International Journal of Computer Science
Oddity...Probe...Reviste...
http://www.ijcsjournal.com Volume 2, Issue 1, No 1, 2014. ISSN: 2348-6600
Reference ID: IJCS-033 PAGE NO: 176-181

Execution Fault Localization in Large Dynamic Web Application Using Randomized Input Generation

C.P.Shabariram[#1], R.Balasubramaniam[#2]
[#1]*PG Scholar,* [#2]*Assistant Professor*
Kathir College of Engineering, Coimbatore, India
[1]shabariram91@gmail.com

***Abstract*— In today's online oriented environment, localization of dynamic faults has become a major role in web application development. This paper describes the novel framework named as Pin Down, an automated root cause analysis of dynamically generated web pages. Execution fault localization in dynamic web application is a prominent problem of finding root location that causes failure in which source code changes need to be made in order to fix predicted faults. Execution faults are web application error that causes incorrect functionality to the dynamic web application that also called as execution bugs. The dynamic execution nature of the web application separates the source cause of an execution bug by various fault localization techniques. To identify execution faults in dynamic web applications, some fault localization algorithms improved by using source mapping and extended domain for conditional and functional call statements, but it did not focus on server side vulnerability. This study presents a randomized input generation technique for dynamic web application to check whether the web page is vulnerable to SQL injection or not. An automated random input generation is constructed for all executable statements in source code to determine the execution failures, such as missing include file, incorrect SQL query and uncaught exception of the corresponding statement. In addition, the framework used to determine the HTML failures such as generated HTML page is not syntactically correct according to an HTML validator by checking appropriate tags with closing tags using parsing the DOM tree. Other than execution and HTML failure novel framework Pin Down used to locate execution failures caused by deprecated language constructs that are all produce obstructive exception and error message during execution. Final result shows that code coverage improved from 95% to 100% were the result indicates 100% code covered is a reliable indicator of the effectiveness of a test case.**

***Index Terms*— Pin Down, Fault Localization, Bug, Random input generation, dynamic web application, SQL Injection, Execution failure, HTML failure, HTML validator, DOM tree.**

I. INTRODUCTION

A false declaration of information and functional statements in a source code causes an unexpected failure to execution. It is a fundamental weakness of the design and implementation aspects. In source code debugging process, fault localization is a prominent method. Statistical fault localization can be divided into two mechanisms. The initial half is spot fault code by using various localization techniques. This part mainly identifies the program bugs. The second half is for debuggers to examine the identified source code to decide whether it contains bugs or not. In first part faulty code is prioritized based on its probability of bugs using statistical fault localization techniques. The source code having lower priority must be processed after the code has high priority. In the second part, verifies bug detection is perfect or not. So the programmer can easily classify the program into faulty and non-faulty. The amount of code need to be validated may increase the bug discovery process. Execution fault localization techniques reduce the human involvement needed for fault localization. These features lead to the suggestion and development of variety of techniques over recent years.

II. FAULT LOCALIZATION

In general, web application source code is written in a mixture of some programming languages, like HTML, Java Script, CSS and Flash for client side, PHP and embedded mysql in server side development. Due to the malformed HTML failures and programming language error web page cannot load properly in web browsers. Dynamic web application mainly suffered from execution failures generated from web page dynamically generated by the web server.

All Rights Reserved ©2014 International Journal of Computer Science (IJCS)

Published by SK Research Group of Companies (SKRGC).

International Journal of Computer Science
Oddity...Probe...Reviste...
http://www.ijcsjournal.com Volume 2, Issue 1, No 1, 2014. ISSN: 2348-6600
Reference ID: IJCS-033 PAGE NO: 176-181

In this paper execution failure are localized by using an automated framework called Pin Down. This study presents a randomized input generation technique for dynamic web application to check whether the web page is vulnerable to SQL injection or not. An automated random input generation is constructed for all executable statements in source code to determine the execution failures, such as missing include file, incorrect SQL query and uncaught exception of the corresponding statement. In addition, the framework used to determine the HTML failures such as generated HTML page is not syntactically correct according to an HTML validator by checking appropriate tags with closing tags using parsing the DOM tree. Other than execution and HTML failure novel framework Pin Down used to locate execution failures caused by deprecated language constructs that are all produce obstructive exception and error message during execution. In the existing system, failures are identified based on empty inputs. But it did not correctly predict the failures present in the source code. Further location of the fault identified using many statistical fault localization algorithms like Ochiai [1], [3], [11], Tarantula [1], [12], Jaccard [7], [13] algorithms. The proportion between passing and failing tests of executing statements are calculated. Suspiciousness rating is calculated for all executed statements for localize the faults present in a source code. The efficiency enhanced by source mapping and extended domain. These techniques are applied to the function call and conditional statements of a source code. But suspiciousness rating cannot be calculated whenever some statement is missing.

A. *Source Mapping and Extended Domain*

Execution fault localization in dynamic web application is a prominent problem of finding root location that causes failure in which source code changes need to be made in order to fix predicted faults. Existing system records the output of every statement executed. In source mapping technique PHP code translated into HTML code. HTML code validated against the HTML validator. Statistical fault localization algorithms assume the existence of test set. This approach is similar to combined concrete symbolic execution [9], [10]. The Extended domain [1] concept applied to statements present in a source code. The source code validated using APOLLO [9] system, first automated tool for finding bugs in web application. Suspiciousness rating calculated in extended domain concepts. Dynamic symbolic execution [9] only process conditional statements present in a dynamically generated source code. So the code coverage is less compared with other algorithms. Existing system [1] also uses APOLLO [9] for implementation of fault localization techniques.

III. PIN DOWN

A. *Objective*

The Fault localization algorithms explored in this paper attempt to predict the location of a bug based on randomized input values for server side execution in case of validation with empty inputs.

B. *Overview of Pin Down framework*

In this Pin Down framework execution failures are caused by a missing include file, an incorrect mysql query and uncaught exception in web application source code. These types of failures halt the execution and easily identified from error message generated by PHP Interpreter. In addition, the framework used to determine the HTML failures such as generated HTML page is not syntactically correct according to an HTML validator by checking appropriate tags with closing tags using parsing the DOM tree. Other than execution and HTML failure novel framework Pin Down used to locate execution failures caused by deprecated language constructs that are all produce obstructive exception and error message during execution. Server side vulnerability validated using the randomized input generation technique. This novel technique identifies dynamically generated web page is vulnerable to SQL injection or not. However the web page ha vulnerable gives suggestion to avoid such vulnerability in server side. Fixing execution failures by managing html entities, magic quotes, strip slashes, add slashes and auto load problems.

C. *System Architecture*

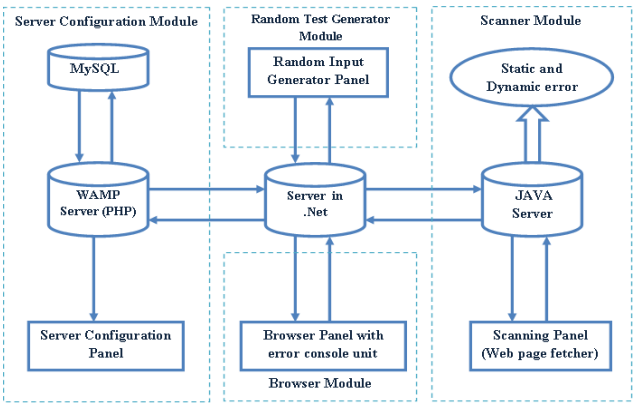


Fig 1. System Architecture

All Rights Reserved ©2014 International Journal of Computer Science (IJCS)
Published by SK Research Group of Companies (SKRGC).

International Journal of Computer Science
Oddity...Probe...Reviste...
http://www.ijcsjournal.com Volume 2, Issue 1, No 1, 2014. ISSN: 2348-6600
Reference ID: IJCS-033 PAGE NO: 176-181

IV. IMPLEMENTATION

We have implemented Randomized input generation for server side and client side input fields. Dynamic web application is combination of various scripting language. In this paper, pin down framework implements server side scripting language and predict the fault. Then compare the performance of the fault localization. To increase the efficiency of fault localization process, use CREST tool. Create a web application as an input for determine the bug identification. Finally implement pin down framework and execution bug identification algorithms such as Ochiai and Tarantula for compare results derived from this system.

Major components of pin down framework:

- Scanner panel
- Server Configuration panel
- Application Browser panel
- Random Input Generator panel
- Setting panel

A. Scanner Panel

Either html or PHP files can be scanned when it is placed in the scanner module and error will also be detected. This panel will detect browser compatibility issue (blink tag work only in internet explorer browser) resource missing, syntax errors. Pin down framework includes the scanner tool which provides static analyzer that checks the static bugs on given Input files like HTML as well as PHP. And also many files can be added or removed to scanning for detecting errors. And also it will display how many files selected.

B. Server Configuration Panel

Server configuration panel of Pin Down framework shows detailed configuration information present in the server. Based on the configuration detail, we can predict the vulnerability nature of dynamic web application. The values are retrieved from configuration files present on the local web server.

C. Application Browser Panel

Application Browser panel shows web page for given URL whenever the user of this framework present in the online. Otherwise the default page is displayed on Application Browser panel screen. The web page dynamically validated in this panel.

D. Random Input Generator Panel

The Random Input Generator panel is most important in our framework because server side vulnerability validated in this panel. First, it provides different number of forms, images and hyperlinks present in the web page. And also shows the exception message for wrongly entered value. The result generated by our framework shows the vulnerable nature of dynamically generated web pages. Pin Down framework also shows number iteration needed to detect dynamic bugs in Randomized input generation technique.

E. Setting Panel

In this Setting panel predefine the path for Fault localization. First field describes the localhost address value that should be entered when Application Browser Panel and Random Input Generator Panel displayed on the screen. It must be set to http://localhost. The second entry field has a root directory value of host and it set to c: /xampp/htdocs. Similarly php root directory set to c: /xampp/php. Finally the configuration file path is assigned in upcoming fields. Configuration file present in separate directories.

F. Implementation Procedure

In this section explains the procedure to localize execution fault present in web application using Pin Down framework.

Step 1: Add php file or html file one by one in scanner panel. It checks the web page statically for localize html failures and mild execution failures.

Step 2: Specify an URL in Random Input Generator Panel.

Step 3: If any errors in the given web page, then the errors will be displayed along with the error type and its line. Also displays number of hyperlink, image content, and forms.

Step 4: If no input field is found in the web page, then it simply displays a different number of forms, image content and hyperlinks on web pages.

Step 5: Select the Application Browser panel, then it will ask the URL. After URL given, it automatically displays the web page of the corresponding URL. After that we can dynamically validate the web page shown on the screen.

G. Data Flow Diagram

Figure 2 shows the DFD level 1 for suggested framework in which the function of fault localization process

All Rights Reserved ©2014 International Journal of Computer Science (IJCS)

Published by SK Research Group of Companies (SKRGC).

International Journal of Computer Science
Oddity...Probe...Reviste...
http://www.ijcsjournal.com Volume 2, Issue 1, No 1, 2014. ISSN: 2348-6600
Reference ID: IJCS-033 PAGE NO: 176-181

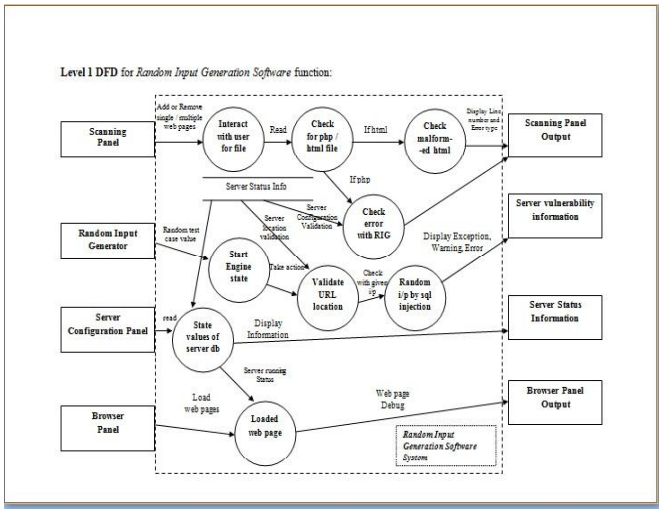
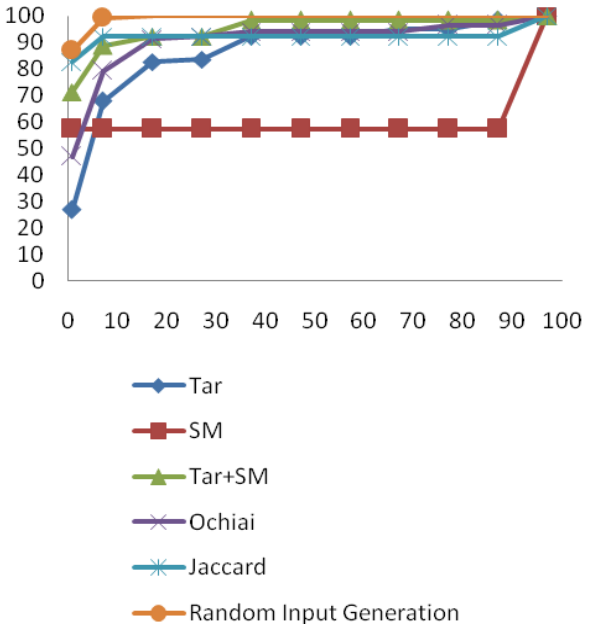


Fig 2. : Level 1 DFD for Pin Down framework

V. Result and Discussion

This section describes about the performance comparison of Pin Down framework. In general number of executed statements covered increase the code coverage (line of code covered) of source code present in dynamic web application. Compare Pin Down framework with other existing algorithms, Pin Down having high code coverage that is the total number of executed statements covered is high so that fault localization performance also increased. Ochiai algorithm have the next highest code coverage. Tarantula algorithm has moderate code coverage, but combined with source mapping code coverage equals to Ochiai algorithm. In the source mapping technique code covered is constant and less than other statistical algorithms. X axis show number of executed statement values. Y axis shows the code coverage against the coefficient values. Compared with manual fault localization process, it reduces the time needed for fault localization. Pin Down framework has better performance when compared to existing statistical fault localization algorithms. And also it increases the speed of fault localization process. Final result shows that code coverage improved from 95% to 100% were the result indicates 100% code covered is a reliable indicator of the effectiveness of a test case.



VI. Related Work

Literature review performed in the Software Engineering domain and the software testing process. The research articles by different authors are reviewed and some of concepts are described in this section.

A. Program Spectral based approach

A program spectral [3] is a collection of functional data that provides a specific view on the dynamic behavior of software systems. The coefficients are calculated based on the Passed and failed runs. The result is generated by evaluating the similarity coefficients of program code. This approach is a fully automated diagnosis of software faults and improves the efficiency of debugging and testing process. Program spectral is the classified as black box testing. Final result indicates Ochiai coefficient is best from others. It supports following tools

- Pinpoint [19]: It is a framework for root cause analysis on J2EE platform. Its preferred for dynamic web services.
- Tarantula [20]: Developed for 'C' language.
- AMPLE [21]: It is systems for identifying faulty classes in object oriented software.

All Rights Reserved ©2014 International Journal of Computer Science (IJCS)
Published by SK Research Group of Companies (SKRGC).

International Journal of Computer Science
Oddity...Probe...Reviste...
http://www.ijcsjournal.com Volume 2, Issue 1, No 1, 2014. ISSN: 2348-6600
Reference ID: IJCS-033 PAGE NO: 176-181

B. Accuracy of Spectrum fault localization

This paper [4] describes a program spectrum approach with reduced debugging effort. It is a light weight automated diagnosis technique. In this approach same Siemens test case benchmark used for all executable statements. The final result indicates a program spectrum is independent of test case design. Here observe the quality impact of similarity coefficient by measure observation quality, Varying coefficient factor and Revisit the coefficient again.

C. Execution Slices and Dataflow Tests

This paper [5] describes a tool which supports execution slice and dice based on test set. This tool reports the result using heuristic techniques in fault localization. Execution slice is a set of program basic blocks executed by a test inputs. Dice are a set of basic block in one execution, which does not appear during execution that is difference between two slices. In this we need to know that the fault resides in the slice part of a test suite which fails in its execution. The remaining code of the program is ignored in the fault localization process. Here we assume that the fault does not lie in the slice of the test, but it's present in the dice part. The statements present in the failed slice do not appear in the successful slice. It processes the code using the following tools.

- Chislice: Slicing and Dicing tool for ANSI or standard C language.
- Dataflow test tool: Evaluate the test adequacy using ATAC.

D. Directed Automated Random Testing

This paper [7] presents a new tool named DART for Testing the software automatically. It consists of following main techniques:

- Automated extraction of the interface present in a program using source code parsing.
- Automatic generation of test driver by random testing.
- Automatic generation of new test inputs to execute the statements.

Random testing is a simple and famous technique which can be effective at finding software bugs. But it provides low code coverage. DART framework automates the random testing by combining it with first main technique. It supports random testing as well as dynamic symbolic execution and alternative approach to static analyzers. Here the result was guaranteed by sound notification. It is overall complementary to static analyzer. It can also gather the knowledge dynamically so we call it as directed search. DART effectively implemented in 'C' program.

E. Automatically generate Inputs of Death

EXE [6] is an automated tool to generate input that crashes read code. This tool does not run the code manually where as it runs the code on symbolic input for initial condition. It forces the execution using feasible program path. It supports real code execution. It automatically generates its own test set. EXE easily find the bugs present in a source code. Execution Generated Test cases similar to EXE tool. But test case generated by executing a statement. EXE works in following systems.

- Udhcpd DHCP Server.
- Linux packet filter
- Pearl regular expression

F. Statistics based Techniques

Several statistical fault localization algorithms are available in fault localization process [1]. It only concentrates on conditional and functional call statements. The suspiciousness rating of each statement calculated, but it must be depend on statements present in the source code. Here code coverage is less. Time needed for localization of faults was high because we have to calculate the coefficient for passing and failing tests. So the same source code is tested at least twice. The efficiency of the fault localization algorithm improved by source mapping and extended domain.

G. Server side security Issues

Server side attacks [17] are mainly caused by SQL Injection, Cross site Scripting and User name Enumeration. SQL Injection is computer security vulnerability. In this information present in a database retrieved using inject SQL query to a particular database. Cross site scripting is a client site script attack and Username Enumeration is retrieving the information by entering wrong information in the entry field.

VII. CONCLUSIONS

In this paper, we introduce pin down framework for fault localization. This process is achieved by using Random Input generation technique. This method determined the situation of source code changes from the executed statement in the server side scripting language on web application. In future, the following methods are implemented with its increased code coverage.

All Rights Reserved ©2014 International Journal of Computer Science (IJCS)

Published by SK Research Group of Companies (SKRGC).

International Journal of Computer Science
Oddity...Probe...Reviste...
http://www.ijcsjournal.com Volume 2, Issue 1, No 1, 2014. ISSN: 2348-6600
Reference ID: IJCS-033 PAGE NO: 176-181

- Support for PEAR packages
- Finding .htaccess vulnerabilities.
- Identify SQL injection vulnerabilities in PHP applications.
- Debugging against XSS attacks

REFERENCES

[1] S. Artzi, "Fault localization for dynamic web applications," J. Dolby, F.Tip, M. Pistoia., IEEE Transl, vol. 38, no 2, [March or April 2012], pp.314–335

[2] C.P.Shabariram, "Fault Localization for Dynamic Web Application : A Survey on Recent Developements." International conference on Knowledge Collaboration in Engineering [2014].

[3] R. Abreu, "An evaluation of Similarity Coefficient for software fault localization," P. Zoeteweij, A. J. C. Vangemund., International symposium on dependable computing[2006], pp.39–46.

[4] R. Abreu, "On the accuracy of spectrum based fault localization," P. Zoeteweij, A. J. C. Vangemund., Conference, [sept 2007], pp89–98.

[5] H.Agrawal, "Fault localization using execuion slice and dataflow tests," J. R. Horgan, S. London, W.E. Wong., International symposium on software reliability engineering[1995], pp.143–151.

[6] C. Cadar, "EXE: Automatically generating the inputs of death," V. Ganesh, P. M. Pawlowski, D. L. Dill, D. R. Engler., Conference on computer and communication[2006].

[7] P.Godefroid, "DART: Directed automated Random testing," N. Klarlund, K.Sen., Conference on programming language design and Implementation[2005].

[8] S.Horwitz, "Interprocedural slicing using dependence graph," T. Reps, D. Binkly., ACM Trans on programming languages and system [1990].

[9] J. Lyle, "Automated Bug localization by program slicing," M. Weiser., Second International conference on computer and applications[1987].

[10] S. Artzi, "Directed test generation for effective fault localization," J. Dolby, F.Tip, M. Pistoia., International symposium on software testing and analysis[2010].

[11] S. Artzi, "Practical fault localization for dynamic web applications," J. Dolby, F.Tip, M. Pistoia., International conference on software engineering[2010].

[12] S. Artzi, "Finding bugs in web application using dynamic test generation and explicit state model checking," J. Dolby, F.Tip, M. D. Ernst, A. Kiezun, D. Dig, A. Paradkar., IEEE Transl on software engineering, vol. 38, no 2, [march or april 2010], pp.274–294.

[13] P. Arumuga nainar, "Statistical debugging using compund boolean predicates," T. Chen, T. Rosin, B. Libit., International symposium on software testing and analysis[July 2007].

[14] S. Artzi, "A framework for automated testing java script web applications," J. Dolby, F.Tip, A. Mollor, S. Jensen., International conference on software engineering[2010].

[15] B. Baudry, " Improving test suites for efficient fault localization," International conference on software engineering[2006].

[16] Y. Yu, "An emprical study of the effects of test suite reduction on fault localization," International conference on software engineering[2008].

[17] C.P.Shabariram et al, "Novel Dynamic Fault Localization for Server side Vulnerabilities." International conference on Global Innovations in computing Technology [2014].

[18] C.P.Shabariram et al, "Pin Down : Fault Localization in Large Dynamic Web Application." International conference on Knowledge Collaboration in Engineering [2014].

[19] M. Y. Chen et al, " Pinpoint: Problem determination in large,dynamic internet services. In Proc. DSN'02, pp. 595–604, Washington, DC, USA,[2002].

[20] J. A.Jones et al, "Visualization of test information to assist fault localization. In Proc. ICSE'02, pp. 467–477. [2002].

[21] V. Dallmeier et al, " Lightweight defect localization for Java. In Proc. ECOOP'05, UK, [2005].

All Rights Reserved ©2014 International Journal of Computer Science (IJCS)

Published by SK Research Group of Companies (SKRGC).