

April 10, 2007

(c. 26,000)

APPLICATIONS

Segregating Duties While Improving Developer Productivity

By Paul Farr

One of the main challenges that institutions face when complying with federal regulations is finding a solution that can restrict access to sensitive information. Demand for identity management has created a \$2.4 billion market for authentication, single sign-on, user directory, and access control solutions, according to a recent Yankee Group report. Internal controls such as identity management and segregation of duties are vital processes for meeting regulatory mandates. However, many organizations bypass their own processes when they need to solve application problems in production - creating the possibility of a serious security and compliance breach.

Segregation of Duties

Following the implementation of Sarbanes-Oxley section 404, regulators have put a strong emphasis on internal controls that prevent individuals from having the power to harm a company's critical systems, such as intellectual property or private customer records. "Segregation of duties" essentially ensures that roles are assigned to individuals in a manner so that no one individual can control a process from start to finish. By separating functional groups in IT, it becomes next to impossible, without collusion, for any one individual to damage the company's IT assets. This internal process of segregation of duties has become an integral

component of a company's security best practices.

To that end, financial regulators scrutinize IT policies to enforce such controls on the developers' access to the production environment. Allowing developers free reign in the production environment exposes the company to potential risks such as infringement of users' privacy or malicious or unintentional code changes resulting in application instability and performance issues.

Many organizations believe that deploying change management or access control systems to uphold segregation of duties is sufficient. While these technologies are effective for controlling access during planned events, such as system upgrades, they do not address the workflows involved when developers require access to the production environment to resolve unexpected problems. Often, if an application problem occurs in the production environment, if the support staff is not able to resolve a problem, the issue escalates to the developer, which can mean granting the developer full access to the production environment in order to solve the problem. Since most application problems cannot be reproduced in a lab environment, troubleshooting techniques require that the developer is provided access to the production environment to try and figure out the root cause of a problem. Typically, this may involve replicating the error, debugging

code, downloading patches, and conducting performance tuning.

Traditionally, there has been no practical way for developers to solve application problems without penetrating the production environment to troubleshoot the software problems. Without this access, developers were left to piece together what had occurred using only anecdotal user reports and disparate system logs. This would involve a lengthy, iterative process of gathering information from every component of the system, network configuration and then finally down to the code level. The typical diagnosis process would involve conducting one to several end-user

Productivity page 22

APPLICATIONS

Productivity

Continued from page 9

interviews, and possibly, making costly on-site trips to see firsthand what is happening. Thus, when a problem occurs in a mission-critical application, locking developers out of the production environment is simply not an acceptable answer.

A New Approach

There is a better alternative that provides developers with both the information they need to solve tough application errors and eliminates security and compliance risks. By implementing application problem resolution technology, organizations can capture a complete record - from user actions to code-level changes - involved in the uncovered software defect, allowing them to quickly get to the root of the problem. All of the relevant data that the developer will need to investigate the error is extracted and stored in a secure repository that contains secure controls around the access and privileges given to developers requesting to get the digital records for application problem resolution. Full access controls based on authorizations and authentication, filtering of the digital records and an audit trail documenting who got access to what are maintained. Thus, developers retain the critical knowledge to efficiently solve production issues, without

having to enter the production environment for troubleshooting. What's even more effective, application problem resolution technology vastly reduces the time and effort necessary to discover and resolve application problems.

Troubleshooting in Production

Resolving errors in production code can be particularly troublesome. Each user has his own configuration of hardware, software, operating system, peripheral devices, and so on, creating endless iterations of configuration differences. These variations make reproducing the production environment virtually impossible. Therefore, when software problems occur, IT teams typically go through a costly, iterative process to gather information and replicate the problem before beginning the root cause analysis and resolution phases.

With application problem resolution technology, there's no need to replicate the error. The developer simply views a system log to discover the source of the issue, and reviews the problem history, correlating user interactions with the underlying code to pinpoint what caused the error to occur. This is particularly useful for identifying the cause of intermittent problems. By eliminating the need to recreate the problem in the environment in which it occurred, root cause analysis can be accelerated by up

to 80 percent while ensuring the integrity of mission-critical applications.

Automated Environment of Control

To maintain the highest degree of security, businesses must automate controls over privileged access and segregation of duties workflows for individuals working with production applications.

This can be achieved using technology that securely captures all the problem data in the production environment; extracts the problem logs to a secure repository outside of the production environment; enforces access controls to the secure repository; provides controls that securely filter the problem logs provided to developers, and creates an audit trail of the access activities and data seen by developers.

Conclusion

Financial institutions need to implement processes to ensure regulatory compliance and protect sensitive data. By implementing application problem resolution technology along with a strict environment of controls, businesses can eliminate commonly exploited security risks while enhancing developer productivity.

Paul Farr is vice president, marketing, Transaction Management Business Unit, BMC Software.