

A Clean Environment for Web Applications Using Lightweight Virtualization

Yih Huang, Jiang Wang, Angelos Stavrou, Anup Ghosh

Abstract:

Vulnerable applications bring significant risks for the operating system, which may host sensitive data and access to other sensitive systems. Currently, most network applications and non-networked applications run side by side on a single operating system. Once a network application is compromised by an outside attacker, the whole operating system is “owned” by the attacker, and often unbeknownst to the system owner. As a result, people may lose sensitive data such as passwords, credit card numbers, and on-line banking credentials. Anti-virus software detects the virus, worm and other malware largely by signatures, so they are largely ineffective in detecting zero-day attacks. Intrusion detection systems face similar problems in being reactive in detecting threats.

Alternative approaches that use sandboxing or privilege-based access control, such as IE7 can limit the damage to the operating system; however, it cannot prevent malware that can escalate its privileges. Another method to protect network applications is using hardware virtualization to separate applications from other parts of the operating system. Running network applications in a virtual machine provides root security for the host operating system as long as the virtual machine monitor’s integrity remains intact. But hardware virtualization introduces heavy overhead, because each virtual machine has its own operating system.

In order to balance the isolation and performance, we use operating system level virtualization to isolate every instance of user applications, and use a stackable file system to separate the system and application binaries from user’s data. Operating system level virtualization has a better performance because it does not virtualize the hardware, but shares the operating system kernel with many virtual environments. By using operating system virtualization, we can isolate every instance of network applications while maintaining good performance. In addition, we use a stackable file system, unionfs, to separate system binaries from user data.

We create the virtual environment on-demand for each instance of an application when it is opened. After termination, our system removes all the changes to the virtual environment. Our goal is to run every instance of applications, such as web browser and office productivity software and media players in an isolated container and protect the basic binaries of virtual environment from being changed by malware.

We implemented a prototype to demonstrate our approach; the results show that it has good scalable performance. The prototype is based on OpenVZ — an operating system level virtualization engine for Linux — and unionfs file system. We first created a base operating system which contains minimal operating system binaries and a web application – the Konqueror browser in our implementation. However, our architecture can be used for any application, such as work processing software and email client. We mount the basic operating system as a read-only layer of unionfs plus another empty writable layer together to construct a new virtual environment. Since the basic operating system is mounted as read-only, it will never be changed by the software in the virtual environment. All the writes to the file system go to the writable layer. By this means, we can easily remove the whole writable layer after the application is closed, while not causing faults for the applications. Moreover, the read-only layer of basic operating system can be mounted to multiple virtual environments at the same time, saving a lot of disk space. We compared the performance of OpenVZ virtual environment with VMware Workstation and native Linux.