Kernel-level dynamic binary instrumentation method using binary translation

Dongwoo Lee, Inhyuk Kim, Changwoo Min and Young Ik Eom
School of Information and Communication Engineering, Sungkyunkwan University
Gyeonggi-do 440-746, Korea
[e-mail: {cowsboys, kkojiband, multics69, yieom}@ece.skku.ac.kr]
*Corresponding author: Dongwoo Lee

Abstract

Binary translation is a kind of the emulation method which converts a binary code for the particular instruction set architecture (ISA) to the new one that can be run on another ISA. It is mostly used for migrating legacy systems to new architecture. However, because it enables instrumenting programs dynamically, binary translation on the same architecture is also useful. In user-level, there already exists some instrumentation software using binary translation, such as dynamic binary analyzers (DBAs) like Valgrind and PIN, and virtual machine monitors (VMMs) like VMWare, VirtualPC and Bochs. On the other hand, in order to be benefited from binary translation in kernel-level, a few issues, which include system performance, memory management, privileged instructions, calling convention and some kernel functions, should be treated. These matters are derived from the structure of the kernel, and the difference between the kernel and user-level application. In this paper, we present a scheme to apply Binary Translation on kernel-space. Moreover, we demonstrate that binary translation, whether we implement it on Linux kernel, adds an insignificant overhead to performance of the system.

Keywords: Binary Translation, Dynamic Binary Instrumentation, Linux Kernel

1. Introduction

Binary translation[1] is a kind of the emulation method which converts a binary code for the particular ISA to the new one that can be run on another ISA. It is commonly used as migration tools for transitting from old architectures to newer ones.

On the other hand, it is also useful to do binary translation on the same architecture. It enables that the additional code, especially purposeful code[2], is inserted to the original code of the target program, or the original code is replaced by others at run-time. This is convinent for users, as it does not require any preparation. Also, it gives the full of instrumentation coverage of user-mode program, without requiring source code.

In user-level, there already exist many DBI programs, such as DBAs and VMMs. In DBA tools like Valgrind[3], Pin[4] and Dynamo[5], and, as adding the analysis code by using DBI, it is possible to analyze programs at the level of machine code without recompiling or relinking. Also in VMM like VMware[6] and VirtualPC[7],

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology (2010-0022570)