Unobtrusive Mechanism Interception

Patrick Lampe, **Markus Sommer**, Artur Sterz, Jonas Höchst, Christian Uhl, Bernd Freisleben

LCN 2022

Problem Statement

- How to deal with unsupported, proprietary systems?
- Lacking features or even security issues
- How can we enable old systems to use new technologies?

Mechanism Interception

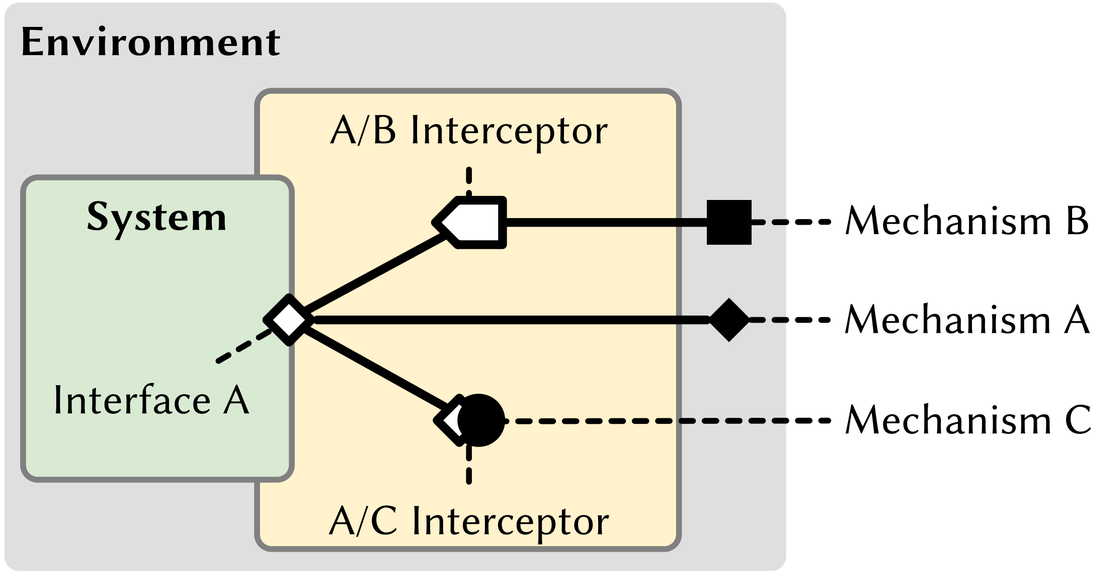
Examples

- Network Address Translation:
  - Intercept IP traffic, mangle packet headers, redirect traffic
  - Existence largely invisible to network participants
- WINE:
  - Intercept system calls of Windows applications
  - “Redirect” to native Linux system calls

Mechanism Interception

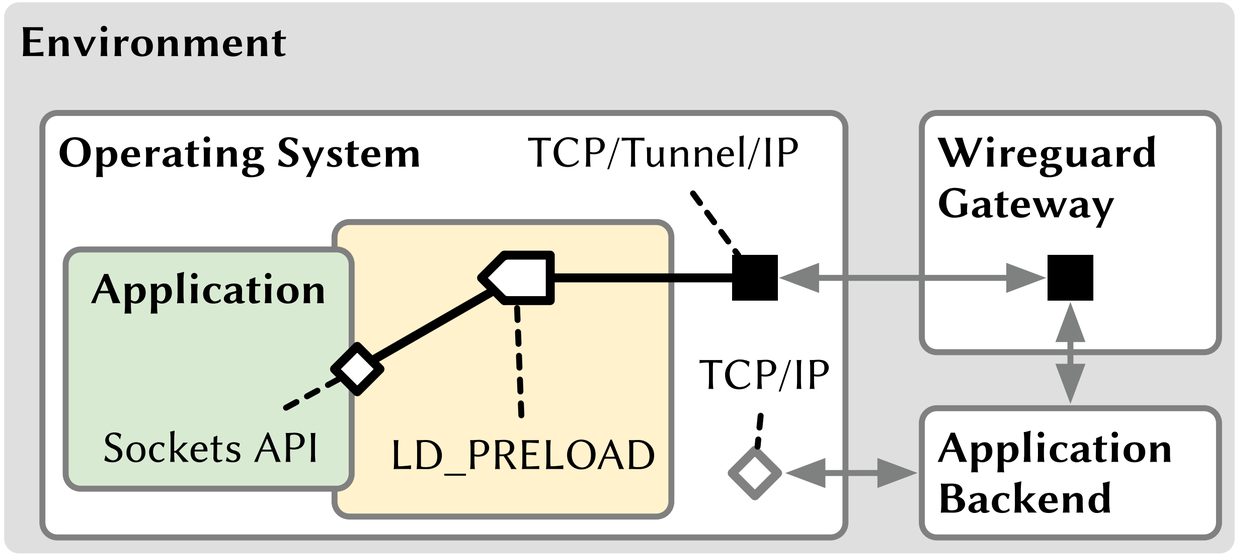
Generalisation

- *System* is proprietary, uncontrollable
- *Environment* can be controlled
- NAT:
  - System is a device in a private network
  - Environment is network
- WINE:
  - System is a Windows application
  - Environment is OS



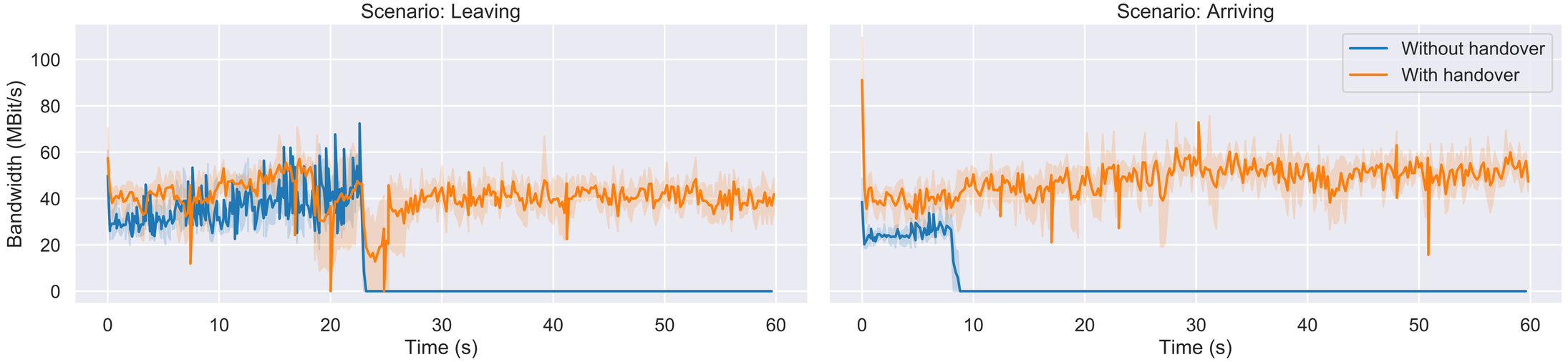
Application
Tunnel Handovers

- During *vertical handover,* transport-layer connection is interrupted
- Connection must be re-established on both transport- and application-layer
- `LD_PRELOAD` allows interception of function calls
- Traffic is sent over WireGuard tunnel to gateway
- Wireguard connection resilient to interruptions



Evaluation

- Network emulation via CORE, test application iperf3
- Simulate handover from WiFi to LTE and vice-versa
- Without interception: TCP connection breaks, throughput falls to 0
- With interception: throughput recovers after a short dip



Thank you for your time!

Patrick Lampe	lampep@informatik.uni-marburg.de
Markus Sommer	msommer@informatik.uni-marburg.de
Artur Sterz	sterz@informatik.uni-marburg.de
Jonas Höchst	hoechst@informatik.uni-marburg.de