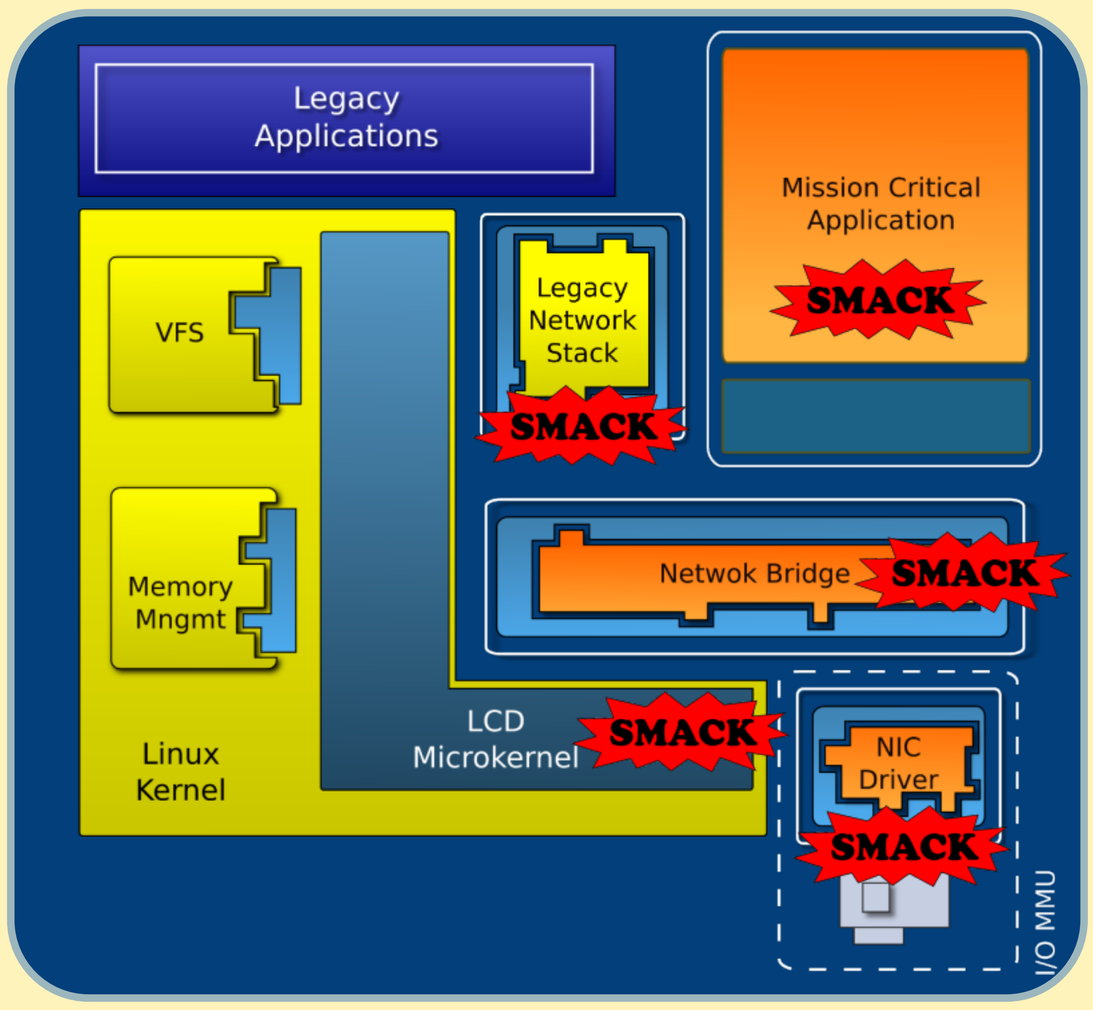
DEKER: DECOMPOSING OS KERNELS

PIs: Zvonimir Rakamaric, Univ. of Utah, Anton Burtsev, UC Irvine
https://www.flux.utah.edu/project/deker

In a modern system, an attacker is **one kernel vulnerability** away from gaining complete control of the entire machine. Can we make our systems more secure?

Conventional OS kernels, despite being notoriously complex, are a part of many mission critical systems. Modern kernels are still developed with legacy software engineering techniques – a combination of an unsafe programming language, and virtually no testing or verification tools. We are creating Deker, a framework for decomposing and verifying commodity operating system kernels. Deker turns a standard commodity operating system kernel into a collection of strongly isolated subsystems suitable for verification. Such subsystems are then verified using our SMACK software verifier. In combination, these techniques improve kernel security.



Approach

- Decompose commodity kernel into strongly isolated modules
- Decomposition environment relies on fast asynchronous cross-core IPC, cooperative thread scheduling, and asynchronous execution runtime.
- Rigorous *interface definition language* (IDL) for specifying protocols that govern decomposed subsystems
- Compiler analysis for automated decomposition
- Light-weight verification of modules in isolation using SMACK software verifier

Mechanisms for enabling decomposition

We worked towards developing a set of *decomposition patterns* – design and development principles aimed at breaking typical patterns of existing monolithic kernel into isolated subsystems with only minimal changes. This includes development of Deker interface definition language (IDL) and custom LLVM static analysis for decomposition.

SMACK software verifier

We extended SMACK software verifier to make it suitable to be used within this project. First, we developed support for checking of important memory safety properties such as buffer overflows. Second, we put in place an automated build infrastructure that allows for SMACK to be run on kernel modules.

Efficient decomposed environment

We performed integration of our fast IPC communication primitives with the Deker domains. We further worked on completing design and development of the asynchronous execution runtime.

Preliminary case study

We have built a first prototype of the in-memory file system (PMFS), and are currently working on decomposing two most performance-critical subsystems in the Linux kernel: a non-volatile memory layer and a 10Gbps network device driver.

Interested in meeting the PIs? Attach post-it note below!

National Science Foundation
WHERE DISCOVERIES BEGIN

The 3rd NSF Secure and Trustworthy Cyberspace Principal Investigator Meeting
January 9-11, 2017
Arlington, Virginia

SCHOOL OF COMPUTING
UNIVERSITY OF UTAH
UCI