

# Internet Cleanroom: a system using on-demand virtualization to enhance client-side security while keeping usability

Jiang Wang, Anup K. Ghosh, and Yih Huang

{jwanga, aghosh1, huangyih}@gmu.edu

Vulnerable network applications such as web browsers, email clients, media players, and office productivity software bring significant risks to the operating system and users. To protect the user's computer, currently anti-virus software is the standard defense used to detect malware largely by signature matching. As a result, anti-virus software usually cannot detect new attacks and is largely ineffective at addressing metamorphic code and other current malware threats. Intrusion detection systems face similar problems in being reactive in detecting threats.

Sandboxing is another method to protect the end user's computer. Different levels of sandboxing are feasible, ranging from language software fault isolation, process level system call mediation, to hardware virtualization level. Unfortunately, language and process level sandboxing are susceptible to bypass, and current hardware virtualization sandbox, such as the Tahoma system, separates the applications as well as the data that they usually share, therefore sacrificing usability.

To provide strong isolation between applications while keeping usability, we use hardware virtualization to create virtual machines on-demand and supply a persistent storage to save useful data. We separate network applications or applications that run untrusted code in their own virtual machine and seamlessly integrate it with user's host desktop by using a redirection module. Unlike the Tahoma system, which modifies the source code of the browser applications, our method is compatible with current proprietary commercial applications.

We implemented a prototype called Internet Cleanroom on Microsoft Windows. It uses VMware Workstation as the virtualization layer and includes redirection modules, a control console and dispatcher modules. To enable seamless integration with the standard desktop experience, we intercept Windows API calls from the host operating system that create new processes, and then re-direct the creating process events to a host dispatcher. The dispatcher then redirects network applications to a pre-built virtual machine. If the virtual machine is not started yet, the control console starts it first. The guest dispatcher running inside the virtual machine launches the network application for the user. The control console also brings the virtual machine to the front and show its window if it is minimized. We use VMware Workstation as the virtualization engine.

In each virtual machine, we install a basic Windows XP distribution plus the applications that should be redirected to each virtual machine(VM). We use snapshot function of VMware to start the virtual machine in a pristine state as well as to restore the VM to its pristine state periodically or by user control. On the GUI of control console, an indicator shows the health state of each virtual machine. When the time reaches the configured restore time or if malware detection software in the virtual machine detects some malware, the indicator turns from green to red to warn the user. The guest dispatcher probes the log file of malware detection software to find the alerts.

For persistence of data, we share a folder on the host to each guest as the persistent storage for the virtual machine. The user needs to manually save his data to this folder for persistence beyond the current session. A program running on the host (which is trusted) moves the files in the persistent storage to another host folder after the virtual machine terminates. This prevents the malware in the guest to access all the persistent data while letting the user keep data between sessions. We evaluated the performance of Internet Cleanroom for five popular applications.