Koen Koning’s thesis explores how we can protect our software from malicious attackers, in an efficient manner by leveraging the underlying hardware. Koning completed his research under the supervision of Herbert Bos and Cristiano Giuffrida (both from Vrije Universiteit Amsterdam).

Software, ranging from your web browser and operating system to servers in the cloud, are riddled with bugs. Hackers use such bugs to attack the software and steal sensitive information. One common type of bugs is memory errors, which are present in programs written in unsafe languages like C and C++. Here, the programmer is responsible for the—error prone—task of manually managing computer memory. Microsoft estimates over 70% of its security issues are caused by memory error bugs.

One solution for these issues is to use a safe programming language, such as Python or Rust. However, in practice this is not often not possible because of the high costs. Therefore, researchers have instead looked at automatically protecting existing unsafe programs. During compilation or at run time, additional checks are inserted to ensure proper operation of the software. But despite decades of research, no practical defenses exist. The biggest problem with existing systems is either incompatibility with existing software, or excessive overhead (significant slow-down or high resource usage).

In his thesis, Koning proposes several new techniques to improve automated software defenses, in particular by utilizing the underlying hardware. Modern processors have a myriad of features, for example to efficiently run virtual machines in the cloud. This research takes such features and instead uses them to increase performance and increase security. In particular, Koning demonstrates a number of features that can be used to increase security of existing defenses, perform bounds checking without checks, and to efficiently run multiple copies of software and monitor for divergences in behavior.