Dynamic Binary Instrumentation Technology Overview

Kunping Du
National Digital Switching System Engineering & Technological Research Center
Zhengzhou, China, 0371-81632000

Fei Kang
National Digital Switching System Engineering & Technological Research Center
Zhengzhou, China, 0371-81632000

Hui Shu
National Digital Switching System Engineering & Technological Research Center
Zhengzhou, China, 0371-81632000
hnzzdkp@163.com

Li Dai
National Digital Switching System Engineering & Technological Research Center
Zhengzhou, China, 0371-81632000

***Abstract*—The Dynamic Binary Analysis technology is a newly emerged technology which can analysis program execution dynamicly. Using this technology, the process of program analysis became more simple and accurate. Foreign researchers had put forward several Dynamic Binary Analysis Platform in recent 10 years. Based on these platforms, users can easily build useful analysis tools which satisfy their own needs. This paper introduces five most representative Dynamic Binary Analysis platforms first. Then, four significant fields and existing applications closely related with Dynamic Binary Analysis technology are explored. In the end of this paper, the feature research hot spots are discussed.**

Keywords- Dynamic Binary Analysis, program analysis technology, Dynamic Binary Instrument

I. FOREWORD

Dynamic Binary Analysis[1](DBA) technology is a kind of dynamic program analysis method which can analyze program's memory structure and add specific instructions for monitoring and testing program's execution.The DBA technology enables users to monitor program's behavior under the premise of not affecting the results of program execution by inserting additional appropriate analysis code into the target program, this procedure called Dynamic Binary Instrument(DBI).In addition,using DBA technology,the analysis can complete without source code, no need to recompile and link,so that this technology can be used in many cases.The research on DBA technology began in the 1990s, initially applied to the dynamic optimization and testing of the program.Due to its versatility and accuracy of the analysis process,it has been used for memory testing,software behavior monitoring,reverse engineering and some other research areas recently..

This paper first introduces five most widely used DBA platform,they are Shade, DynamoRIO, Valgrind, Pin and Nirvana. On this basis, summarizes the application status and popular tools build on DBA platform in the field of memory testing and optimization, data flow tracking, software behavior analysis, reverse engineering and parallel program analysis. Finally, the application prospects of DBA technology are discussed.

II. DYNAMIC BINARY ANALYSIS PLATFORM

So far, the foreign researchers had put forward a number of DBA platform, such as Shade, DynamoRIO, Valgrind etc. Based on these platforms, users can easily develop their own Dynamic Binary Instrumentation(DBI) tool. Below, we will detail the Shade, DynamoRIO, Valgrind, Pin, and Nirvana.

A. Shade[2]

It is the first the DBI platform which implements in Solaris system. Shade uses binary translation and cache technology, it has inner support of recording the register state and opcode information..

B. DynamoRIO[3]

DynamoRIO is an open-source dynamic binary optimization and analysis platform which evolves from Dynamo. It is available both in Windows and Linux system, and can record the execution instruction information efficiently, but doesn't support data flow recording. This platform is mainly used for the dynamic optimization of program in instruction level.

C. Valgrind[4]

An open source DBI platform under Linux which can efficiently record the instructions flow and data flow information of executable file in Linux. But because of the different operation mechanism of Linux and Windows system, this platform is still difficult to transplant to Windows system.

D. Pin[5]

Pin is a dynamic binary instrumentation framework for the IA-32 and x86-64 instruction-set architectures that enables the creation of dynamic program analysis tools. The tools created

using Pin, called Pintools, can be used to perform program analysis on user space applications in Linux and Windows. Pin provides a rich API that abstracts away the underlying instruction-set idiosyncrasies and allows context information such as register contents to be passed to the injected code as parameters. Pin automatically saves and restores the registers that are overwritten by the injected code so the application continues to work. Limited access to symbol and debug information is available as well. Pin was originally created as a tool for computer architecture analysis, but its flexible API and an active community (called "Pinheads") have created a diverse set of tools for security, emulation and parallel program analysis. Pin is proprietary software developed and supported by Intel and is supplied free of charge for non-commercial use. Pin includes the source code for a large number of example instrumentation tools like basic block profilers, cache simulators, instruction trace generators, etc. It is easy to derive new tools using the rich API it provides.

E. Nirvana[6]

Microsoft's latest development DBI platform, mainly includes two key module: program simulation execution module and JIT (just in time) binary translation module. But it has not been to market, only for Microsoft internal use. According to relevant data, the platform can well support tracking and playback function of Windows executable files in instruction level. There will be very good application prospects especially in software reverse engineering.

III. DBI APPLICATION FIELD

A. Memory testing and optimization

DBI framework developed up to now, the most widely used application is for the building of memory monitoring tools. DBI-based memory testing tools have obvious advantages than the common memory detection tool in the detection efficiency and detection accuracy, as well as the support of the underlying system. Therefore, there have been a lot of DBI based memory monitoring tools since DBI technologies emerged. Most of those tools can not only detect the memory using situation of a program, memory errors that may exist in the program, illegal use of memory, memory leaks, but also can detect buffer overflow accurately. The following details on several of BDI-based memory monitoring tools and related research.

a) Memcheck: Memcheck is a memory error detector based on Valgrind. It can detect many common problems appear in C and C++ programs, such as: accessing memory you shouldn't, using undefined values, incorrect freeing of heap memory, memory leaks etc.

b) Dr.Memory: Dr. Memory is built on the open-source dynamic instrumentation platform DynamoRIO. It is an excellent memory checking tool that supports both Windows and Linux. Dr. Memory uses memory shadowing to track properties of a target application's data during execution. So that it can detect memory error more accurately. In addition, Dr. Memory provide two instrumentation paths: the fast-path and the slow-path. The fast-path is implemented as a set of carefully hand-crafted machine-code sequences or kernels covering the most performance-critical actions. Fast-path kernels are either directly inlined or use shared code with a fast subroutine switch. Rarer operations are not worth the effort and extra maintenance costs of using hand-coded kernels and are handled in the slow-path in C code with a full context switch used to call out to the C function. Through using different path in different situation, the efficiency of detection is increased greatly.

B. Dynamic Taint Analysis

The dynamic taint analysis technology is a common technique in the field of application security detection. By analysis of the data used in the program, the program's data is marked as "contaminated"(Tainted), and "not contaminated" (UnTainted) categories, while in the process of implementation of the procedures to control the spread of contaminated properties by analyzing the illegal use of the data propagation path of the contaminated property to find the loopholes that exist of the program. DBI based platform, you can build a dynamic data flow tracking tools, such data flow tracking tool with a wide tracking range, and analysis results are accurate. Here are two methods based on DBI data flow tracking tool.

a) TaintCheck: TaintCheck is a dynamic taint analysis tool based on Valgrind, for the automatic detection, analysis, and signature generation of exploits on commodity software. TaintCheck's default policy detects format string attacks, and overwrite attacks that at-tempt to modify a pointer used as a return address, function pointer,or function pointer offset. Its policy can also be extended to detect other overwrite attacks, such as those that attempt to overwrite data used in system calls or security-sensitive variables. TaintCheck gave no false positives in its default configuration. in many cases when a false positive could occur, it is a symptom of a potentially exploitable bug in the monitored program. For programs where the default policy of TaintCheck could generate a false positive. Once TaintCheck detects an overwrite attack, it can automatically provide information about the vulnerability and how the vulnerability is exploited. By back-tracing the chain of tainted data structure rooted at the detection point, TaintCheck automatically identifies which original flow and which part of the original flow have caused the attack.

b) Dytan: A Generic dynamic taint analysis framework based on Pin. The goal of this tool is to be a generalized tainting framework that can be used to perform dataflow and control-flow analysis on an x86 executable. The dynamic tainting of Dytan consists of: (1)associating a taint label with data values;(2)propagating taint labels as data values flow through the program during execution.As long as user provides XML configuration file, in which specify: taint sources, propagation policy, and sinks.

C. Reverse engineering application

Dynamic tracking is one of the commonly used method in reverse engineering. The procedure of dynamic tracking is like this: using dynamic debugging software (eg: OllyDebug) load the program, then follow the tracks of program execution

step-by-step. This approach can be summarized in a word: analysis when tracking. And the analysis relies heavily on manual, it is difficult to automate it. By means of DBI platform, one can separate the analysis work to the tracking process by using DBI tool recording the execution information of target software, analyzing the recorded information by other automatic tools. Such processing procedure can save a lot of human labor. And the automatic analysis of the recorded information also can greatly reduce the software reversing cycle.

In 2008 blackhat Danny Quist. etc propose a DBI based temporal reverse engineering. By DBI platform Pin, they get the basic block execution sequence. By analyzing and visualizing these block information, it help analyst understand the program behavior quickly. In addition, in reference[7], the author proposed a DBI based protocol reverse method, the main idea of the paper is recording the data-flow of a software with DynamoRIO, then parse the protocol field with their own automatic tool.

D. Parallel program analysis[8]

With the development of high performance computing technology, the design of parallel programs is becoming increasingly important. Parallel debugging and performance evaluation of parallel programs are difficult problems in the field. The traditional Parallel debugging and performance evaluation tools are mostly based on source code instrumentation, which makes the workload of analyzing parallel programs very huge, and as the coding language and software upgrade, testers need to do some modifications. The most deadly is if you can't get the source code of the parallel program, the test can't be conducted. DBA technology making the analysis of parallel programs has nothing to do with the source code, the analysis process is more transparent and more efficient. The following is several parallel program analysis tools based on DBI framework.

a) Intel Parallel Inspector: The Intel Parallel Inspector analyzes the multithreaded programs' execution to find memory and threading errors, such as memory leaks, references to uninitialized data, data races, and deadlocks. Intel Parallel Inspector uses Pin to instrument the running program and collect the information necessary to detect errors. The instrumentation requires no special test builds or compilers, so it's easier to test code more often. Intel Parallel Inspector combines threading and memory error checking into one powerful error checking tool. It helps increase the reliability, security, and accuracy of C/C++ applications from within Microsoft Visual Studio.

b) CMP$im: Memory system behavior is critical to parallel program performance. Computational bandwidth increases faster than memory bandwidth, especially for multi-core systems. Programmers must utilize as much bandwidth as possible for programs to scale to many processors. Hardware-based monitors can report summary statistics such as memory references and cache misses; however, they are limited to the existing cache hierarchy and are not well suited for collecting more detailed information such as the degree of cache line sharing or the frequency of cache misses because of false sharing. CMP$im uses Pin to collect the memory addresses of multithreaded and multiprocessor programs, then uses a memory system's software model to analyze program behavior. It reports miss rates, cache line reuse and sharing, and coherence traffic, and its versatile memory system model configuration can predict future systems' application performance. While CMP$im is not publicly available, the Pin distribution includes the source for a simple cache model, dcache.cpp.

IV. FUTURE RESEARCH

DBA technology as a new program analysis method, have not yet been widely used. As people get more comprehensive understanding on its properties and advantages, it will play a role in more areas in more fields. Future research on dynamic binary analysis techniques are mainly concentrated in the following aspects:

*a) Improvement of performance for DBI platform:*Based on DBI build tools have a common weakness: a certain degree of reduction on efficiency to instrumentation program. In general, the use of DBI make the original program run rate 3-5 times lower, in future studies, how to improve the performance and efficiency of the DBI platform is an important research direction.

*b) The combination of static analysis methods:*DBA method has many advantages, but it is essentially a dynamic analysis method that can not overcome the shortcoming of only one execution path can be passed by a time. In the future, how to combine the dynamic binary analysis with the static analysis methods is a future research focus.

c) solve the problem of huge amount of record information: Using DBI instrument a program ,weather in instruction level or function level, the record set could be very huge. How to reduce the volume of the record set in the premise of ensure enough information, how to improve the efficiency of information processing, how to visualize those information are all the research spot in the future.

DBA technology, with the advantages of extensive (needn't source code) and accuracy (run-time instrument), has already come to the forefront in several areas, and provides new idea to solve the problems in related field. The DBA technology would bring more breakthrough for more field in the future.

REFERENCES

[1] Nicholas Nethercote. Dynamic Binary Analysis and Instrumentation or Building Tools is Easy [D]. PhD thesis. University of Cambridge, 2004.

[2] Bob Cmelik and David Keppel. Shade: a fast instruction-set simu lateor for execution profiling [R]. In:ACM SIGMETRICS, 2004.

[3] Derek L. Bruening. Efficient, Transparent, and Comprehensive Runtime Code Manipulation [D]. PhD thesis, M.I.T, 2004. http://dynamorio.org/.

[4] N.Nethercote. Valgrind: A Framework for Heavyweight Dynamic Binary Instrumentation [C]. In:Proceedings of the 2007 ACM SIGPLAN conference on Programming language design and implemention, San Diego,California,USA: 2007. 89-100. http://www.valgrind.org.

[5] Chi-Keung Luk. Pin: building customized program analysis tools with dynamic instrumentation [C]. In:Programming Language Design and Implementation. 2005: 190-200.

[6] Sanjay Bhansali. Framework for Instruction-level Tracing and Analysis of Program Executions [C]. Second International Conference on Virtual Execution Environments VEE, 2006.

[7] HE Yong-jun, SHU Hui, XIONG Xiao-bing. Network Protocol Reverse Parsing Based on Dynamic Binary Analysis.[J]. Computer Engineering. 2010.36(9):268-270

[8] Moshe Bach, Mark Charney, Robert Cohn, etc. Analyzing Parallel Programs with Pin. [J]. IEEE Computer. 2010:34-41.