Containment-based Security Architecture

Team Members: Hansen Zhang and Sotiris Apostolakis
Advisor: David I. August
Current Approach to Security is ineffective

Software

- 'Stunning' Increase in Data Breaches in 2017
- Hacked Dropbox login data of 68 million users is now for sale on the dark Web
- Bigger than Heartbleed, 'Venom' security vulnerability threatens most datacenters

Hardware

- Rowhammer hardware bug threatens to smash notebook security
- How Soviets used IBM Selectric keyloggers to spy on US diplomats
- COUNTERFEIT CHIPS PLAGUE U.S. MISSILE DEFENSE
- Researchers built devious, undetectable hardware-level backdoor in computer chips
Current Approach to Security is ineffective

“Clean Slate” Approach to Security is impractical

System

Applications
Libraries
Operating System
Hypervisor

Main Memory
Processor
External Interface

System Bus

>1 Billion LOCs

Trusted & Untrustworthy
Trusted & Trustworthy

seL4 9.1k LOC\textsuperscript{1}

Linux kernel \textasciitilde19.1m LOC\textsuperscript{2}

\textsuperscript{1} Gerwin Kleine, et al. 2009. seL4: formal verification of an OS kernel. SOSP '09. \textsuperscript{2}https://www.wired.com/2015/02/nsa-firmware-hacking/

\textsuperscript{2} Linux Kernel, http://www.phoronix.com/scan.php?page=news_item&px=MTg3OTE
Containment-based Security is both effective and practical.
A few thousand lines of code can ensure correct output of an entire database server despite flaws and vulnerabilities (known and not-yet-known)
Prototype

Only connection to network
Network Interface
PCIe Interface
FPGA Sentry
Verifying Computation & Memory Integrity in the Sentry

Insight 1: Leverage work done by the untrusted system
Insight 2: Offload extra work to the untrusted system
Plan: Beyond a Proof of Concept

### Software Stack for Applications

<table>
<thead>
<tr>
<th>Programming Language</th>
<th>Libraries</th>
<th>Debugging Tools</th>
<th>Runtime System</th>
<th>Compiler</th>
<th>Operating System</th>
</tr>
</thead>
</table>

### Software Stack for Containment

| Containment Programming Model | Composable DSC/Policy Libraries | Sentry Emulator & DSC/Policy Debugger | Sentry Runtime | Sentry Compiler | Sentry Libraries & OS Wrappers |
Plan: Beyond a Single Device
Team Strengths

Hardware Lead

Hansen Zhang
4th year PhD student

Computer Architecture & Security
Built Sentry Prototype

Software Lead

Sotiris Apostolakakis
3rd year PhD student

Compilers & Security
Built Redis DSC, Software Support

Built successful containment-based security architecture proof of concept!

The Liberty Research Group at Princeton University has made significant contributions to computer architecture, security, and language tools for over 15 years.
Questions?