Unlocking CPU telemetry for software identification

As modern compute systems grow increasingly complex, so does the attack surface for novel exploits. For instance, side-channel attacks such as Spectre have demonstrated how micro-architectural optimizations can lead to unforeseen vulnerabilities. Many existing methods to harden systems to such known threats focus narrowly on *known* attack vectors. In the spirit of Zero Trust, there is a need for a more general hardware immune system that can react to a changing threat landscape by identifying known behaviour or changes to behaviour in existing systems: in other words, symptomatic behaviours.

Unwanted software can hide in a variety of ways, such as file-less attacks or encrypted polymorphic malware. We introduce a novel use of performance monitoring sensors to profile the behaviour of software to identify it, not by what it is, but by what it does. We ask the question: can processors infer the nature and character of the software running on them? And what are the limits to this distinguishability?

In this talk, we propose and implement a software identification framework called **Akira**. Instead of aggregating telemetry from performance monitoring sensors, we project them, in real-time, using a novel feature extractor which preserves the temporal interactions of raw events. The output of this feature extractor is then classified using a multi-tiered, multi-label ranking system. The proposed system is composed of several elements. The computing system's existing performance monitoring sensors generate events in response to state changes in the system. We efficiently extract temporal features from this telemetry in novel fashion by computing a path signature transform in hardware. These signatures are then forwarded to machine learned algorithms for classification. We evaluate several classifiers for use in identifying software behaviours.

We show that the ranking performance of this classification pipeline outperforms the state of the art in PMU counter-based software identification/malware detection and at a much finer granularity. After training, this detection system is capable of identifying kernel functions. We evaluate its multi-label ranking performance using Label Ranking Average Precision scores and demonstrate identification of kernel functions from event sequences just 25 events long. Then, this ability is expanded with a similar approach to identifying user-space functions.

Finally, we identify a path to extend this ability to identify functions into a full software identification stack using a quasi-control-flow-graph approach.

Presenter Biography:
Brendan Moran is a Principal Security Architect and security researcher at Arm, where he has worked for the last 9 years in OS development, secure software delivery, and security research. His focus is in developing new security solutions for processors in an evolving threat landscape.