Kurzfassung

Unikernel werden erstellt, indem ein Bibliotheks-Betriebssystem zusammen mit einer Anwendung in ein minimales, einem Zweck dienendes Betriebssystemabbild kompiliert wird, das nur einen einzelnen Adressbereich besitzt. Typische Anwendungsfälle sind Cloud-Computing und High-Performance-Computing. Während es viele Publikationen zur Performanz von Unikerneln gibt, gibt es nur wenige Publikationen, welche unabhängig die Sicherheitsbehauptungen analysieren, die von vielen Unikerneln aufgestellt werden.
Diese Arbeit stellt eine Übersicht über Unikernel sowie generelle Sicherheitskonzepte zur Verfügung und diskutiert Sicherheitskonzepte in Bezug auf Unikernel.
Der Hauptbeitrag dieser Arbeit ist zweigeteilt: Der erste Teil ist eine tiefgehende Sicherheitsanalyse des Forschungsunikernels RustyHermit. Es wurde gezeigt, dass RustyHermit einige der grundlegendsten Sicherheitsmechanismen nicht implementiert oder deren Implementierung fehlerhaft ist. Neben anderen Fehlern wurde ein Out-of-bounds Lesezugriff in RustyHermits Netzwerktreiber gefunden, was zeigt, dass auch ein in einer memory-safe Sprache geschriebener Unikernel Sicherheitslücken mit Bezug zu Speicherzugriffsverletzungen aufweisen kann.
Der zweite Teil ist eine Übersicht über den aktuellen Status der Sicherheit von Unikerneln. Die bekannten Projekte OS^v, Unikraft, nanos und Mini-OS wurden auf die Implementierung der grundlegendsten Sicherheitsfunktionen untersucht: ASLR, Stack Canaries, W^X und die Erzeugung von Zufallszahlen. Es wurde gezeigt, dass bei mehreren der Projekte eine funktionierende und sichere Implementierung dieser Sicherheitsmaßnahmen fehlt.
Mehrere Möglichkeiten wurden aufgezeigt, wie Angreifende, die eine verletzbare Anwendung ausnutzen, den gesamten Unikernel übernehemen oder Zugriff auf sensible Informationen erhalten können, da Sicherheitsfunktionen fehlen oder fehlerhaft sind, die in traditionellen Betriebssystemen vorhanden wären.

Stichwörter: Unikernel, Betriebssysteme, Sicherheit

Abstract

Unikernels are created by compiling a library operating system together with an application into a minimal single-purpose single-address-space image. Typical use cases are cloud computing and high performance computing. While there are multiple publications on unikernel performance, there are only few on unikernel security independently discussing the security claims made by multiple unikernels.
This thesis provides an overview of unikernels as well as general security concepts and discusses security concepts related to unikernels.
The main contribution of this thesis is two-fold: The first part is an in-depth security analysis of the RustyHermit research unikernel. It was shown that RustyHermit does not implement some of the most basic security mechanisms or that their implementation is flawed. Besides other bugs, an out-of-bounds read in RustyHermit's network driver was found, showing that a unikernel written in a memory safe language might still contain memory corruption related vulnerabilities.
The second part is an overview of the current state of unikernel security. The prominent projects OS^v, Unikraft, nanos and Mini-OS were analyzed for the implementation of the most basic security features: ASLR, stack canaries, W^X and random number generation. It was shown that many of the projects lack a working and secure implementation of some of these security features.
Multiple examples are given on how an attacker exploiting a vulnerable application can take over the whole unikernel or gain access to sensitive information due to missing or flawed security features, which would be present in a traditional operating system.

Keywords: Unikernel, OS, Operating Systems, Security