Cloud Computing Application Security Policy Automation as a Service: Gaps and Solutions", Feasibility Studies for Digital Services, UK Technology Strategy Board, unpublished.
NIST. Recommended security controls for federal information systems and organizations. Special Publication 800-53 Revision 3; August 2009.
NIST. NISTIR 7628-Guidelines for smart grid cyber security; August 2010.
NIST. DRAFT cloud computing synopsis and recommendations. Special Publication 800-146; May 2011a.
NIST, SCAP website, http://scap.nist.gov/; 2011b.
OASIS. Extensible Access Control Markup Language (XACML), OASIS Standard, 2.0, xml.coverpages.org/xacml.html; March 2005.
ObjectSecurity. OpenPMF whitepaper, www.openpmf.com; 2011a. online publication.
ObjectSecurity and Promia implement XML security features for next-generation US military security technology. April 2010, http://www.objectsecurity.com/doc/20100430-objectsecurity-promia-navy-soa3.pdf; 2011b.
PCI Standards Council. PCI DSS quick reference guide – understanding the payment card industry – data security standard version 2.0; October 2010.
Portio Research for Colt Telecom Group, Study. Security: the single biggest barrier to cloud adoption, http://www.colt.net/managedservices/uk/en/news-events/press-releases/security-the-single-biggest-barrier-to-cloud-computing-adoption-en.htm; 10 December 2009. online publication.
Ritter T, Schreiner R, Lang U. IEEE distributed systems online. Integrating security policies via container portable interceptors, vol. 7, no. 7. Published by the IEEE Computer Society; July 2006. art. no. 0607–o7001, 1541-4922 © 2006.
UK Cabinet Office. G-Cloud programme phase 2 documents, http://www.cabinetoffice.gov.uk/resource-library/g-cloud-programme-phase-2; October 2009.
US Government CIO Council. Proposed security assessment & authorization for U.S. government cloud computing. Draft version 0.96; November 2, 2010.
U.S. Department of Health & Human Services. HIPAA privacy rule. summary website, hhs.gov/ocr/privacy/hipaa/understanding/summary/index.html; 2011a.
U.S. Department of Health & Human Services. HIPAA security rule. summary website, hhs.gov/ocr/privacy/hipaa/understanding/srsummary.html; 2011b.
Wikipedia. Security through obscurity, en.wikipedia.org/wiki/Security_through_obscurity; 2011.

Dr. Ulrich Lang is co-founder and CEO of ObjectSecurity®, "The Security Policy Automation Company™". ObjectSecurity's OpenPMF™ product makes application security manageable through automation. Ulrich is a renowned thought leader, author and speaker on model-driven security, security policy, cloud/SOA/middleware/application security, and has over 15 years of experience in information security. He received a PhD from the University of Cambridge Computer Laboratory (Security Group) on conceptual aspects of middleware security in 2003, after having completed a Master's Degree in Information Security with distinction from Royal Holloway College (University of London) in 1997.

Rudolf Schreiner is co-founder and CTO of ObjectSecurity®, "The Security Policy Automation Company™". ObjectSecurity's OpenPMF™ product makes application security manageable through automation. Rudolf received his Master's (Dipl-Phys.) in physics from University of Munich in 1993. After graduation he worked as an independent software engineer and consultant on various computer security projects in banking, manufacturing and telecommunications. He is a renowned thought leader in model-driven security. Rudolf is currently the lead software architect for the OpenPMF product.