eWEEK

REAL LABS. REAL PRODUCTS. REAL RESULTS. | ELECTRONICALLY REPRINTED FROM JANUARY 5, 2011

IT Security & Network Security News

Application Development Security Considerations for the Cloud

By: Brian Prince

Building extensions to SAAS apps brings its own set of security challenges to the software development table.

In their list of predictions for 2011, application security specialists at the Denim Group predicted software development teams will start to shift their focus to building extensions to software-as-a-service applications instead of writing custom software from the ground up.

In the company's crystal ball, business-to-business providers will lead the way in this, though extensions to consumer-oriented applications will increase as well. As could be predicted however, this kind of shift would bring with it its own set of challenges for developers looking to integrate their creation securely, experts told eWEEK.

"The overarching problems with securely integrating with SAAS [software-as-a-service] applications is that the systems involving these integrations have more complicated threat models than normal Web applications and the integration patterns between custom code and SAAS services are not as standardized or well-understood," said Dan Cornell, CTO for the Denim Group. "This creates a situation where developers do not necessarily understand how to build these interactions securely, and it also makes it challenging to provide standardized guidance to developers because, in the absence of specific platforms and desired features, this guidance is often 'it depends' or 'it's complicated.'"

The dependency on SAAS components they don't control poses a challenge for enterprises as well, Forrester Research analyst Mike Gualtieri told eWEEK.

"Enterprises become more vulnerable because of the dependency created by SAAS 'Franken-apps,'" he said. "If you find a vulnerability in a component and cannot fix it yourself, then you are now at the mercy of the SAAS provider

to fix it. What do you do in the meantime? Take the application down? Architects must threat model the entire application and provide contingency plans in case a vulnerability surfaces during runtime."

The approach to security requires a number of things, Cornell said. For one, developers need to validate data coming from the SAAS applications to prevent injection and other attacks from being propagated between portions of the system, he said. In addition, data sent to the SAAS provider needs to be properly encoded. Then there is the issue of authentication.

"Depending on the characteristics of the system, some sort of credentials for access to the SAAS provider must be used," Cornell said. "If these are provided by the user, then proper management might only require them to be encrypted while in transit. However, if all access to the SAAS provider is anonymized behind a single account, the credentials for that account must be stored in a secure manner while at rest and proper logging must be maintained in order to determine what users attempted what actions."

The prospect of an attacker getting past the authentication page by being a customer of the SAAS application means the entire application functionality is the attack surface, noted Chris Wysopal, CTO of Veracode.

"Many app developers downplay the insider threat and feel protected just because their Web app has authentication and only authorized employees can log in," he said. "With a multitenant SAAS application, this is not the case anymore as attackers can be other customers of the SAAS app provider. This increases the need to get application security right for cloud-based applications."

A further risk for SAAS is the fact that the administration of the systems housing the customer data is typically performed by the SAAS developer and the hosting provider, which makes building in data encryption at rest and in transit a must, Wysopal said. Good auditing and logging are also requirements, he added.

"So in summary, the threat space and the attack surface have increased greatly for SAAS apps," he said. "This makes application security features like encryption, auditing and two-factor authentication much more necessary. Vulnerabilities are much more exposed in a multitenant world, so secure coding, security testing and vetting of third-party components are requirements."