

Role of Automated GUI Testing in Secure Software Development of Indian Spacecraft Ground Software, GEOSCHEMACS

K. V. Maruthi Prasad¹, J. Krishna Kishore²

^{1,2} ISRO Satellite Centre, HAL Airport Road, Vimanapura P.O, Bengaluru-17, Karnataka, India

Abstract: *Inadequate attention to secure software development can make any software susceptible for attacks and runtime malicious errors. Inherent security vulnerability with reference to input invalidation was one of the main software security vulnerabilities and required to be tackled. GUI enabled software needs more testing combinations for finding input data vulnerabilities. GEOSCHEMACS (Geostationary Earth Orbit SpaceCraft Health Monitoring, Analysis and Control Software) is Indian Spacecrafts main ground software element and contains GUI presentation & analysis packages. Objective of this paper was to highlight the importance of GUI based applications test automation in achieving secure GEOSCHEMACS. It presents the excerpts of the tool developed for automating the X/Motif applications GUI testing. This paper brings out the results of the experiments carried out with reference to role of GUI test automation in attaining secure software as part of the research being pursued on “Secure Software for Indian Spacecraft Ground Software Elements, GEOSCHEMACS”.*

Keywords: GUI Testing, Secure Software, Automation, GEOSCHEMACS.

1. Introduction

Ground software has been used for monitoring and controlling the health of satellites. GEOSCHEMACS is ISRO's (Indian Space Research Organisation) in-house built major ground software package and primarily developed using C, X/Motif and Java, HTML, JavaScript. It is a package of multiple software elements based on client/server distributed computing architecture and of over one million lines of source code.

Vulnerabilities in the software are the weaknesses or flaws that can be exploited by the attackers. Security vulnerabilities that have been left in the software during development and deployment were the main causes behind exploitation. Flaws in the software can cause hang or crash, can cause software to act in unintended way. Software security vulnerabilities such as buffer overflows, invalidated input, race conditions, insecure file operations, access control problems are some of them to be highlighted. Insecure mission ground software elements can become a reason in the disturbance of space services.

Secure software is the output of SSDLC (Secure Software Development Life Cycle) phases which possess the capacity of eliminating most of the vulnerabilities in source code during development and with the ability of withstanding malicious attacks by functioning correctly. Towards secure development of GEOSCHEMACS, the vulnerabilities related to GUI invalidated input were also required to be handled. X/Motif GUI has been the predominant user interface in GEOSCHEMACS operational environment.



Figure 1 GEOSCHEMACS operational environment

This paper presents the role of GUI test automation in achieving secure GEOSCHEMACS. The paper has been divided into sections titled Introduction, Background, Necessity of GUI test automation, How to achieve GUI test automation?, Software for X/Motif applications GUI auto testing, Role of test automation in secure software development, Results and Conclusion.

2. Background

Malicious data input software vulnerability [2] is one of the most likely vulnerabilities to surface in software. Many software vulnerabilities [2] result from unexpected behaviours triggered by a program's response to malicious data. It is required to be addressed how and where user-malleable data gets accepted by the software system. Nominally the data input is either through command line or specific GUI (Graphical User Interface) or through web request. Interfaces are the mechanisms [2] by which software components communicate with each other and the users. Software often places too much trust in its communication

peers and makes assumptions about the data's potential origins & contents. Attackers try to explore every accessible interface to a piece of software and look specifically for any assumptions the developer made.

GEOSCHEMACS has various software components for spacecraft health presentation and analysis and also for control. All these GUI rendered applications are mostly based on X/Motif, Java and HTML & Java Script. As part of the research titled "Secure Software for Indian Spacecraft Ground Software Elements, GEOSCHEMACS" and towards secure GEOSCHEMACS, it is necessary to handle and validate the data input properly to avoid malicious input into the system. It is not too difficult for an insider who wants to exploit the invalidated input of the software components. The malicious data input may cause the software to crash or / and unavailable.

3. Necessity of GUI test automation

GUI-based software application is tested by performing sequences of events on GUI widgets. Correctness of the software is determined by examining the state(s) of the GUI widgets. To reasonably [1] verify the system's functionality; a large amount of testing is required. Performing these tests manually is costly and can be practically impossible. Therefore, it is necessary to perform automated testing. Automation testing is a process of writing a computer program, to do testing that would otherwise need to be done manually. Too many repeatable tests are challenge to man. It doesn't matter for machine to do so. That's main reason [1] that automation of GUI testing is required. Another reason is for regression test purpose. The idea of software test automation is to let computer simulate what human do when manually running a test on the target application. The test scripts once written or developed for the application or the software under test can be run repeatedly as per the requirement. Also, it is a quick and efficient process without manual intervention [1].

Test automation [1] can enable some testing tasks to be performed more efficiently than by testing manually. Automation of testing makes the effort involved in performing regression tests at minimal. GUI based applications test automation allows the tester to run more tests in less time and also to execute them more often. Automation of GUI based application testing enables us to execute test cases of input entry with greater accuracy, run difficult or impossible test cases to do manually. GUI applications test automation gives increased confidence on the software under test. Test automation ensures the consistency & repeatability of tests and reuse of tests. Test automation reduces costs and increases the quality of the testing tasks [1].

Automated software testing is preferable in GUI based applications, since test case execution occur high number of times. One more important reason is that most of the GEOSCHEMACS presentation and analysis software elements share a common and similar interface, test automation is quite essential. To reduce the testing time for a

given version, GUI test automation is very helpful. Manual testing is often error prone and there are chances of most of the test scenarios left out. Implementing automated GUI testing for the application early in the software development cycle speeds up development, improves quality and reduces risks towards the end of the cycle. The important benefits include, higher test coverage levels, greater reliability, shorted test cycles, ability to do multi user testing at no extra cost, increased level of confidence in the software.

4. How GUI test automation?

Automated GUI Testing is use of software program and scripts to detect if the GUI application is functionally correct [6]. Automated GUI Testing involves carrying set of tasks automatically and comparing the result of same with the expected output and ability to repeat same set of tasks multiple times with different data input and same level of accuracy. GUI Testing includes how the application handles keyboard and mouse events, how different GUI components like menu bars, toolbars, dialogs, buttons, edit fields, list controls, etc. reacts to user input and whether or not it performs as per the requirements.

A test automation framework is the process being used to implement automation. Several test automation frameworks such as record & playback, capture & playback, keyword driven have been applied by various organizations. Different tools are available both at commercial and open access level for GUI test automation. Majority of the tools are based on record & playback framework. However, record & replay technique is labor intensive and expensive to generate and manage entire test cases. GEOSCHEMACS presentation & analysis packages contain various technologies include X/Motif, Java, Web. Majority is rendered with X/Motif with supporting applications of Java & Web technologies.

No single GUI Test automation software can accommodate for covering all the above mentioned technologies. Different tools for different technologies are required. Keyword based GUI test automation software would enable the capacity of automatic GUI testing for all required and possible test case combinations with low maintenance and better readability of test scripts or test input files. For all reasons, a customized and easy to operate GUI test software for each of the technology is desirable. Towards that, GUI test automation software for X/Motif applications, Java GUI test automation software, Web applications GUI automation tool are necessary. X/Motif GUI test automation software has been accomplished as part of the research "Secure Software development for Indian Spacecraft Ground Software, GEOSCHEMACS".

5. GUI Auto Testing Software

Automation software such as QF-Test, are primarily record and playback technique based and can be used for GUI including Web applications except X/Motif. A tool from Xaccessory provides auto testing of X/Motif on capture & replay basis. Since the idea is to spend minimum time in executing maximum test combinations with highest accuracy,

keyword approach based GUI test automation software is necessary. One such software has been realized for testing X/Motif applications. Similar attempts are under the way in Java and Web GUI test automations.

5.1 Software for X/Motif GUI test automation

Approach of mouse buttons and key board key input virtualization along with the test case input file(s) generated based on keywords and the corresponding input against keyword has been the concept behind this software. "XSendEvent" Xlib routine has been used in executing the test cases automatically by sending Xevents to the identified X window identifier. Test case input contains the X-window identifier for which the Xevents required to be sent. Editres Protocol has also been utilized in getting the widget hierarchy and window identifiers. The test case results have been auto verified with the state and presence & absence of corresponding GUI widgets. This test automation software is able to recognize the components of GUI of the AUT (Application Under Test), able to send GUI Xevents such as mouse clicks and text field input for the required window identifiers and finally able to test the functionality underlying a GUI set of components.

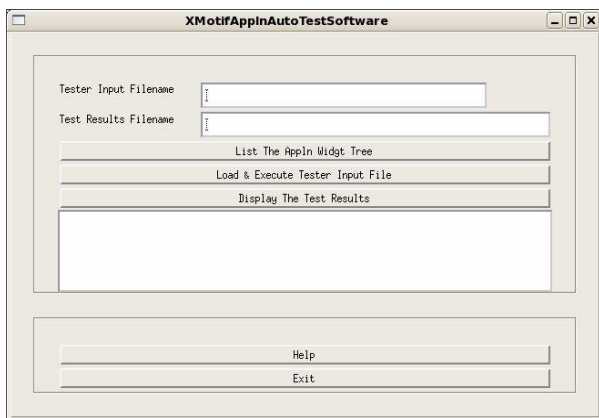


Figure 2: Main GUI of test automation software

6. Role of GUI test automation in secure software development

An entry point is a place where input is supplied to the software. Secure software must assume any input that comes from any entry points is potentially malicious. User interface and command line argument input are among the common entry points along with many others. Entry points are the optimal places for exploitation by any attacker or insider to ultimately break the software or crash or not available. Malicious data input through any user interface can even trigger buffer overflow. Input validation for all the entry points is one of the principles of secure software development which would be practiced across all the phases of the life cycle. Most of the input invalidation vulnerabilities would be cropped in implementation or coding phase.

Secure static analysis code review is one of the vital activities required to be practiced in achieving secure software. However, software testing at unit, integration and acceptance levels with embedded security concerns in test plans can only give confidence and quality about the

software. Maximum number of test case combinations shall be applied over the software. The number of test case combinations are much high in the case of user interface testing. More number of test cases would enable to find out the cases of input invalidation. Tests of mechanical nature of different data ranges and malicious input will be very difficult in executing manually without error and in the available time and cost. Regression tests of the same would cause effect of difficulty under manual testing. So, GUI test automation is essential. Approach of GUI test automation at unit, integration, acceptance and regression levels would be very useful in attaining secure GEOSCHEMACS.

In addition to testing normal functional aspects of software, it is essential to test for the abuse and abnormal cases also in secure software development. Developers are more inclined to trusting of input and data. Input contents, input formats, input sources and data submission formats are assumed trustworthy in most of the cases [3]. With reference to web applications, hidden form fields are not validated in majority cases. It is required to accommodate test cases for validating every GUI input with reference to 0, 1, maximum / minimum size of the data type, maximum size of the data type + 1, minimum size of data type - 1, exploration of format details & length. It is important to include some of the format parameters in GUI input. It is very much necessary to test for long strings, largest and smallest allowable values in numeric fields for each GUI. Fuzz testing is required to find out security vulnerabilities by unexpected and malformed data to program's entry points. These all reasons would make automated GUI testing is very crucial in secure software development.

7. Results and Conclusion

The results of GUI test automation with reference to Java and Web technologies are to be tabulated and the process is under the implementation. X/Motif GUI test automation has been well accomplished through the above mentioned GUI test automation software in section 5. All X/Motif GUI components covering various widget types such as push / toggle / arrow button, option / pull down / pull right menu, radio / check / combo box, scroll list, text widget, main window have been well accommodated in GUI test automated testing. Test input files of capacity ranging from 1 to 1000 test cases have been used. Many applications under test were developed to cover all types of widget and applied through GUI test automation software.

Many of the GEOSCHEMACS presentation and analysis packages were tested automatically through this test automation software. The results were very much in the line of expected output. This X/Motif GUI test automation software is very helpful in covering many of the test cases which could not be done manually. It is very useful in regression testing and finding runtime security vulnerabilities. The system could complete all the test cases of the given tester input file in very short time span and could log each test case result. Preparation of tester input files was very easy and not dependent on any scripting or programming language. The system was helpful in auto

testing at all levels of testing including unit testing.

Automated GUI testing is a powerful tool when it is used with intelligence and skill. Good culture and software process practices are necessary to make test automation successful. The same is true with secure software development also. Test automation shall be integrated as part of secure software development life cycle with proper tester input files and well trained people in using test automation software. Testing of multiple times such as regression and smoke testing and requirement of more test case combinations would enable test automation yield excellent results.

References

- [1] K.V.Maruthi Prasad, "A Distinct Approach for X/Motif Application GUI Test Automation," International Journal of Research in Engineering and Technology, Volume 3, Issue 5 May 2014, pp. 270-275, 1992.
- [2] Mark Dowd, John McDonald, Justin Schuh; "The Art of Software Security Assessment: identifying and preventing software vulnerabilities", Addison-Wesley, Boston, July 2012.
- [3] Maura A., Van der Linden; "Testing Code Security", Auebach Publications, 2007.
- [4] Software Test Automation: Effective use of test execution tools by Mark Fewster & Dorothy Graham; Addison-Wesley.
- [5] "Test Automation" by Macario Polo, Pedro Reales, Mario Piattini and Christof Ebert, IEEE Software, January/February 2013.
- [6] "Test Automation vs Manual testing in software development" Available: <http://www.embedded.com/electronics-blogs/say-what-4439669/Test-Automation-vs-Manual-Testing-in-Software-Development>. [Accessed on July 31, 2015].
- [7] The definitive guides to the X window system (Volume One): Xlib Programming Manual for Version 11 by Adrian Nye; O'Reilly & Associates, Inc.
- [8] Graphical User Interface with X-windows and Motif By Muralidhar R. Rao & Ganga Prasad G.L; Wiley Eastern Limited.
- [9] K.V.Maruthi Prasad, J.Krishna Kishore; "Testing Open Source Static Security Vulnerability Scanners for CERT C Secure Coding Rules Compliance", International Journal of Advanced Research in Computer Science and Software Engineering; Volume 4, Issue 12, December 2014.
- [10] The definitive guides to the X window system (Volume Four): X Toolkit Intrinsic Programming Manual for OSF / Motif 1.2 Edition by Adrian Nye and Tom O'Reilly; O'Reilly & Associates, Inc.
- [11] The definitive guides to the X window system (Volume Six A): Motif Programming Manual for OSF / Motif 1.1 Edition by Dan Heller; O'Reilly & Associates, Inc.
- [12] Stephen de vries, "Security testing web applications through automated software tests", OWASP Europe conference, Leuven, Belgium, May 30-31 2006.