A Local Application Security Detection System Based on PaaS

Jie Shang\textsuperscript{a}, Xiuquan Qiao\textsuperscript{b*}, Yahui Lu\textsuperscript{c}, Xiaofeng Li\textsuperscript{c}

\textsuperscript{a} State Key laboratory of Networking and Switching Technology
\textsuperscript{b}Beijing University of Posts and Telecommunications Beijing, P.R. China
\textsuperscript{c}China mobile Beijing Branch Data Service Department

Abstract

PaaS is one of the hot topics in computer technology, in which users build their applications with the provider's on-demand tools and collaborative development environment and don't have to worry about hosting, updating, or maintaining applications. However, PaaS systems still have to provide high-quality development tools to succeed. A security checking system described in this paper integrates security checking into development curriculum of web applications based on PaaS, thus enabling developers to build and release products more quickly. Based on our evaluation results, we provide recommendations for future research.

© 2011 Published by Elsevier Ltd. Selection and/or peer-review under responsibility of Harbin University of Science and Technology

Keywords: PaaS; Security check; Application development;

1. Introduction

In October 2007, IBM introduced the White Paper on cloud computing gives the following definition: "Cloud computing\textsuperscript{[1]}" is used to describe a platform or a type of application, a cloud computing platform for dynamic supply-demand provision, configuration, reconfigure and deprovision and so on. PaaS is not only to provide some basic API as the traditional system software, but also provide more advanced service-oriented API, so that the upper application software developers can easily use the service interface to design specific applications. Current representatives of PaaS\textsuperscript{[2]} platform are Google App Engine, Salesforce.com, SINA App Engine and so on. Coming of Cloud computing era has brought the software development models and business models into a new era.
Current study is relatively rare for PaaS, current situation of building applications using PaaS platform is as follows:

- Different from traditional development model, development in cloud computing era stresses that applications can be developed by anyone, at anywhere. Cloud computing offers more storage and processing power.
- Security concerns are less during application development. It should be noted the entire software life cycle of cloud formation of a horizontal spiral, vertical incremental software development chain. This new software development model demands more efficient methods of detecting security of the application at local. However, following problems exist in current application development using PaaS[3]:
  - Existing security research of local development process is less. Requiring that all third-party developers (especially ordinary developer) to develop a very secure application is unrealistic. Therefore strict security review is necessary when accessing third-party applications. Repeated develop and deploy between developer and reviews brought complex work to both sides.
  - For developers, comprehensive security policies can help them take the initiative to self-detect applications to avoid security risks, which will effectively speed up the deploy process. For reviewers, providing security detecting tools and guiding developers to take targeted detection of applications will help preventing most of the potential security vulnerabilities [4].

Based on the above analysis, this paper presents ASCS(Application Security Check System over PaaS). ASCS is aimed to provide better application security check service, transfer the main tasks of security checking from platform side to developers. ASCS is Implemented in the Eclipse plug-in approach and based on OMP application running engine simulation platform.

2. A Local Application Security Detection System Based on PaaS

2.1. Application run-time engine simulation platform

OMP application run-time engine simulation platform is a cloud computing cluster B / S model of the network platform environment, which offers a variety of business logic execution environment. Run-time engine for the application provides the underlying business logic required to provide a variety of computing resources and data resources, and applications do not need to know resources details of which provide the underlying node.

System is mainly divided into AppMaster, NodeAgent and AppRouter module, the structure is shown in Figure 1.

![Fig. 1 OMP Application run-time engine system](image1)

![Fig. 2 ASCS system structure](image2)

2.2 ASCS (Application Security Check System over PaaS )
ASCS aims to provide a easy-to-operate, easy-to-use cloud application security detection systems for developers. ASCS provides users with a standard integrated development environment [6], and to support the expansion of the form of plug-in support for a unified user interface program editing, testing the entire process. System architecture is shown below[7]. The system composition and functions of each layer are described as follows.

- **ASTS Application development layer**: This layer provides users with a standard integrated development environment, unified graphical user interface support for program editing, compiling, debugging, and load the whole process.

- **ASTS Application Security Checking Layer**: This layer provides users with a standard integrated development environment, unified graphical user interface support for program editing, compiling, debugging, and load the whole process.

Static code security check: for developers to code in the editing process, and compile the code synchronization is complete safety tips, check the code for that part of the source code itself is based on the premise of no compilation errors, code for the vulnerability discovery and safety tips.

Vulnerability check directory: for developers in the development process, the application works on all directory scan, locate the required classification of non-Web project file entries, temporary files, backup files, encrypted files and hidden files directory to help developers find loopholes.

Common web security tips: detect the main web of the common security vulnerabilities such as SQL injection and XSS attacks[8].

Other security constraints of Cloud platform: the SDK filtering of specific key values including the cloud platform.

![Relationship between the local application security check modules](image)

The figure shows relations between the various sub-modules of the static code security check function. Each module are implemented based on spread point of eclipse plug-in approach. Green elements of each module represent the core of the interface module or extension points.

Security policy module includes security policy configuration and security policy implementation two parts. Security policy is defined and generated by security check of four types, and can be selected and canceled through option menu.

Marker module is used to generate the Marker used to identify different types of security flaws, compiled by the synchronization module creation, and ultimately reflected in the views module ViewPart, helping developers to locate security flaws.

View module, on one hand, provides interaction interface between the system and user: ViewPart; on the other hand provides adapters for all types of Marker interface.
Context menu module is responsible for fast response to file detections, as well as rapid positioning of various types of security flaws[9].

Vulnerability report generation / printing module conducts safety checking, generates real-time report, which is presented on report view in work bench of eclipse.

Resource unification location module is responsible of interaction between modules and the eclipse kernel. Synchronize compile module is core module of security check.

Syntax resolution tree of, which, according to security policy module policy configuration, check out security vulnerabilities in the generated code as IMarker, and try to achieve the adapter monitor control window, which is used to monitor IMarker generation, and in a timely manner to the display on interface ViewPart. The work flow is as follows:

![Synchronizing compile module workflow](image)

3. ASTS Application Build Environment

The following actual cloud application development process indicates the actual important operation steps ASTS participates.

- **Use the application wizard to create applications**: Figure A is a wizard ASCS provides for the application developers to create applications. Developers can create platform-independent remote cloud custom applications. This is the basis to develop, compile and deploy the applications.

- **Application configuration**: Figure b shows the basic configuration. At current, ASTS mainly support security testing for application developed by java language.

- **Security policy specification**: Developers can assign security policy through option menu provides by ASTS, and obtain a particular type of security test results. If security policy not specified, ASCS loads default security policy. As shown is Figure c.

- **Application security checking and result generation**: ASTS operates automatically when the save operation is performed by developers, after loading security policy configured, generates security reports and shows them in a specific view of workbench of eclipse. As shown in Figure d.

- **Application uploading**: Main forms of uploading applications cloud platform supports are: web upload, SVN and ftp.

- **Application security check at cloud side**: As security check in local simulation environment has been adopted, application can be successfully deployed after security check is performed at cloud side, effectively reducing repeated operations between local development environment and safety check at cloud platform side.
4. Summary and Future Research Prospects

This paper, starting from the limitations of traditional development mode and the new demands of the current cloud application development environment, makes some exploration on how to quickly deploy applications to the cloud platform, and achieves ASTS. Experiments show that ASCS has a high value in resolving the problems of a long development cycle raised by the status that application security cannot live up to the high standards of the cloud platform.

In the ideal cloud application development model, any detecting error will be immediately displayed on the IDE, and proposals will be given. Once the developer fixes a flaw in the code, following operations of building, testing, and uploading to the cloud can be done immediately in the IDE, which is the future direction of the cloud application localization.

Acknowledgements

This work was supported by the National Key Basic Research Program of China (973 Program) under Grant No. 2012CB315802, the National Natural Science Foundation of China under Grant No. 61171102, Beijing Nova Program under Grant No. 2008B50 and the New Generation Broadband Wireless Mobile Communication Network Key Projects for Science and Technology Development under Grant No. 2011ZX03002-002-01.

References