ERP Security and Segregation of Duties Audit:

A Framework for Building an Automated Solution

By David Hendrawirawan, Huseyin Tanriverdi, Carl Zetterlund, Hunaid Hakam, Hyun Ho Kim, Hyewon Paik, CPA, and Yeohoon Yoon

Many firms rely on enterprise resource planning (ERP) systems to implement business processes and integrate financial data across their value chains. This reliance increases the importance of ERP system security in protection of a firm's information assets.

In recent years, the audit of ERP security has gained importance and begun receiving an increasing percentage of firms' audit budgets. However, the audit of ERP security remains a complex, lengthy and costly task due to a confluence of factors.

ERP systems are inherently complex systems spanning many functional areas and processes along a firm's value chain. They are designed to provide flexible solutions to business problems. The sheer number of possibilities available for configuring an ERP system implies many potential security configurations. However, ERP systems pay little attention to potential conflicts and problems in those security configurations. Deployment and implementation of ERP systems also pay little attention to security implications, as the main purpose is to solve business problems within time and budget. In postimplementation stages, auditors have access to rudimentary ERP tools and capabilities for auditing security configurations. There are also shortages of staff members trained in ERP security.

This article focuses on segregation-of-duty (SOD) conflicts as a major class of potential ERP security problems. It proposes an approach for reducing the complexity, length and costs associated with ERP system audits aiming to identify and resolve SOD conflicts.

With increasing public and regulatory expectation for companies to implement and operate internal controls, ERP security and SOD have become popular subjects in the audit profession. ERP is an application system that integrates a company's business processes and financial data in one platform. While integration provides better quality and availability of financial information, it also increases the risk of fraud and misappropriations by users who have excessive authority (e.g., access to multiple duties that should have been segregated or access to critical transactions that should have been restricted). These users can execute multiple transactions that are in conflict from an SOD standpoint, allowing them to create a fictitious or fraudulent entry using one transaction and conceal the fraud using another transaction. Recently, the following trends have been observed in the world of information systems (IS) audit and control:

- A significant portion of the audit budget (internal and external) being devoted to review ERP security and SOD
- Increased training demands in the area of ERP security implementation and audit
- An increase in new product offerings aiming to implement and monitor SOD within an ERP system

Challenges in Auditing ERP Security

Unfortunately, the increased enthusiasm on this subject has been met with complex and costly challenges. Many companies and audit firms are not yet prepared to tackle the need for a rigorous ERP security audit. Many factors contribute to this reality, including the following:

- **The complexity of ERP systems leads to security vulnerabilities.** ERP systems must be able to process a wide array of business transactions and implement a complex security mechanism that provides granular-level access to users. For example, in SAP R/3, hundreds of authorization objects are used to allow access to various actions in the system. A small or medium-sized organization may have 100 transactions that are commonly used, and each transaction typically requires at least two authorization objects. If the company has 200 end users who fill a total of 20 different roles and responsibilities, there are approximately 800,000 (100*2*20*200) ways to configure security in the ERP—and this scenario excludes other complexity factors, such as multiple transactions sharing the same authorization objects, an authorization object having up to 10 fields that can be assigned to various values, and the possibility of using position-based security. The point of this illustration is that the inherent complexity of an ERP system increases the complexity of security configurations and leads to potential security vulnerabilities. Flaws, errors and SOD conflicts become more likely. Consider a scenario in which a security administrator has to grant read-only access to transaction X, which requires him/her to assign 10 authorization objects to the role. At a later point in time, management decides to grant write access to transaction Y, which implies assigning five more authorization objects. One of the objects is common to both transactions and determines the write capability. Although these two changes are seemingly independent, due to the shared authorization object granting write privileges, the unintended consequence is a potential SOD conflict. An ERP system does not automatically check for these kinds of security vulnerabilities. Unless the security administrator is well trained and employs rigorous positive and negative testing, he/she is likely to miss the unintended consequence of allowing write access to both transactions X and Y. As the number of potential

configurations and authorization objects increases, it becomes increasingly difficult and costly to analyze the security implications of ERP configurations, such as the unintentional creation of SOD conflicts.
- **There is a shortage of staff members trained in ERP security.** Most ERP training programs are designed for implementation efforts. They offer very little on ERP security and audit. Thus, there is a shortage of auditors who are trained in ERP security.
- **Implementors pay inadequate attention to ERP security during deployment.** Many companies do not pay adequate attention to security implications of ERP configurations during the deployment and implementation of ERP systems. Implementation teams are usually tasked with finishing the implementation projects on time and within budget. They do not pay adequate attention to security implications since it increases implementation time and budget. Due to limited emphasis on security implications, ERP security becomes too lax, making postimplementation problem identification and remediation very costly.
- **ERP tools for security audit are inadequate.** Most of the security tools available in ERP packages are not designed to facilitate efficient and effective audit of ERP security. The main emphasis of ERP tools is on security configuration and maintenance. Recently, there has been an increase in the number of third-party product offerings assisting with ERP security and SOD reviews. However, many users complain that those tools often generate false positives and create more work for auditors.
- **The customization of ERP systems to firms inhibits the development of standardized security solutions.** Every ERP implementation contains some level of customization specific to the firm undertaking the implementation. However, customization makes it difficult to develop a standard approach or methodology for conducting ERP security audits.

Conventional Approach

Most ERP security audits today are performed using a manual approach. There is little automation beyond the use of native tools that come standard with ERP packages. Most ERP native security reporting tools are designed with the purpose of assisting security administrators in validating the accuracy of security configurations. Although they are not meant for security audit *per se*, IT auditors with appropriate skills and knowledge can exploit these tools in extracting information about who has access to critical transactions. For each transaction to be tested, the IT auditor must set up different queries in the reporting tool and extract the output separately. The output is usually stored electronically and will later be processed using a computer-aided auditing tool such as ACL or a spreadsheet. Once the output is parsed and formatted to allow easier analysis, the auditor has a report that lists the users or roles that have the ability to perform a particular transaction, which will be validated by corroboration with management or company policies and procedures. This process needs to be repeated for every transaction included in the audit plan.

If the IT auditor is testing SOD, the list of users who have access to one transaction is compared to the list of users who have access to conflicting transaction(s). A user who appears in more than one list is considered a potential exception unless there is adequate mitigating control. This comparison needs to be repeated for every pair of conflicting transactions from an SOD perspective.

Unfortunately, the bottleneck of the manual approach is the limitation of the native security reporting tools found in most ERP products. These native tools are not designed to facilitate a large-scale audit effort, but rather to help security administrators perform occasional validation of the accuracy of security configuration. They allow reporting on only a single transaction per query, which may be adequate for a security administrator who works full time and handles each transaction request individually; however, it is not as practical for an IT auditor who is expected to perform the audit in a limited period of time and must test a large number of transactions. Although some IT auditors are able to utilize technology to perform this process more efficiently than others, as long as the process is based on the same philosophy of manual extraction followed by analysis, it continues to be an incredibly tedious and time-consuming task. The manual method is also prone to human errors.

Proposed Approach

This article proposes a new approach that can assist auditors in addressing the aforementioned challenges. The approach is based on the premise that IT auditors can use technology to partially automate ERP security audit processes. A conceptual model of an automated tool was created to perform ERP security audits, and the concept was validated by implementing and testing it in the context of SAP R/3. Steps in the approach are as follows:

1. **Understand the security concepts and mechanisms of the specific ERP system being audited.** To understand how to audit, it is necessary to understand how a particular user or role gains access to perform an action in the ERP system. In the context of SAP R/3, a user is assigned a specific set of authorization objects to gain access to perform an action or create a transaction. To audit those who have access to perform a particular transaction, it is necessary to research which authorization objects are required to access the transaction.
2. **Study the back-end components of the ERP system, such as tables and programs, that facilitate operation of the ERP security mechanism.** Since SAP R/3 stores information in a relational database, security configuration data can be found on a number of tables. In particular, it is important to identify tables storing information about users, roles, authorization objects and the tables that map relationships among these entities (e.g., which user has which role and which authorization objects are assigned to which roles).
3. **Design an *ad hoc* tool that emulates the back-end components of the ERP system and produces reports depicting the current security configuration.** As understanding of the tables that contain the security

configurations is gained, these tables can be extracted from the ERP system and placed into an external, *ad hoc* database. With an understanding of the ERP security mechanism, the mechanism can be reverse-engineered to create queries and programs in the *ad hoc* database and identify which users and/or roles have which authorization objects. Since the *ad hoc* tool is independent of the ERP system being audited, there is no dependence on the ERP native reporting tools, and the tool can be customized to overcome limitations of the ERP system's native tools.

4. **Identify security criteria that allow access to critical transactions or SOD conflicts.**[1] Although there is now an *ad hoc* security reporting tool, it will query only what it is told to query. The authorization objects required to perform a transaction (or in the case of SOD, a set of conflicting transactions) need to be determined, and the key to accomplishing this task is having the functional knowledge of the transactions that have an impact on financial reporting. For example, consider a test of the segregation between "maintain vendor" and "enter accounts payable voucher" transactions. To perform the "maintain vendor" transaction, the user must have authorizations A and B. To perform "enter accounts payable voucher," the user must have authorizations C and either D or E. Based on this knowledge, the following criteria are used to identify potential conflicts: search for users who have authorizations ABCD or ABCE.
5. **Create queries and reports that identify users who have potential SOD conflicts.** Using the criteria identified in the previous step, an extension is developed within the *ad hoc* tool to automate and simultaneously run queries required for identifying SOD conflicts. The extension is basically an additional table to map the transactions (or sets of transactions) to the sets of authorizations that would enable the user to perform them. The table makes it possible to automate the queries and "mass-produce" the security reports. Additionally, since the search criteria are not hard-coded, it is possible to customize and increase the extent of the testing without much human effort. This is a powerful capability; under the manual method, the IT auditor must run one query per transaction, and the queries cannot be performed simultaneously.

Benefits of the Proposed Approach

In essence, the proposed solution attempts to make the audit process more efficient by overcoming the limitations found with the ERP native reporting tools. It turns a repetitive and time-consuming part of the task into a split-second job. With this automation, it was possible to reduce a two-week manual querying process into a two-day automated querying process. The two days are spent extracting the back-end tables from SAP/R3, importing them into the *ad hoc* tool, entering criteria to be queried, running the queries and formatting the outputs. The auditor can use the remaining time saved by the automation to perform root cause analysis to yield more value-adding insights and recommendations, rather than merely identifying problems.

This approach creates a repeatable and consistent audit process. The test query criteria, which were saved in a table, can be reused in future audits. This allows the auditor to compare results between the current and previous audits and focus on significant differences between the two. In addition to saving time on the data extraction effort within one audit, the auditor can shorten the analysis time over multiple audits, thanks to the ability to repeat and reuse the knowledge base.

Besides the efficiency benefits, the proposed approach improves audit quality. As previously mentioned, the auditor can leverage the time savings gained from the automation to perform root cause analysis. Often a root cause analysis shows that the initial test scenario used by the auditor may not truly match the client's environment; this is often reflected by a high rate of "false positives" in test results. Accordingly, the auditor has to revise and perform the test scenario. If it were a manual approach, this would require much more time and effort. However, with the automated approach proposed in this article, it is a matter of changing the data in the query criteria table and rerunning the queries.

Risks of the Proposed Approach

The proposed approach entails a number of risks that must be addressed to maximize the potential benefits:

- To get accurate results, the test query criteria in the automated tool must be configured properly. To configure the queries properly, one must understand technical security concepts and the mechanics of the ERP product under consideration, as well as firm-specific business environment and functional customs. Failure to properly configure the tool can result in inappropriate or inefficient results. If the query criteria are incomplete or not sufficiently rigorous, the tool may not catch all of the SOD violations. On the other hand, if the query criteria are excessively rigorous, the tool may yield false-positives that take a lot of time as the auditor follows up and management tries to resolve issues that do not really exist.
- Management should be aware that the automated tool alone will not solve all of the SOD issues. Once the tool provides the SOD violations in the system, management must commit to making organizational changes and redesigning processes to resolve the violations. In considering whether to adopt the tool, management should be prepared to take action to resolve findings of the tool.
- It is erroneous to assume that an automated tool is a silver bullet to all SOD violations. Management and auditors can mistakenly become overly complacent because of this erroneous assumption. The automated tool should be used only as a means to an end. Management must be proactive in resolving the issues identified by the tool.
- Finally, management should refrain from using this tool to replace or significantly reduce auditors. While it is true that the tool can reduce the amount of time and effort spent on ERP security audits, it is not meant to replace auditors. Human involvement is still critical for planning, analysis and interpretation. The purpose of this tool is to enable auditors to save time on routine aspects of the audit and spend additional time on more critical tasks such as root cause analysis.

Conclusion

This article proposed a new approach for reducing the complexity, length and costs entailed in the audit of segregation of duty conflicts in ERP systems. This approach requires combining accounting and technical skills and knowledge. Leveraging technology to partially automate the audit task allows an auditor to save time on mechanical aspects of the audit and channel the saved time to value-adding tasks such as root cause analysis. As a result, the quality of audit findings and recommendations is improved. However, technology and automation are only part of the proposed solution. They must be used in conjunction with human personnel who understand security concepts and accounting principles. No amount of automation can replace human ability in analyzing the root causes of issues and interpreting query results from the automated tools. Management should refrain from treating an automated tool as a silver bullet to all SOD violations. Instead, management must fully commit to take action on the findings generated by the automated tool.

Endnote

[1] Since this requirement applies regardless of whether an IT auditor employs a manual or partially automated method, and since this project is primarily aimed at highlighting the benefits of automation, the authors relied heavily on practitioners' experience to obtain this knowledge.

Authors' Note:

This article was developed as part of a student project in the IT Audit and Security Course at the Red McCombs Business School (Texas, USA). The project was completed under the professional guidance of David Hendrawirawan and the academic supervision of Professor Hüseyin Tanriverdi. The project won the Best Student Project Award of the ISACA Austin (Texas, USA) Chapter in the first half of 2006.

David Hendrawirawan
is a manager at Deloitte & Touche LLP, in the audit and enterprise risk services division. He is involved in information system controls audit and consultancy, specializing in ERP security and application controls. His experience includes SAP, PeopleSoft, and Oracle Application Security and SOD audit.

Huseyin Tanriverdi
is an assistant professor at the University of Texas at Austin (USA). He teaches IT audit, security and business data communications courses, and researches risk/return implications of IT and business strategies.

Carl Zetterlund
has two years of experience as an IT administrator for a local firm and has completed an internship with Deloitte & Touche LLP in the Houston (Texas, USA) ERS practice.

Hunaid Hakam
worked for USAA in the summer of 2005 and recently completed an internship with Deloitte & Touche LLP in the Dallas (Texas, USA) ERS practice.

Hyun Ho Kim
is currently interning with the Walt Disney Company as a tax associate.

Hyewon Paik, CPA
has worked for Oracle in Korea as an ERP consultant with a focus on analyzing clients' business processes and requirements, and mapping all these factors into the Oracle financial applications.

Yeohoon Yoon
works for KPMG. He has two years of auditing experience with PricewaterhouseCoopers Korea.

Information Systems Control Journal is published by ISACA. Membership in the association, a voluntary organization serving IT governance professionals, entitles one to receive an annual subscription to the *Information Systems Control Journal*.

Opinions expressed in the *Information Systems Control Journal* represent the views of the authors and advertisers. They may differ from policies and official statements of ISACA and/or the IT Governance Institute® and their committees, and from opinions endorsed by authors' employers, or the editors of this *Journal. Information Systems Control Journal* does not attest to the originality of authors' content.

www.isaca.org