

The common challenges of secure VMs

—
Janosch Frank

frankja@de.ibm.com

frankja@linux.ibm.com

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time

this information was published. Such trademarks may also be registered or common law trademarks in other countries.

A current list of IBM trademarks is available on the Web at “Copyright and trademark information” at www.ibm.com/legal/copytrade.shtml

The following are trademarks or registered trademarks of other companies.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

AMD, the AMD Arrow logo, and combinations thereof are trademarks of Advanced Micro Devices, Inc.

Intel, the Intel logo, and other Intel marks are trademarks of Intel Corporation or its subsidiaries.

Other company, product, or service names may be trademarks or service marks of others.

Whoami

- Janosch Frank
- KVM & KVM-unit-tests s390 co-maintainer
- Implemented & tested large parts of the secure VM support on s390
- Unit test enthusiast


Disclaimer

- There are only so many pages of architecture I can read and comprehend
- For the sake of readability a lot of detail was left out

Outline

- Recap: Secure VMs
 - What, Why & Who
- Challenges:
 - Runtime protection
 - Boot
 - IO
- Current collaborative development
- Future
- Summary

Key

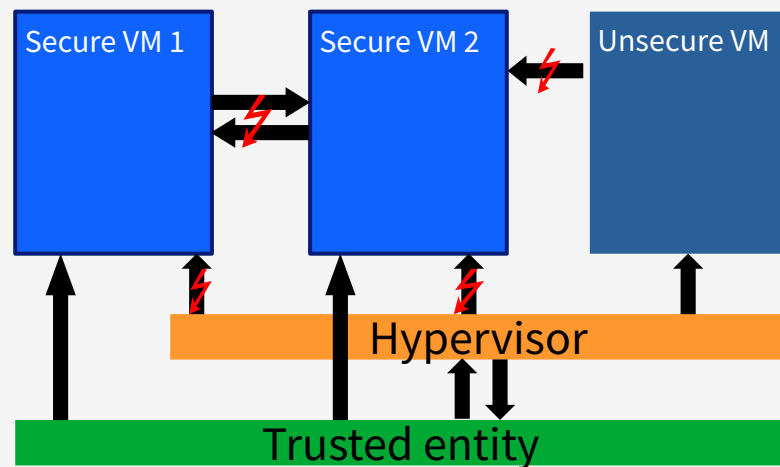
- In the following slides you'll see batches like these:  IBM AMD Intel
- They indicate which vendor chose to implement the technique / feature we're currently discussing.
- For AMD® there might be a „ES“ or „SNP“ in there which means that it was introduced in one of the extensions.

Recap

Recap

What & Why

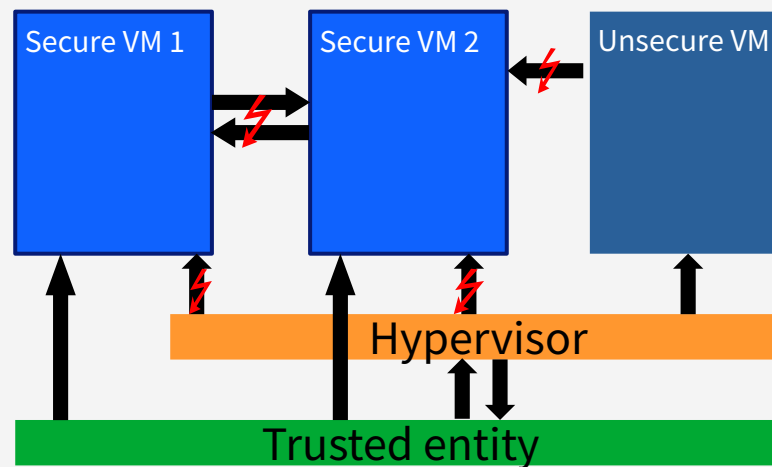
- VMs whose sensitive state is not accessible from the OS / hypervisor
- Instead a trusted entity manages sensitive VM data
- Hypervisor cooperates with the trusted entity to run secure VM



Recap

What & Why

- Protects against host to VM attacks
- Protects against VM to VM attacks
- Allows users to confidently deploy sensitive workloads into a public cloud



Recap

What

- **Sensitive state:**
 - (Initial) memory contents
 - Registers
 - VM controls (emulation controls, interrupt injection)
- **We want to combat:**
 - Data leakage & manipulation
 - Manipulation of execution flow

Recap

How

- Basic building blocks:
 - Encryption / hiding
 - Access control
 - Integrity verification

Recap

Who

- Who is in on it?

Recap

Who

- AMD® SEV (Secure Encrypted VMs, since 4.15)
 - Extensions ES and SNP
- IBM® SE (Secure Execution, since 5.4/5.7 on P/Z)
- Intel® TDX (Trust Domain Extensions, work ongoing)
- Surely more to come in the future

Recap Summary

- Secure VMs are protected against VMM's and other VM's accesses and managed by a trusted entity
- Most major architectures have secure VM technology
- Three basic building blocks provide security
- A lot of challenges are shared between architectures

Challenge: Runtime Protection

Challenge: Runtime protection

Overview

- **Memory protection**
 - Protect secure VM memory from VMM and other VMs
- **State protection**
 - Hiding VM CPU and interrupt state from VMM

Challenge: Runtime protection Memory

- **Challenge:**
 - Memory needs to be unreadable for VMM
 - Even better if the memory is also unwriteable
 - Best if also integrity protected

- Solution 1: Memory encryption
- Solution 2: Memory access protection
- Solution 3: Page table and swap protection

Challenge: Runtime protection Memory - Encryption

- VM's memory is encrypted by memory controller
- Each VM has its own key
- Key is kept in hardware
- Read and write with the wrong key will result in random data

Challenge: Runtime protection Memory - Encryption

- **Advantage:**
 - Memory is always accessible
 - Cold boot attack protection
- **Drawbacks:**
 - Limited amount of keys
 - No integrity protection
 - VMM controls host to guest
pageable

Challenge: Runtime protection Memory – Access protection



- Reads/writes from outside of a secure VM will result in an exception

- **Advantage:**
 - Integrity protection
 - Access tracing

- **Drawbacks:**
 - IO r/w also leads to exceptions
 - Owner tracking of each physical page necessary
 - No cold boot attack protection

Challenge: Runtime protection Memory - Integrity



- **We need to protect page integrity when:**
 - The VMM can manipulate (write to) a secure VM page
 - The VMM can manipulate the VM's memory mappings
 - Memory can be swapped
- Trusted entity does or safeguards swap and page table management
- VMM can ask the trusted entity for changes

Challenge: Runtime protection Registers



- **Challenge:** Registers need to be unreadable and (optionally) integrity checked
- Registers could contain encryption keys or other sensitive data.
- The trusted entity selectively hides / encrypts registers unneeded for instruction emulation

Challenge: Runtime protection Interrupts



- **Challenge:** Injecting interrupts can lead to unexpected instruction flow changes
- Trusted entity manages and / or guards interrupt injection

Challenge: Runtime protection Interrupts



- **Challenge:** Injecting interrupts can lead to unexpected instruction flow changes
 - Firmware manages and / or guards interrupt injection
 - Can drive developers mad (trust me)

Challenge: Boot

Challenge: Boot Summary

- **Challenge: Only boot customer approved executables**
- Solution 1: Remote attestation
- Solution 2: Encrypted boot data

Challenge: Boot Attestation

- A trusted entity authenticates its hardware and the VM's software to a remote host

- **Advantage:**
 - Flexible machine authorization

- **Drawbacks:**
 - Complex to implement
 - Needs connection to outside to boot VM

Challenge: Boot

Boot data encryption

- Executable is fully encrypted (IBM Z) or only measured and boot seed is encrypted (IBM Power)
- Non-secure start followed by move into „secure“ mode

- **Advantage:**
 - No remote machine needed to be able to boot
 - Executable can be loaded by untrusted non-secure code (BIOS)
 - VMM doesn't know anything about VM executable when fully encrypted
- **Drawbacks:**
 - Updating kernels and revocation is complicated
 - Bootloaders
 - Encryption public keys need to be distributed

Challenge: Boot Tooling

- All of the solutions need extensive tooling:
 - AMD[®]: sev-tool
 - IBM[®] Z: genprotimg
 - IBM[®] Power: „conversion/preparation tool“
 - INTEL[®]: ?

Challenge: MM

Challenge: IO



- **Challenge:** IO data needs to be r/w by host and by the guest
- Solution: Remove encryption / protection for some pages
- Currently done by bounce buffering IO data to those shared pages
- Special handling for IO also means less performance

Challenge: Swap



- **Challenge:** Get swap to work
- Solution: Memory is made available encrypted for the host to swap
- Optimally integrity and replay checked on swap-in

Development

Development

- QEMU host-trust-limitation patches
- Kernel VIRTIO IOMMU hooks
- Kernel MM changes

- Currently focused on AMD SEV encryption
- Hooks also used on IBM Power and Z

Development: CGROUP FW limits

- The trusted entity can only manage a certain number of secure VMs
- We need to be able to limit secure VM creation

- **AMD: ASIDS**
 - SEV ASIDs (15 for EPIC Gen 1, 509 on Gen 2)
 - SEV-ES ASIDs
- **Intel: Key Ids**
 - Private keys
 - Shared keys
- **IBM:**
 - Z has lots of VM IDs but sees value in limiting nevertheless

Development: Test

- How can we integrate the tests into our test frameworks?
- Platform dependent solutions:
 - With boot data encryption we can easily convert tests
 - With attestation we need more setup to run tests

Development: Test

- **A lot of new code & firmware to test**
 - Firmware API
 - KVM & firmware cooperative emulation paths
 - KVM IOCTLs
 - IO changes, paging
 - QEMU, Libvirt and integration into VM management
 - Upcoming features
- **A lot of new or changed execution paths**

Future

Challenge: Migration

- Migration is only allowed to certain hosts
- All data needs to be transferred encrypted and integrity checked to destination
- CPU state needs to be exported from the trusted entity
- Backwards compatibility needs to be managed by trusted entity

Challenge: Migration

- Migration policies will decide migration eligibility of destination host
- New APIs for VMM to cooperate with Firmware for migration
- AMD® presented their solution last year
- It certainly will be a lot of work

Future: Secure IO devices

- I expect to see „secure“ IO devices in the future
 - Devices might be authenticated and / or measured
 - Devices only respond to the VM's IO requests and not to VMM
- We'll lose flexibility for IO device attachment
- Managing such devices will be a challenge

Future: Dump

- Without VM introspection, debugging is complicated
- KDUMP can write dumps to an encrypted disk
- What happens if we can't boot into KDUMP?

Future: Dump

- Dumping will need cooperation from trusted entity
 - Exporting encrypted guest registers and memory to VMM
 - New KVM IOCTLS needed
- Maybe we can add one interface instead of per architecture ones?

Future: additional protection

- SMT disabling
- Debug & performance counter disabling
- Cache flushing against side-channel attacks

Summary

- Basic building blocks of secure VMs are similar across architectures
 - Difference is in the specific implementation
 - Secure VM technology is getting more important and complicated
- Now is the time to collaborate

Thank you

Janosch Frank

—

frankja@de.ibm.com

frankja@linux.ibm.com

© Copyright IBM Corporation 2020. All rights reserved. The information contained in these materials is provided for informational purposes only, and is provided AS IS without warranty of any kind, express or implied. Any statement of direction represents IBM's current intent, is subject to change or withdrawal, and represent only goals and objectives. IBM, the IBM logo, and ibm.com are trademarks of IBM Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available at [Copyright and trademark information](#).

