Linux Mandatory Access Control Project

Project Background

A recent strong trend in Consumer Electronics is to build devices around a programmable Linux platform, which offers many advantages such as flexibility and access to lots of open source software. From a security point of view on the other hand this also introduces new threats, both from the outside world (viruses and worms) but also from the user/owner itself who might e.g. install a tool to steal keys used to encrypt Pay-TV programs.

Recently, the ISS group at Philips Research has started to look at ways to "harden" the Linux operating system against such attacks. One of the goals in our approach is to achieve strict process isolation. This means that even when an attacker gets root privileges by exploiting a known OS vulnerability on a running machine, (s)he should not be able to steal secret keys, credit card numbers etc.

"Mandatory Access Control" (MAC) seems to be an attractive solution to this problem. With MAC, essentially the standard Linux permissions are extended with further restrictions enforced by the kernel. Many open source projects exist implementing MAC for Linux, such as SELinux (http://www.nsa.gov/selinux/) and Umbrella (http://umbrella.sourceforge.net/). An open problem with this approach is still exactly which further restrictions should be imposed given the threat model sketched above.

Project Description

The project should identify the security sensitive spots in Linux accessible from a root shell which threaten process isolation. MAC should restrict access to such resources. Examples of these resources are /dev/kmem and insmod. The investigation should result in a report. Based on this knowledge a proof-of-concept demo MAC system should be built, time permitting.

Necessary Qualifications

- Student seeking a Batchelor's or Master's degree in Computer Science/Electrical Engineering.
- should be available for at least 3 months
- has a good knowledge of:
  - the Linux OS and Kernel architecture and internals.
  - Linux boot sequence
  - typical system services running
  - device drivers and how they fit together with the kernel
  - organization of the file system, esp. where security sensitive data (or code) resides in the file system.
- Familiarity with security enhancements to Linux such as SE Linux Umbrella or similar form of Mandatory Access Control would be advantageous

For more information contact:

Joop Talstra, WY 71
Philips Research
Prof Holstlaan 4
5656 AA Eindhoven
Netherlands
phone: +31 (40) 27-43896
fax: +31 (40) 27-46622
email: Joop.Talstra@philips.com

Linux Virtual Machine Monitor Project

Project Background

A recent strong trend in Consumer Electronics is to build devices around a programmable Linux platform, which offers many advantages such as flexibility and access to lots of open source software. From a security point of view on the other hand this also introduces new threats, both from the outside world (viruses and worms) but also from the user/owner itself who might e.g. install a tool to steal keys used to encrypt Pay-TV programs.

Recently, the ISS group at Philips Research has started to look at ways to "harden" the Linux operating system against such attacks. One of the goals in our approach is to achieve strict process isolation. This means that even when an attacker gets root privileges by exploiting a known OS vulnerability on a running machine, (s)he should not be able to steal secret keys, credit card numbers etc.

It turns out that current processor hardware (x86/ARM/MIPS) actually goes a long way to enforce strict process isolation, but popular operating systems tend not to fully exploit these facilities, in the sense that when an attacker gets root access, (s)he has full control of the system. This problem is familiar from the world of "Virtual Machine Monitors" (VMMs) where multiple operating systems are hosted simultaneously on the same hardware, and it is necessary to prevent kernel code of one OS to affect the state of the other OS's. VMMs like VMware (commercial) and Xen (open source, see http://www.cl.cam.ac.uk/Research/SRG/netos/xen) get this problem under control precisely by exploiting previously unused hardware features: e.g. on x86 the VMM runs at the highest privilege level (ring 0), the kernels run at ring 1 and the applications run in ring 3 (as usual).

Project Description

The project should identify the security sensitive spots in Linux accessible from a root shell or compromised kernel/driver code, which would threaten process isolation. The VMM should restrict access to such resources. Examples of such resources are the MMU, PCI configuration registers, DMA controllers etc. In particular, it would be interesting to know whether the VMMs like Xen can realize this control task with a reasonable performance overhead also in the case of heavy multimedia processing (video decompression/rendering). The way how device drivers are be integrated into the system is essential here. Although the project will start out on an x86 platform, an important question is how well this translates to typical embedded processors like ARM and MIPS.
The investigation should result in a report. Also part of the project will be to get Xen up-and-running, time permitting.

Necessary Qualifications

- Student seeking a Batchelor's or Master's degree in Computer Science/Electrical Engineering.
- should be available for at least 3 months
- has a good knowledge of:
    - the Linux OS and Kernel architecture and internals.
    - PC hardware architecture: PCI, MMU, x86 privilege levels
    - x86 assembly
    - HW interface handling in Linux
    - device drivers and how they fit together with the kernel
- affinity with ARM/MIPS on these same subjects is a bonus.

For more information contact:

Joop Talstra, WY 71
Philips Research
Prof Holstlaan 4
5656 AA Eindhoven
Netherlands
phone: +31 (40) 27-43896
fax: +31 (40) 27-46622
email: Joop.Talstra@philips.com