Improving Hypervisor Security

Team: Daniele Buono, Carlo Bertolli, Tobin Feldman-Fitzthum, Hubertus Franke, James Bottomley
IBM Research, T.J. Watson Research Center
The Hypervisor is a very powerful piece of software.

- The Hypervisor is managing the (software) wall
- Hypervisor in Linux is split
  - KVM, with kernel-level privileges
  - VMM, with user-space privileges
- QEMU is the most complex VMM
  - Many of the features are actually not needed
  - Some may not be tested/maintained properly
- More features -> more code -> larger surface of attack -> Less security
Measuring surface of attack via lines of code

- Since most of the features of QEMU are not needed for the specific case of Cloud Virtualization, why not remove them?
- QEMU is not modular
Measuring surface of attack via lines of code

- Since most of the features of QEMU are not needed for the specific case of Cloud Virtualization, why not remove them?

QEMU is not modular

- QEMU has done a lot of progress in terms of modularity and code reduction capabilities in the past two years; however

- Such changes have yet to reach most Linux distributions
  - Slow production cycles
  - Loss of generality
Measuring surface of attack via lines of code

- Since most of the features of QEMU are not needed for the specific case of Cloud Virtualization, why not remove them?

QEMU is not modular

- QEMU has done a lot of progress in terms of modularity and code reduction capabilities in the past two years; however
- Such changes have yet to reach most Linux distributions
  - Slow production cycles
  - Loss of generality

- Collaborating with QEMU maintainers, we were able to explore the current modularity of QEMU
- The picture is very promising, with QEMU already a serious contender to Cloud Hypervisor
Improving Metrics in Security

Binary Size

Defect Density
(https://scan.coverity.com/projects/qemu)

Converging to
[2.5-4.5] MB
Qemu, Cloud Hypervisor

0.01 over ~2M LOCs
average Open Source project is 0.65

Coverage under common Cloud Workloads

Qemu Control Flow

CVE frequency in popular and unpopular/unused paths

Bug Location in Code

Popular

Unpopular or removed
• Location of fixed bugs through git commit log:
  - CVE-2019-12068
    • hw/scsi/lsi53c895a.c
  - CVE-2018-7550
    • hw/i386/multiboot.c
  - CVE-2018-5683
    • hw/display/vga.c
  - CVE-2018-19489
    • hw/9pfs/9p.c
  - CVE-2018-19364
    • hw/9pfs/9p.c
  - CVE-2018-17963
    • net/net.c.net/net.h
  - CVE-2018-16872
    • hw/usb/dev-mtp.c
  - CVE-2018-16867
    • hw/usb/dev-mtp.c
  - CVE-2018-16847
    • hw/block/nvme.c
  - CVE-2017-9503
    • hw/scsi/megasas.c
  - CVE-2017-8379
    • ui/input.c

• Next Step:
  - Work on Coverage Reports
    • Isolate popular and unpopular paths
    • Isolate unreachable paths, such as:
      • devices not enabled at runtime
      • interfaces not accessible from the guest (e.g. monitor)
    • Correlate past CVEs to the popularity of the paths
Improving Hypervisor Security

- Current focus only on VMM -> QEMU
  - Code reduction
    - Latest QEMU versions have multiple features for specialization
  - Work on metrics to measure security
    - Coverage, code path classification
  - Bug Localization
    - How many of the bugs found in the past 4 years would have been avoided with specialization?
    - Are bugs more frequent in most used or least used parts of the code?