Pioneer: Verifying Code Integrity and Enforcing Untampered Code Execution on Legacy Systems

Pioneer is our first-step toward externally-verifiable code execution on legacy computing systems. We define legacy computing systems as those that do not have secure co-processors such as the Trusted Platform Module (TPM) or CPU-based security technologies like Intel's LaGrande Technology, and AMD's Pacifica and Presidio. Using Pioneer, an external verifier can obtain the guarantee that execution of an arbitrary piece of code on a legacy computing system is untampered by any malware that may be present. In particular, the verifier obtains the guarantee that any pre-existing malware does not: modify the code image, invoke an alternate (malicious) code, or modify the execution state of the code during execution. We have implemented Pioneer on the Intel Penitum IV Xeon processor with 64-bit extensions.

Known issues

We are aware of the following two issues with our current Pioneer implementation. If you discover any other issues or attacks, please let us know. We will be happy to list them here and acknowledge your contribution.

- Nov 2, 2005: A malicious OS can execute at a lower privilege level than the Pioneer Verification Function. In Pioneer, we verify if the verification function executes at the highest CPU privilege level, by checking if it is allowed to turn off maskable interrupts. We perform this check by including the interrupt status bit in the checksum. A malicious OS can execute verification function in user space with maskable interrupts turned off. To do this, the OS modifies the saved copy of verification function's interrupt status bit on the kernel stack and calls the verification function using a system call return. The processor will restore the saved copy of the interrupt status flag to the appropriate register as part of processing the system call return. Now, the verification function runs in user space, but maskable interrupts are turned off.

- Nov 18, 2005: Defeating the stack trick on an x86 processor with 64-bit extensions. We keep a part of the checksum on the kernel stack, under the assumption that the checksum will get overwritten by the CPU if an interrupt or an exception occurs during the checksum computation. This assumption is not true on 64-bit x86 architectures that support dedicated stacks, distinct from the kernel stack, for interrupt and exception handlers.

Code and installation instructions

While we have tested our code, we do not make any guarantees. Use the code at your own risk.

Platform requirements:

- Untrusted platform - A PC based on an Intel Pentium IV processor with EM64T (64-bit) extensions, having an Intel 82545GM Gigabit Ethernet Controller, running the Fedora Core 3 Linux distribution, and Linux kernel version 2.6.7.
- Dispatcher - A PC based on an Intel Pentium IV processor, having a 3Com 3c905C ethernet controller, running the Fedora Core 3 Linux distribution, and Linux kernel version 2.6.11.8.

Installation instructions for the untrusted platform:

1. Uncompress the source code and copy resulting directory into /usr/src/linux-2.6.7/drivers/net.

2. cd into /usr/src/linux-2.6.7/drivers/net/e1000.
3. Change the arrays `h_dest` and `h_source` in the `e1000_clean_rx_irq` function in `e1000_main.c` to the MAC addresses of the dispatcher and the untrusted platform respectively.
4. Run the makethis script.
5. Run the copyover script.
6. Run the restart script.

Source Code

Installation instructions for the dispatcher:

1. Uncompress the source code and copy resulting directory into /usr/src/linux-2.6.11.8/drivers/net.
2. cd into /usr/src/linux-2.6.11.8/drivers/net/3c59x
3. Change the array `self_addr` in the `isSelf_ether` function and the array `other_addr` in the `isOther_ether` function in `3c59x.c` to the MAC addresses of the dispatcher and the untrusted platform respectively.
4. Run the makethis script.
5. Run the copyover script.
6. Run the restart script.

Source Code

The dispatcher and the untrusted platform should be on the same ethernet segment. To send a Pioneer challenge from the dispatcher to the untrusted platform, send a ping packet to the untrusted platform from the dispatcher. Both the untrusted platform and the dispatcher will print out the checksum computation times for the correct Pioneer checksum code and the attacker's checksum code.