A case study of separation of duty properties in the context of the Austrian "eLaw" process.

Andreas Schaad & Pascal Spadone
SAP Research
805 Av. Maurice Donat
06250 Mougins, France
{andreas.schaad,pascal.spadone}@sap.com

Helmut Weichsel
Federal Chancellery Vienna
Ballhausplatz 2
1014 Vienna, Austria
helmut.weichsel@bka.gv.at

ABSTRACT

Over the last few years rapid progress has been made in moving from conceptual studies, "whitepapers" and initiatives to the actual deployment of e-Government systems [13]. In this paper we present the case study of an existing e-Government system (eLaw) which already supports key legislative processes in the country of Austria[1]. The study has been performed in the context of the EU FP6 project "eJustice".

We present a detailed system and workflow representation referring to the example process of changing a federal law in Austria. Since such processes and their results, i.e. the laws of a country, have an enormous impact on society, they need to be secured against external and internal alteration, be it inadvertent or malicious. This is even more important in the electronic world.

Instead of discussing the obvious security requirements like virus protection or network-level access control, our focus is on an often neglected form of organisational security and control properties called separation of duties. We will analyse and discuss a set of these in terms of the described eLaw process.

Categories and Subject Descriptors

J.1 [Administrative data processing]

Keywords

Legislation, electronic documents, e-Government, workflow security, organisational control

1. INTRODUCTION

Whereas private corporations have been using information and communications technology (ICT) to improve the efficiency of their business for two decades, public sector agencies have only started to consider it rather recently. Nevertheless, governments are now aware that offering their services online will help them to reduce costs [23]. In that context, various services such as applying for a passport, registering as a voter or filing tax returns have been made available online in several countries [24], [13].

However, the judicial domain has so far not shown as much interest for ICT as other public administrations. One reason often heard when speaking to responsible staff is that the judiciary world seems to be afraid that computers will take away some of its independence. Still, the room left for increasing the efficiency of current judicial administrations is recognized by judicial professionals and citizens alike; hence it is not surprising that most governments agree on the fact that ICT solutions need to be adopted in their judicial administrations [15, 5].

One notable exception to this pattern of low usage of ICT in the judicial system is the Austrian eLaw system, which aims at replacing all paper-printed law in Austria. This system supports all legislative stages from drafting a document, its review and debate in the appropriate chambers, application of digital signatures, to final publication in an internet-accessible database. It allows each Austrian citizen to access the laws, regulations and other supporting legal information of their country. Here we present a case study of one particular process that uses this system: The process of changing a federal law concerning the Austrian Highway Code.

Since such processes and their results (the laws of a country), have an enormous impact on society, they need to be secured against external and internal alteration, be it inadvertent or malicious. This is even more important in the electronic world. In fact, strong evidence [11] suggests that internal alteration is far more critical than often perceived in the current age of Internet-connected systems. Since possible attacks can come from someone inside the organisation, who might even have the appropriate access rights, technical measures like network firewalls do not address this threat.

An often neglected, but very effective [19] form of enforcing organisational security and control properties is through separating duties, e.g. by assigning roles that are strongly separated and mutually exclusive to principals who work with critical resources.

SAC'05, March 13-17, 2005, Santa Fe, New Mexico, USA.

[1] This case study has been performed in joint collaboration between the BKA Austria and SAP Research in the context of the EU FP6 project "eJustice". The views expressed in this paper are not representative of SAP and its products or strategies.

This paper is organized as follows. Section 2 presents an analysis of the process of making changes to a federal law in Austria, including descriptions of the involved principals. In Section 3 we derive some necessary security requirements with reference to threat scenarios that could apply at different steps of the case study. We then present a set of separation properties in section 4 and discuss how these can address the previously elicited requirements. Section 5 provides a discussion of related and further work.

2. SCENARIO DESCRIPTION

This section describes a scenario as observed at the Austrian Federal Chancellery (BKA) in the context of the Austrian Legal Information System (RIS) (http://www.ris.bka.gv.at/). The overall aim of this system is to replace printed law texts by digitally signed electronic documents [12]. We first introduce the system as a whole, and then we detail the steps of the process we will study.

2.1 Background

The Federal Chancellery is one of 12 ministries in Austria. To fulfil their administrative duties these ministries use a variety of supporting IT systems.

One of the systems is called eLaw. This is an electronic legal records processing system which certain ministries make shared use of. This system can, for example, be used to facilitate and manage changes to existing laws. As such, eLaw may be classed as a records management system for public administration. The workflows implemented in eLaw are enforced through Fabasoft's eGov Suite 5.0.

Legislative information (e.g. gazettes, instruction edicts or tribunals) and the law that has been agreed upon by the involved political parties, is published in the Austrian Legal Information System RIS. The aim of this system is to replace printed law texts with digitally signed electronic documents, which are legally binding. The RIS currently provides services to more than 17,000 public administration officers over a nation-wide Intranet dedicated to the task. In addition, the general public may access the electronically published law via the Internet. RIS users access more than 6.5 million documents each month. The daily update rate of the RIS information repository can be up to several hundred documents changed on-line, with the system required to be constantly available: 24 hours a day, 7 days a week, all-year round.

The eLaw system is one of more than 30 public administration systems that feed data for publication into the RIS. Other such systems include the Supreme Court, the Administrative Court and the State governments.

2.2 Process Description: Updating a Law

A typical scenario detailing the use of eLaw and its interaction with the RIS is that of a change to existing law, e.g. a change to the law concerning the Austrian Highway Code. This process is illustrated by figure 1, and in the rest of this subsection each paragraph describes a step of the process (denoted by a rectangle in the diagram). Note that each step is based on a specific law which prescribes the exact legislative procedure, however, a more detailed analysis is not possible in this context.

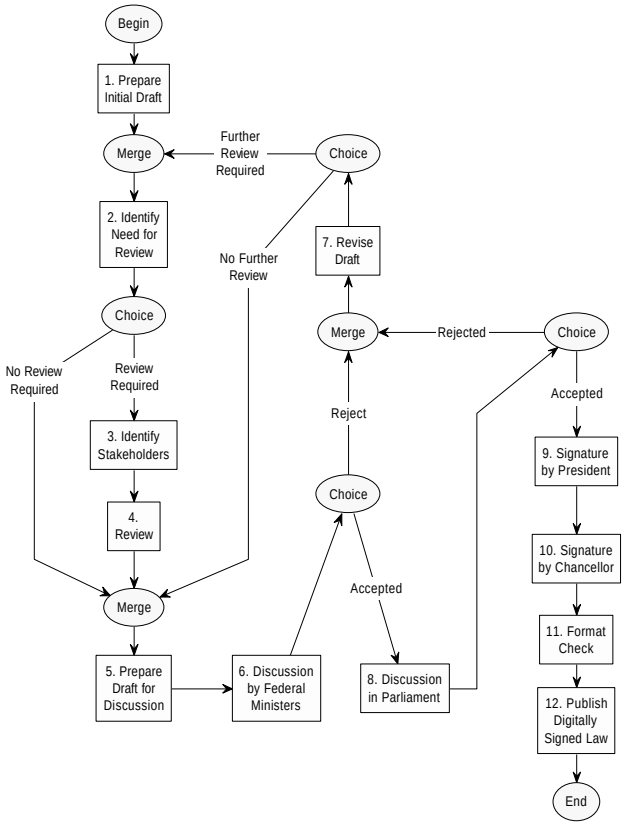

Figure 1: Workflow Representation

A law clerk working for the ministry of transport prepares a draft of the proposed change (step 1). This draft is initiated within the eLaw system (i.e. it is given an initial classification and some MS Word Documents are created). It is then decided whether the draft should be reviewed by external stakeholders (step 2). In our case of a change to the Highway Code, the draft would normally be sent to the Austrian Automobile Club for comments. The stakeholders are identified and invited either electronically (via email) or by post to review and comment on the draft bill (step 3). The draft is made available to them by physically sending them a copy of the draft or through the RIS (but not using an electronically signed format at this stage).

The stakeholders then review the draft (step 4), and send back their review either electronically or by hardcopy. A deadline may have been set for the review. Once the deadline for the stakeholders' review and comments has passed, the draft is prepared for further discussion (step 5) by a department responsible for coordinating the meetings of ministers, and the draft is eventually put on the agenda of the weekly meeting of the Austrian federal ministers.

If at the discussion (step 6) the draft is rejected, it has to be proposed again by the initiating ministry. In this case the draft is revised by a law clerk (step 7) and it is decided whether further review of the revised text is required. If the federal ministers agree to accept the draft without change, it becomes a government bill and is put into the RIS. At this stage there is no binding digital signature.

The bill is transferred to the systems of the parliament, which run independently of the eLaw and RIS environments. At this stage, an .rtf file containing the government bill is prepared and is placed on a dedicated server where it can be collected by the parliament's systems.

The government bill is now discussed by national council and then by the federal council (step 8). Both chambers may either agree or disagree, and possibly change the government bill. In case of an agreement, the text is passed back to the eLaw system and BKA to be published in the federal law gazette. In case of a rejection or possible veto, the draft is again revised by the originating ministry (step 7), the same way as if it was rejected by the federal ministers. The publication of the bill must obey very strict structural and layout requirements which are currently enforced through a set of MS word macros (over 70 different templates).

Prior to final publication, the president must sign and approve that the change to the law has been performed according to the constitution (step 9). Note that he approves that the legal process has been correctly followed; he does not approve the content of the bill. The president's approval must also be countersigned by the chancellor (step 10). These two steps currently require a paper-based signature.

A final check of the new or changed law is performed by the constitutional service (which is a department of the federal chancellery), who also give the document the appropriate label with respect to the federal law gazette (step 11). The changed law is then published in the RIS after it has been digitally signed to provide authenticity (step 12).

3. SECURITY REQUIREMENTS

Based on available data from private sector organizations [21, 10, 11], we believe that the threats to systems dealing with law texts originate from the inside of an administration rather than from the outside. Unlike a forged cheque which is only paid out once, a manipulated electronic law text may become legally binding to an entire country and may be applied in thousands of instances. Accordingly, after having described the legislative process, we now analyze some of its specific requirements in the area of security, and more specifically regarding access control. To illustrate our requirements we present possible but partially "hypothetical" threat scenarios from different stages in the process described above.

Requirement 1: One of the most often ignored requirements, despite being one of the most obvious, is that a legal clerk should not work in two incompatible offices. For example, a clerk working in the ministry of transport should not have access to information in the department responsible for releasing public tenders for highway maintenance.

In the specific context of the Austrian Chancellery, there is a strict policy that clerks working in different sections should not interact. This supports the validity of our presented requirement.

Requirement 2: The clerk initiating a legal draft should not be the principal who decides whether reviews by stakeholders are required.

Several possible threats arise if the initiating clerk is responsible for deciding about the need for review. For example, he may have a personal interest in the change he made, and will want to skip reviews, so that changes will go unnoticed.

Requirement 3 a: A clerk should not be allowed to modify a document and upload it onto the RIS at the same time.

The threat here is that a clerk responsible for drafting the document, and the review of the stakeholders' comments, could publish changes to the RIS immediately. In order to address this threat another clerk should be made responsible for first forwarding the changed document to the appropriate chambers and finally uploading it.

Requirement 3 b: A clerk should not be allowed to remove an already agreed upon document from the RIS without having been involved in its prior drafting.

Requirement 3 c: A clerk should only be allowed to remove an already agreed upon document from the RIS if he has not been involved in its prior drafting.

Requirements 3a – 3c represent possible alternatives for a requirement regarding addition and removal of documents to RIS, with 3b and 3c being almost the opposite of 3a. However, we observed that from time to time documents which have been agreed upon have to be removed from the RIS when small mistakes (e.g. a wrong date or spelling) were noticed.

Requirement 4: In case of a rejection (step 8) the draft has to be proposed again by the initiating ministry. However, the draft must not be revised by the same law clerk who initiated the draft.

In this case, it might be that the initiating clerk was too biased in his views and as such a fresh perspective is required.

Requirement 5: A clerk should not perform all the workflow steps involving a legal bill.

As a general security requirement, a clerk should not be permitted to work on a legal bill from drafting through to publication. At least one other clerk must be involved at a critical step (e.g. step 12 – final publication). This is often also referred to as a four-eyes principle or dual-control since two principals must agree on a change or supervise each other.

We immediately see from all of the above scenarios that it is essential to provide clerks with only the access rights they need to perform their tasks, limited to the times at which they need them, and only when such rights are compatible with actions they have previously performed. The workflow is the context-providing concept that is required to achieve these three properties and requirements like those above must be addressed with and within the workflow. That is to say that information used to make access decisions is provided by the workflow (e.g. who performed which step on which object), and that access control must be applied at selected steps in the workflow.

The next section argues that separation of duty principles can help to support the aforementioned requirements. It can be observed that information about the current and past steps of a principal acting in a workflow are essential for enforcement.

4. SEPARATION OF DUTIES

Although the origins of separation principles can be clearly identified in the development of organisational theory, e.g. [16, 14], and internal control and accountancy frameworks, e.g. [7, 4], we believe that separation properties are equally required and applicable in the context of the judicial domain.

Research in the areas of role-based access control, e.g. [9, 6, 17] and distributed systems management, e.g. [2, 8] has led to the definition of taxonomies and frameworks for separation of duties (SoD). In the context of role-based access control systems [18], separation of duties are enforced based on the notion of mutually exclusive roles. Such roles are pairs of organisational roles which are in some way incompatible; that is if a single user acted with both roles simultaneously, it would violate some organisational control or security principle. Mutually exclusive roles thus affect the assignment of access rights to principals.

We present a selection of static and dynamic separation properties, and show how they can be used to enforce the security requirements stated in the previous section. This selection is based on the initially suggested taxonomies by [22] which were later formally refined and validated using automated formal specification and analysis related approaches [19].

4.1.1 Static Separation of Duties

a) (Simple) Static Separation of Duties: *A principal may not be a member of any two exclusive roles.* Despite its simplicity, this category of SoD properties is sufficient to fulfil requirement 1 in section 3: If the roles are defined in the system and users are assigned to roles according to real world requirements, it is possible to ensure that a principal will be a member of at most one of the two incompatible roles.

4.1.2 Dynamic Separation of Duties

a) (Simple) Dynamic Separation of Duties: *A principal may be a member of any two exclusive roles but must not activate them both at the same time.* This category can be used to meet requirement 3a in section 3: Whereas clerks can sometimes modify a text and sometimes send it to the RIS, they can be prevented from doing both at the same time. Depending on how fine-grained the role-structure is and which rights have been defined (i.e. if the right to remove a document is defined), scenarios 3b and 3c can also be addressed using this simple dynamic property.

b) Object-based Separation of Duties: *A principal may be a member of any two exclusive roles and may also activate them at the same time, but he must not act upon the same object through both.* This property could help to meet requirement 1 in section 3. In fact, it clearly illustrates the effect of trying to control access without workflow information, and the work-around based on operating system or application-level sessions and role-activation [17] is the result.

c) Operational Separation of Duties: *A principal may be a member of some exclusive roles as long as the set of authorizations acquired over these roles does not permit them to execute every step of a workflow.* This covers requirement 2 above. A clerk can be prevented from performing the two steps involving submission of the draft document, if these are split out into a separate sub-workflow within the main process.

d) History-based Separation of Duties: *A principal may be a member of some exclusive roles and the complete set of authorizations acquired over these roles may cover an entire workflow, but a principal must not be able to perform all the workflow steps involving the same object(s).* This finer-grained category of SoD is the only one that fully caters for the requirement of scenario 5 whilst allowing maximum flexibility. A clerk can be prevented from acting at every step involving a particular RIS object, even if he is allowed to act at every step involving some other object.

On the conceptual layer these properties are based on mutually exclusive roles. Such roles can be defined at the operating system / network level (e.g. Windows XP), database level (e.g. an Oracle8 database [18]), or application level (e.g. an SAP Human Resources module). However, as far as we are aware of, at none of these different technical levels are workflows taken into account. This means that the properties of 2c and 2d can not realistically be enforced.

5. Discussion

The workflow is the context-providing concept that is required to achieve the requirements presented in section 3. Expressing separation of duty properties must be addressed "with" and "within" the workflow. Surprisingly, only little work, e.g. [1, 3, 26] has been done so far on investigating the role of workflows for access control decisions.

In most workflow applications, access control mechanisms will already be present at the database level and at the application level, either for historical reasons (a workflow is added on top of existing services in order to link them together) or because different security requirements have been identified for each of those levels. [25] addresses this issue in the case of two layers (database and application) of access control. It is necessary to ensure that SoD properties enforced at the workflow-level are consistent with, and do not conflict with the lower-level access control properties.

This raises the question of defining how workflow-level SoD properties will be specified by workflow application designers. So far we envisage two possible approaches: Either to embed the security properties inside the workflow behaviour definition (using a security-aware workflow definition language which does to our knowledge not yet exist), or to specify them outside of the workflow definition (using a policy specification language such as Ponder [8]). The main difficulty comes from the difference that must be made between identifying which users should perform a workflow task and identifying which users are allowed to perform a task. Ponder provides obligation and authorization policies to answer these questions.

Recently, new concepts have also been proposed to delegate and revoke tasks (obligations), in order to change workflows at run-time. These tasks need to be controlled by review and supervision concepts, and we believe that such delegation and revocation activities have a direct impact on the provisioning of access rights and enforcement of separation of duty properties [19, 20].

6. SUMMARY AND CONCLUSION

In the context of this paper we have presented a detailed system and workflow representation referring to the example process of changing a federal law in Austria and its authenticated publication. Since such processes and their results, i.e. the laws of a country, have an enormous impact on society, they need to be secured against external and internal alteration, be it inadvertent or malicious. This is even more important in the electronic world.

Instead of discussing the obvious security requirements like virus protection or network-level access control, our focus was on an often neglected form of organisational security and control properties called separation of duties. We analysed and discussed a set of these in terms of the described "eLaw" process and derived security requirements that refer to different stages of the process. We believe that these properties can be very effective to prevent internal attacks as they closely reflect organisational structure. However, to really exploit their dynamic potential, the context provided by a workflow needs to be taken into consideration.

7. REFERENCES

[1] Atluri, V. and Huang, W. *An Authorization Model for Workflows*. Lecture Notes in Computer Science 1146, 1996.

[2] Belokosztolszki, A. and Moody, K. Meta-Policies for Distributed Role-Based Access Control Systems. *In 3rd IEEE Workshop on Policies for Distributed Systems and Networks*, 2002.

[3] Bertino, E., Ferrari, E. et al. *The specification and enforcement of authorization constraints in workflow management systems.* Transactions on Informations Systems Security 2(1): 65-104, 1999

[4] BIS. *Framework for Internal Control Systems in Banking Organizations*. Technical Report No.40, Bank for International Settlement, Basel Committee on Banking Supervision, 1998

[5] German Federal Administation Office: BundOnline website, http://www.bund.de/ , 2001.

[6] Chen, F. and Sandhu, R. Constraints for RBAC. *In 1st ACM workshop on Role-Based Access Control*, pages 39-46, 1995.

[7] COSO. *Internal Control – Integrated Framework*. Technical report, Committee of the Sponsoring Organisations (COSO) of the Treadway Commission, 2002.

[8] Damianou, N. *A Policy Framework for Management of Distributed Systems*. PhD thesis, Imperial College, UK,2002.

[9] D. Ferraiolo and R. Kuhn. Role-Based Access Control. *In 15th MNCSC National Computer Security Conference*, 1992, pages 554-563

[10] Hulme, G. *The Threat from Inside.* Information Week, April 2003

[11] KPMG, *Fraud Survey Reports* 1996-2002, KPMG International Canada, 2002.

[12] Republik Oesterreich BGBl I Nr. 100/2003.

[13] Lenk, K., Traunmüller, R. (Eds.): Electronic Government, First International Conference, EGOV 2002, Aix-en-Provence, France, Lecture Notes in Computer Science 2456, 2002.

[14] L. Mullins. *Management and Organizational Behavior*. Prentice Hall, London, 5th edition.

[15] Prinz, W., Kolvenbach, S. *Support for Workflows in a Ministerial Environment*. In proceedings of the ACM Conference on CSCW, November 1996.

[16] Pugh, D. *Organization Theory: Selected Readings*. Penguin Business. Beguin Books, 3rd edition, 1990.

[17] Sandhu, R., Coyne, E. et al. *Role-based access control models*. IEEE Computer 29(2): 38-47. 1996.

[18] Sandhu, R., Bhamidipadi, V. *An Oracle Implementation of the PRA97 Model for Permission-Role Assignment. Third ACM Workshop on Role-based Access Control*, 1998.

[19] Schaad, A. *A Framework for Organisational Control Principles*, PhD Thesis. Department of Computer Science, University of York, 2003.

[20] Schaad, A. and Moffett, J. *Separation, Review and Supervision Controls in the Context of a Credit Application Process – A Case Study of Organisational Control Principles.* ACM Symposium of Applied Computing, Cyprus, 2004

[21] Shein, E. *CEO Warns Threats are Coming from the Inside.* eSecurityPlanet.com , June 2004.

[22] Simon, R., Zurko, M. E. Separation of Duty in Role-Based Environments. *IEEE Computer Security Foundations Workshop*, 1997

[23] Prime Minister and Minister for the Cabinet Office of the UK. *Modernising Governmen*t, presented to Parliament, March 1999.

[24] Cabinet Office of the UK: *Directgov webpage*, http://direct.gov.uk, 2002.

[25] Wimmer, M., Eberhardt, D., Ehrnlechner, P. and Kemper, A. Reliable and Adaptable Security Engineering for Database-Web Services*. In 4th International Conference on Web Engineering*. July 2004, Munich, Germany.

[26] Domingos, D., Rito-Silva, A. and Veiga, V. *Authorization and Access Control in Adaptive Workflows.* Proceedings of the 8th European Symposium on Research in Computer Security (ESORICS 2003), Springer-Verlag, LNCS, 2003.