INSTITUTIONEN FÖR DATAVETENSKAP | LUNDS TEKNISKA HÖGSKOLA | PRESENTATIONSDAG 2016-06-03

EXAMENSARBETE LLVM-Based Fortification for Kernel Drivers
STUDENT Caroline Brandberg
HANDLEDARE Jonas Skeppstedt (LTH), Henrik Theiling (SYSGO AG, Germany)
EXAMINATOR Flavius Gruian (LTH)

Securing Device Drivers

POPULÄRVETENSKAPLIG SAMMANFATTNING **Caroline Brandberg**

Today's operating systems that are used in highly safety and security critical domains struggle with serious vulnerabilities. This requires new technology, especially concerning the highest error prone software of the operating system, in particular its device drivers.

We live today in a society where it is hard to find anything which is not dependent on technology. Today's cars, trains, airplanes and health-care is highly dependent on software where security and safety are of highest concern.

To be able to ensure people's safety and security one needs a good foundation, namely the operating system. The operating system acts as the layer between the applications and the hardware, hence controlling that everything is done in a correct manner.

Therefore, when dealing with highly safety and security critical domains it is very important to have a certified operating system which can ensure the best base to build from. One problem is that the system can be extended with new software by a third party, where the extended software is referred to as device drivers. These extensions will then be a part of the critical layer, and must therefore be of the same quality as the rest of the system. An extension which does not follow the same standard may introduce serious problems which may crash the whole system.

Device drivers have been shown to have an extraordinarily higher error rate compared to the rest of the system. Since they are integrated to such critical parts of the system it requires a new technology that can prevent these vulnerabilities from being introduced into the system.

There are especially two problems concerning pointers. The first concern is securing the drivers against malevolent users. Pointers received from a destination where a malevolent user can occur must be checked properly. A failure to do so might reveal or destroy critical parts of the system which can cause serious problems. The second concern is securing the extensions from performing memory accesses in a incorrect manner. These problems are very hard to find and test for, hence an introduction of these problems is unfortunately very easy.

To be able to prevent these kinds of problems, we propose a tool which the extended software can use to achieve a high level quality. The tool uses a combination of successful techniques, where we show that only a combination of those can ensure protection against these vulnerabilities. We propose both static analysis and dynamic runtime checks which both have been extended to the Real Time Operating System PikeOS. The statically part of the extension was proven to provide a highly safety net for driver developers, providing valuable information during compile time, whereas the dynamic parts provide almost full error coverage and only showed a very low overhead.