Security Risk Management using Internal Controls

Simon Foley,
Department of Computer Science
University College Cork, Ireland
s.foley@cs.ucc.ie

ABSTRACT

Rather than treating security as an independent technical concern, it should be considered as just another risk that needs to be managed alongside all other business risks. An Internal Controls approach to security risk management is proposed whereby automated catalogues are built in order to provide information about security controls used to mitigate risk in business processes. Real-time evaluation and measurement of control efficacy in this model become essential to the management of risk using these catalogues and a risk-profile based approach to measuring security risk is described.

Categories and Subject Descriptors

K.6.5 [**Management of Computing and Information Systems**]: Security and Protection

General Terms

Management, Security

Keywords

Enterprise Risk Management, Governance, Internal Controls, Configuration Management

1. INTRODUCTION

The business activities of a modern enterprise—no longer confined within a closed-system—are distributed across open-systems spanning commercial, geographic and political boundaries. As an ongoing process, security governance prioritizes and manages risks to security across that enterprise. Security governance extends the technical infrastructure-centric view of risk management; it considers security risk in the context of the overall business.

In this paper we consider how Enterprise Risk Management (ERM) might be used to manage security risk and thereby support security governance. Section 2 provides an introduction to ERM frameworks such as COSO [2] that are used to document the relationships between business processes, their risks and the controls that are in place to mitigate those risks. This section also considers how the Clark Wilson model [5]—which can be considered as an early example of using Internal Controls for security—compares with a modern ERM framework. Section 3 describes how security risks can be managed within an ERM framework. The ability to measure risk is central to an effective risk management process and Section 4 explores how the risk-model described in [9] can be used to support metrics related to security-risk.

WISG'09, November 13, 2009, Chicago, Illinois, USA.

2. RISK MANAGEMENT AND INTERNAL CONTROLS

Organizations use Enterprise Risk Management in order to manage the risks that are related to their business objectives [14]. COSO-ERM [2] provides a systematic ERM framework for documenting the relationships between business processes, their risks and the controls that are in place to mitigate those risks. When properly maintained, an ERM control catalog provides data for business governance and also provides important documentary evidence for audit and compliance activities. This activity may be required by law. For example, in order to achieve compliance with the Sarbanes Oxley Act 2002, management must implement an effective Internal Controls system in the enterprise. Cobit is a further example of risk management framework that is centered around a collection of best-practices for managing IT systems.

Internal Controls is a process designed to provide reasonable assurance regarding the achievement of objectives in effectiveness and efficiency of operations, reliability of financial reporting and compliance with applicable laws and regulations. COSO-ERM is commonly used by auditors for realizing Internal Controls.

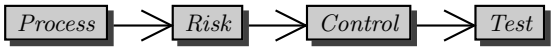
An Enterprise Risk Management framework can be characterized in terms of the following elements.

- Identify all relevant business *Processes* with the enterprise. Each process is composed of a number of *Sub-Processes* that have *Objectives* regarding the desired status for that subprocess.
- Assess the *Risks* that stand in the way of achieving each objective. Risk is the uncertainty in a process that could have adverse impacts on the business policy.
- A *Control* is a framework for management of an ac-

tivity or set of activities meant to prevent a business process risk from occurring.

- A *Test* procedure is a set of test activities executed at a pre-determined frequency to ensure that a control is effectively working as designed.

For simplicity of presentation in this paper we represent an ERM framework in terms of the relationships between these elements, whereby a process has a number of associated risks, which are in-turn mitigated by specified controls with associated test procedures. These relationships are depicted by the UML class style diagram:



Example 2 Consider a purchasing business process (and instance of *Process*) whereby purchase orders are authorized, suppliers selected and orders placed. Internal controls are required in order to address the risks due to fraud and other threats to the process.

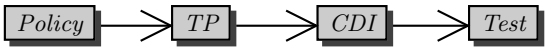
- *Risk:* Unauthorized creation of POs and payments to non-existent suppliers
  - *Control*: POs higher than $5,000 must be double approved.
    - *Test*: inspect a random selection of POs.
  - *Control*: only authorized users may access the payment system.
    - *Test*: inspect the application audit-logs.
    - *Test*: user login spot checks.
- *Risk:* poor demand planning in production may result in inadequate supply of materials.
  - *Control*: no PO higher than $5,000 will be approved at once.
    - *Test*: inspect the application audit-logs.
  - *Control*: staff receive production management training.
    - *Test*: inspect the training records.

A conventional controls catalogue can document a very large number of controls used to mitigate a range of risks, including operational, hazards, strategic and financial. The above example is an illustration of operational controls. △

A well known way to interfere with the effective operation of an enterprise is to work to rule [12, 19]. Sometimes relaxing the rules and accepting a risk is desirable; the overarching goal is to be able to identify, evaluate the seriousness of, and manage the risk. For example, perhaps it turns out to be acceptable for a manager to approve a $10,000 purchase order without a second approver in order to expedite a once-off production issue. Regular execution of the procedure tests and real-time reporting of controls failures becomes critical to the management of risk. Metrics in these frameworks can consider risk in terms of the efficacy of the controls. For example, the processes with risks whose controls have the highest number of audited failures.

Clark and Wilson [5] propose a model of data integrity based on the principles of good accounting. We suggest that the model can be interpreted as addressing one risk, that is, the risk of failure in *external consistency*. Intuitively, data has external consistency when it accurately reflects some real-world item. The Clark Wilson model recommends the use of well-formed transactions (Transform Procedures *TP*s) to ensure that Constrained Data Items (*CDI*s) are changed only in well-defined ways that preserve their integrity (external consistency). Controls include a security mechanism providing access-triple policy enforcement and separation of duty (*Policy*), along with any other controls that might be implemented as part of a *TP*.

Recognizing that there remains a potential for integrity violation, Clark and Wilson [5] also recommend the use of Internal Verification Procedures (*IVP*s) which carry out regular *tests* on the integrity of *CDI*s. These can be interpreted as providing tests on the effectiveness of the controls at ensuring external consistency. Unlike risk management, the Clark Wilson model does not particularly regard integrity security as a process; while *IVP*s are identified, monitoring failed controls and their potential role in managing acceptable (integrity) risk is not considered. For the purpose of comparison, one can approximate how the dependencies in Clark Wilson relate to ERM using the following UML class style diagram, whereby *Policy* (and *TP* to an extent) provide controls and *TP*s correspond to the business processes:



Attack trees [15] and related techniques such as fault trees [17] are also used to help identify, elicit and analyze attacks in an enterprise. Attacks are identified and refined in a top-down manner, along with their associated countermeasures. A variety of metrics [3,15,18] have been defined across attack trees that are used to analyze and/or compare the effectiveness of countermeasures at addressing attacks. Attack trees are typically used to assist attack elicitation and countermeasure selection. While they may be used to provide ongoing recommendation about the best current countermeasures [8], they do not explicitly consider the ongoing-process of testing countermeasure effectiveness. This difference to ERM can be illustrated by the following UML class style diagram, whereby an attack can be regarded as a knock-on consequence of some risk and the countermeasures are the controls that used to mitigate that risk:

We suggest that a logical extension to Attack Trees is the inclusion of countermeasure tests.

3. SECURITY CONTROLS

Operational security controls are an important part of any ERM controls catalog. However, the risks and controls tend to be relatively high-level and advisory rather than providing prescriptive information about detailed configuration. For example, PCI-DSS [6] advises the use of a firewall, but provides limited prescription on how it should be configured, other than according to best practice which may hide many of underlying risks. We take the position that Enterprise

Risk Management can be used to manage *known risks* related to security.

Example 3 The objectives of a business process can be generally at risk due to system compromise. A business-risk *Compromised systems leads to revenue loss* is identified and mitigated, in part, by the following controls.

- *Control:* firewall helps protect system from external attack.
  - *Test:* firewall configuration matches best practice.
- *Control:* ensure software patches are up to date.
  - *Test:* software version matches latest release.
- *Control:* antivirus software helps defend against known attacks.
  - *Test:* antivirus database is up to date
- *Control:* Access Control Lists (ACLs) help prevent unauthorized access
  - *Test:* ACL is consistent with corporate policy.

Suppose that the business processes *customer-support* and *WWW-sales* both have this risk and that copies of the controls (and procedures) are deployed on every server and workstation involved in these processes. We assume that an automated ERM framework such as [13] could be used to track and coordinate the control testing on individual systems. If a number of users involved in customer support neglected to patch their workstation software, then this would be reported as a (top-failing) control in mitigating the risk *Compromised systems leads to revenue loss* with respect to the *customer-support* process. △

Example 4 Specific technical security risks can also be identified, along with controls that provide specific configuration information. For example, the risk *SYN-Flooding results in unavailable system* can be partially mitigated by the following controls.

- *Control:* firewall threshold rule limits packet throughput
  - *Test:* firewall rules include a threshold rule.
  - *Test:* for packet flooding using intrusion detection system.
- *Control:* running `syncache` on server network stack limits flooding.
  - *Test:* system for `syncache` configuration.
  - *Test:* for packet flooding using intrusion detection system.

Again, the effectiveness of these controls can be monitored and provide information to the risk management process. For example, failure of these controls means that the risk *SYN-Flooding results in unavailable system* is not mitigated, which may require immediate treatment if it is related to the *WWW-sales* business process. △

The above examples are relatively simple, but we argue that the approach can be generalized to manage all of the different types of known security risks, etc., across the enterprise. An enterprise will have a very large number of technical security risks and we suggest that attack-tree based methodologies [8,15,17] can help in the elicitation and management of this complexity. Note that we propose the use of ERM to track known risks and do not consider how "unknown unknowns" might be discovered.

4. SECURITY RISK PROFILES

The ability to measure effectively the risks across the enterprise becomes central to an effective risk management process. Conventional ERM measures, such as top-failing-controls and control-failure averages, tend to be primitive and coarse-grained and do not provide much insight into the origin of the risk and/or how it might be mitigated. In [9] we describe an ERM model that supports user-programmable risk metrics. In this paper we use this framework to code examples of security-risk metrics that are closely integrated with the ERM framework.

A key component of the model [9] is a *risk profile*. A risk profile is a container that is used for risk calculations related to risk elements. For the purposes of this paper, one or more (security) risk profiles are associated with every instance of a security control. For example, every workstation related to the *WWW-sales* process has a unique security risk profile. A (security) risk profile defines a collection of *risk-attributes* that identify risk-relevant characteristics of interest. The attribute is intended to reflect a measure of something that is known about a security control. Attributes can be constant, discovered from the system or defined in terms of values of other attributes in a profile.

Example 5 Consider the control *Ensure software patches are up to date* used in the mitigation of the risk *Compromised systems leads to revenue loss* (Example 2). Let risk attribute valu define the value of a server protected by an instance of this security control and attribute patch defines the likelihood of server compromise as a consequence of running software with out of date patches. The risk of system compromise as a consequence of this control might be defined as

$$\mathsf{simpleRisk} = \mathsf{valu} \times \mathsf{patch}$$

This corresponds to the conventional definition of risk as loss multiplied by probability of failure [1]. If valu defines the potential monetary loss due to server compromise then risk corresponds to Annual Loss Expectancy. △

Attribute valu is an example of a (relatively) constant risk attribute. Attribute patch is a probability variable whose value may change based on the history of outcomes of the control test *current software version matches latest release*. In this case, one would expect that recent control test failures would indicate a higher probability of compromise than control test successes. Control test outcome history can be used to set the current value of a risk attribute and [9] describes a strategy based on a logistic regression analysis of subjective knowledge of a domain expert. In the case, for example, of attribute patch, a security expert asserts that, in the absence of any other information, there is small chance that a system, updated within seven days of a new patch

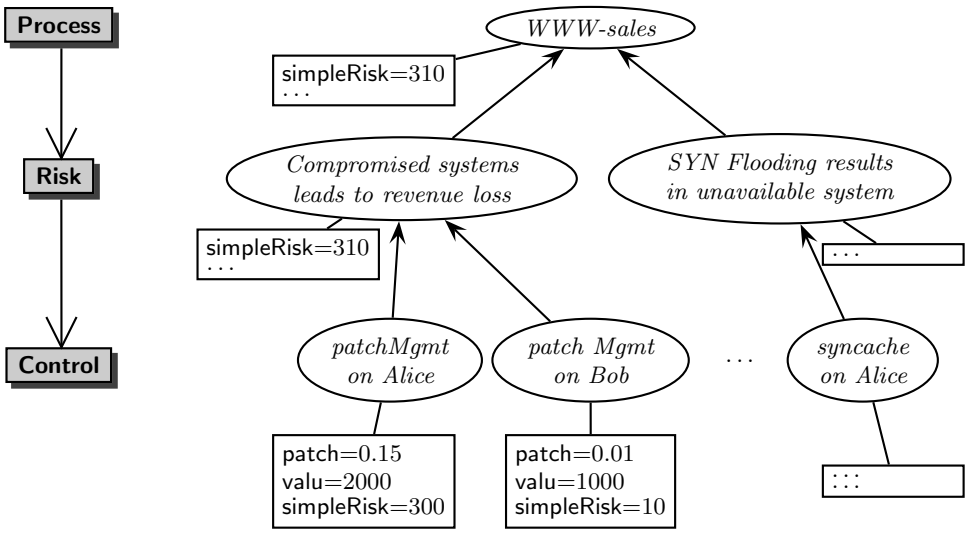


Figure 1: Individual process, risk, controls and their profiles

release, will be compromised, while there is a high probability of compromise if the system has not been updated after thirty days.

4.1 Risk Attribute Aggregation

All of the profiles related to a control can be aggregated, resulting in a risk-profile containing an overview of the risk associated with the control(s). Aggregation is done on an attribute basis, for example, the sum (aggregate) of the simpleRisk attribute values across all profiles related to servers in the *WWW-sales* process give an overall risk indicator in the context of the *WWW-sales* process. Risk profiles can be aggregated ("rolled-up") following the dependency relation $\rightarrow$ in *Process* $\rightarrow$ *Risk* $\rightarrow$ *Control* $\rightarrow$ *Test* providing contextual measures at the levels of controls, risks and processes.

Example 6 Figure 1 depicts a deployment of controls from Examples 2 and 3 for the process *WWW-sales*. A patch management control been installed on servers Alice and Bob, in order to mitigate the risk risk of compromise in the *WWW-sales* process. Individual risk profiles provide measures on each instance of the control's effectiveness and sample values are provided for attributes patch, valu and simpleRisk. These profiles are rolled-up (aggregating attribute simpleRisk using simple addition), providing measurements for the risk *Compromised systems lead to revenue loss* and the process *WWW-sales*. △

4.2 Risk Metrics

The risk-profile model [9] supports risk attributes defined across a metric-space whereby attributes may be aggregated using any triangular norm [7, 16]. Triangular norms are operations that generalize the fuzzy logic operators and [9] describes the use of the probabilistic sum $\oplus$ (or) and probabilistic product $\otimes$ (and) as risk attribute aggregation operators.

Example 7 Continuing Example 2, we identify the following security risk attributes in the risk profile associated with the controls deployed on a server.

- antivirus: probability of compromise due to an out of date virus database.
- firewall: probability of compromise due to misconfigured firewall.
- internalVuln: probability of compromise due to misconfigured ACL.

each of these attributes are updated based on the outcome of their associated procedure tests.

Each server has a risk profile and in this example the risk calculation for each profile is defined as

$$\begin{aligned} \mathsf{externalVuln} &= (\mathsf{antivirus} \otimes \mathsf{patch}) \\ &\quad \oplus (\mathsf{antivirus} \otimes \mathsf{firewall}) \\ &\quad \oplus (\mathsf{patch} \otimes \mathsf{firewall}) \\ \mathsf{risk} &= \mathsf{valu} \otimes (\mathsf{externalVuln} \oplus \mathsf{internalVuln}) \end{aligned}$$

The probability of compromise due to external attack is based on failure of firewall, patch and antivirus controls. The calculation of attribute externalVuln reflects an assumption that two or more control failures are required before an external vulnerability is considered to have occurred.

Overall risk for a profile is defined as probability of (internal or external) compromise times significance. This gives risk values in the range [0..1], where values near 0 are considered to indicate low risk and values near 1.0 are considered to indicate high risk. Probabilistic sum $\oplus$ can be defined as aggregation operator for attribute risk. On roll-up across control profiles, it provides a useful indicator for the risk *Compromised systems leads to revenue loss* in the context of the *WWW-sales* and customer-support processes. △

A key to effective security risk management is the construction of risk profiles that can provide meaningful indicators for decision making during the risk management process. One concern over using conventional arithmetic and/or probabilistic sum is that when making decisions humans do not necessarily aggregate in a linear manner [20], that is, there may be potential for non-linearity in the way that they

perceive combinations. Our risk model supports the compensation aggregation operator $\_ \oplus_n \_$, for neutral element $n : [0..1]$ based on $[4, 11]$. Let $x \oplus_n y$ be the compensating aggregation of values $x, y : [0..1]$ given neutral element n. It is defined as the additive operator $x \oplus_n y = \mathcal{G}^{-1}(\mathcal{G}(x)+\mathcal{G}(y))$ where $\mathcal{G}^{-1}()$ is the inverse of function $\mathcal{G}()$ and

$$\mathcal{G}_n(x) = \left\{ \begin{array}{ll} \ln(\frac{x}{n}) & [x \leq n] \\ \ln(\frac{(1-n)}{(1-x)}) & [\text{otherwise}] \end{array} \right|$$

and

$$\mathcal{G}_n^{-1}(x) = \left\{ \begin{array}{ll} e^y \times n & [e^y \times n \leq n] \\ 1 - \frac{(1-n)}{e^y} & [\text{otherwise}] \end{array} \right|$$

for $0 < n < 1$. Intuitively, this uni-norm operator may be thought of as a combination of probabilistic product when operand severity values are less than n, and probabilistic sum when operand severity values are greater than n. Using this operator, for example with $n = 0.2$, to aggregate risk causes the roll-up to be less sensitive to aggregation of low individual risk values.

5. CONCLUSION

This paper considered how Enterprise Risk Management (ERM) might be used to manage security risk, thereby supporting security governance. Rather than treating security as an independent technical concern, we argue that it should be considered as just another risk that needs to be managed alongside the other business risks.

A COSO-ERM style framework is used to model the relationships between business processes, their security risks and the controls that are in place to mitigate those risks. Real-time evaluation and measurement of control efficacy becomes critical to the management of risk in this framework and a risk-profile based approach to measuring the security risk, based on the model [9], is described. A feature of this model is that metrics are provided in context, with measurements related to individual controls, risks and processes readily available. While we have encoded some traditional risk calculations, encoding security metrics such as those in [10] is a topic for future research

A simple language has been developed that is used to construct arbitrary risk profiles. A compiler translates profile specifications, along with information related to procedure test behavior, into a relational database-based implementation model for integration with an ERM system. While the paper provides straightforward examples, the results are preliminary and further research is needed to determine how the approach might be used in practice.

Acknowledgements.

This research started when the author was a member of Corporate Security Strategy at IBM, on leave of absence from University College Cork. The research is currently supported by Science Foundation Ireland grant 08/SRC/11403

6. REFERENCES

[1] *Special Publication Risk management guide for information technology systems (800-30)*. Washington, DC: U.S. Government Printing Office, 2002.

[2] *Enterprise Risk Management-Integrated Framework, Committee of Sponsoring Organizations of the Treadway Commission (COSO)*. Jersey City, NJ, 2004.

[3] S. Bistarelli, F. Fioravanti, and P. Peretti. Defense trees for economic evaluation of security investments. *1st International Conference on Availability, Reliability and Security (ARES), Vienna*, April 2006.

[4] B. Buchanan and E. Shortliffe. *Ruled Based Expert Systems, The MYCIN Experiment of the Stanford Heuristic Programming Project*. Addison-Wesley, Reading, MA, 1984.

[5] D. D. Clark and D. R. Wilson. A comparison of commercial and military computer security models. In *Proceedings Symposium on Security and Privacy*, pages 184–194. ieee, April 1987.

[6] PCI Security Standards Council. Information supplement: Requirement 6.6 code reviews and application firewalls clarified. 2008.

[7] D. Dubois and H. Prade. A review of fuzzy sets aggregation connectives. *Information Sciences*, 36:85–121, 1985.

[8] S.N. Foley and W.M. Fitzgerald. An approach to security policy configuration using semantic threat graphs. In *Proceedings of 23rd Annual IFIP WG 11.3 Working Conference on Data and Applications Security*. Springer, LNCS 5645, 2009.

[9] S.N. Foley and H. Moss. A risk-metric framework for enterprise risk management. *IBM Journal of Research and Development, Special Issue on Managing Business Integrity through Integrated Risk, to appear.*

[10] A. Jaquith. *Security Metrics: Replacing Fear Uncertainty and Doubt*. Addison Wesley, 2007.

[11] E. P. Klement, R. Mesiar, and E. Pap. On the relationship of associative compensatory operators to triangular norms and conorms. *International Journal of Uncertainty, Fuzziness and Knowledge based Systems*, 4(2):129–144, 1996.

[12] A. Odlyzko. Economics, psychology, and sociology of security. In *Financial Cryptography: 7th International Conference*, 2003.

[13] J. Schablein P. Monson and C. Van Der Woude. *IBM Workplace for Business Controls and Reporting: Administration and Operations Best Practices*. IBM Redbooks, 2005.

[14] Root. *Beyond COSO: Internal Control to Enhance Corporate Governance*. Wiley, 1998.

[15] B. Schneier. Secrets and Lies Digital Security in Networked World. *Wiley Publishing*, 2004.

[16] B. Schweizer and A. Sklar. *Probabilistic metric spaces*. North Holland, New York, 1983.

[17] M. Stamatelatos et al. Fault Tree Handbook with Aerospace Applications. *NASA Office of Safety and Mission Assurance NASA Headquarters Washington,DC 20546, Version 1.1*, August 2002.

[18] L. Wang, T. Islam, T. Long, A. Singhal, and S. Jajodia. An attack graph-based probabilistic security metric. *Lecture Notes in Computer Science*, 5094:283–296, 2008.

[19] C.J. De Wolff, P.J.D. Drenth, and H. Thierry. *Handbook of Work and Organizational Psychology: Personnel psychology*. Psychology Press, 1998.

[20] H.-J. Zimmermann and P. Zysno. Latent connectives in human decision making. *Fuzzy Sets and Systems*, 4:37–51, 1980.