IBM Linux Technology Center

IBM

Secure Virtual Machines on Power

Ram Pai (linuxram@us.ibm.com, pair@us.ibm.com)
- Ultravisor Lead, LTC, IBM
Guerney Hunt (gdhh@us.ibm.com)
- IBM Research

KVM Forum 2018, Edinburgh UK.
October 25th 2018

Agenda.

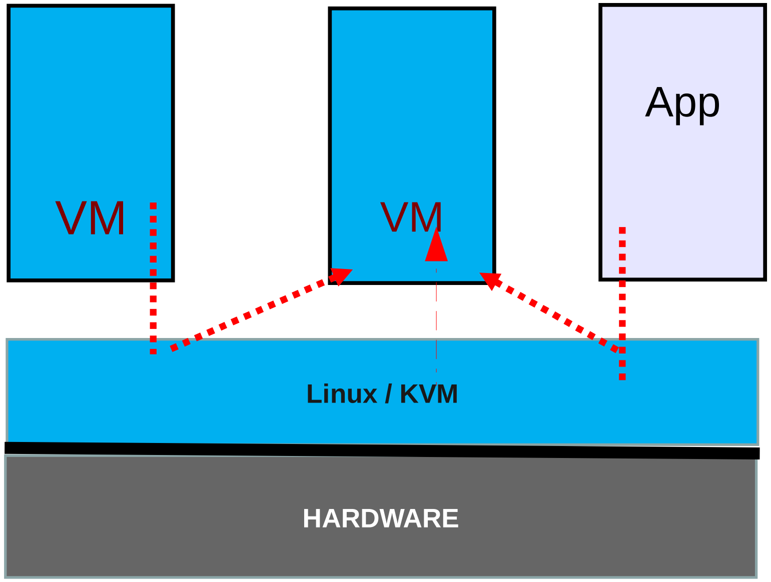
- Problem Statement
- Protected execution facility
- Secure Virtual Machines
    - Image
    - Runtime
- Ultravisor
- Hypervisor

What is the problem?

Security is a major obstacle for cloud adoption, especially in security sensitive sectors such as healthcare, banking, government …

VM can be attacked by

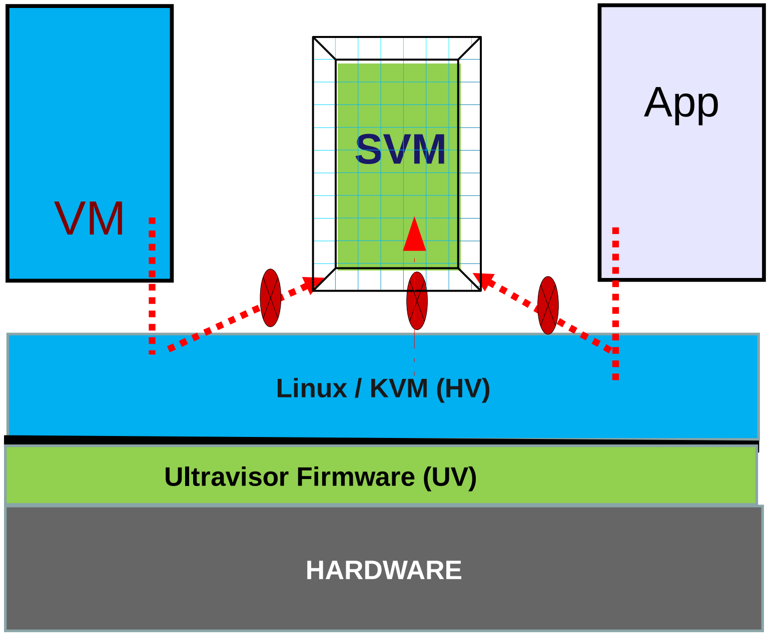
- Rogue/vulnerable hypervisor
- Rogue/vulnerable “other” VMs launching privilege escalation attacks.
- Applications launching privilege escalation attacks.
- Malicious, curious or careless cloud administrator.



Solution: Secure Virtual Machines (SVM)

SVM

- Virtual machines backed by secure memory.
- Hardware and *Protected Execution Ultravisor* firmware (Ultravisor) prevents Hypervisor from accessing secure memory.
- No entity can access the contents of SVM except the SVM and the Ultravisor.

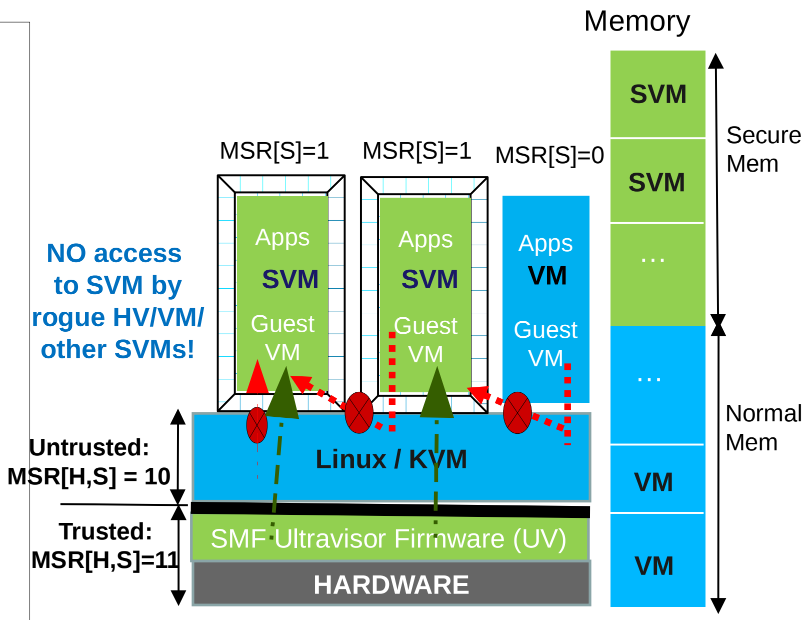


Ultravisor firmware

- Light weight firmware responsible for protecting SVM.

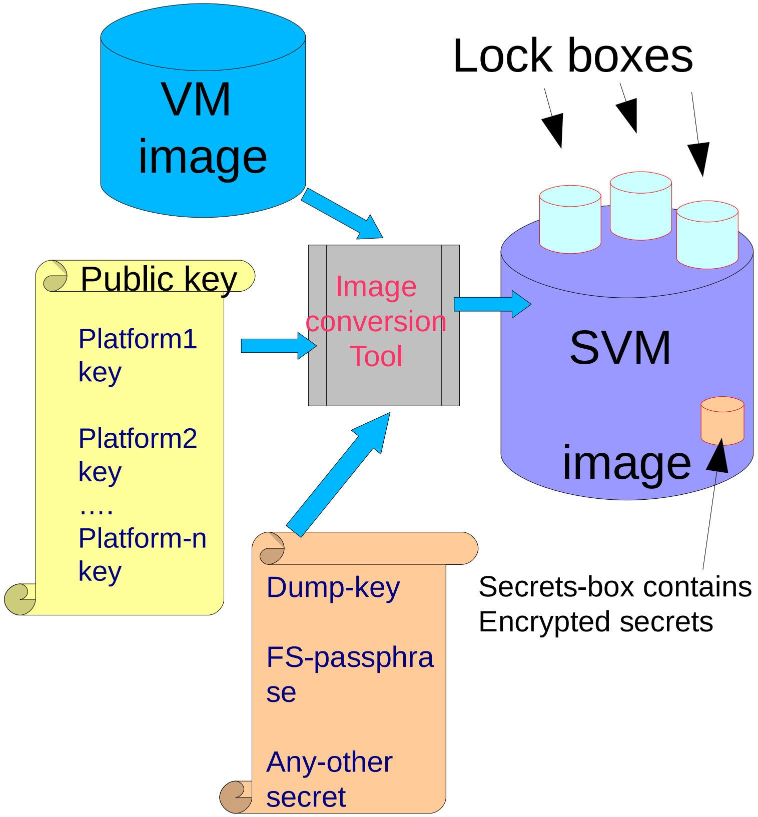
Protected Execution Facility(PEF) on Power9

- Secure Memory
  - Entirely different range of physical addresses.
  - Accessible only if CPU in secure mode.
- Secure CPU mode : MSR(S) = 0b1
  - Can access secure memory.
- Ultravisor CPU mode : MSR(H, S)=0b11
  - Highest privileged CPU mode.
  - Access to all resources.
- Hypervisor CPU mode : MSR(H, S)=0b10
  - Loses access to many key resources including secure memory.
  - Can access the resources through Ultracalls.
- Ultracalls
  - Access to Ultravisor services.



Introduction to SVM

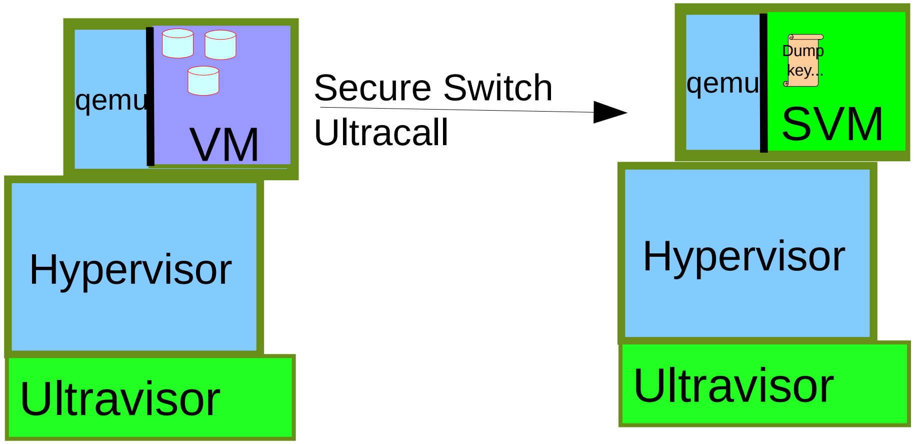
- SVM Image
  - **SVM image = Normal VM image +**
    lock boxes +
    encrypted secrets
  - All secrets in the image encrypted.
  - The encryption key put in the lock box.
  - One lock box per authorized platform, locked using platform's public key
  - A tool to convert a normal VM image to Secure VM image.



Introduction to SVM (cont..)

- **SVM Runtime**
  - All SVM images start as Normal VM, backed with normal pages.
  - VM invokes a ultracall to switch to secure mode (SVM).
  - On successful transition, UV transitions all SVM pages into secure memory.

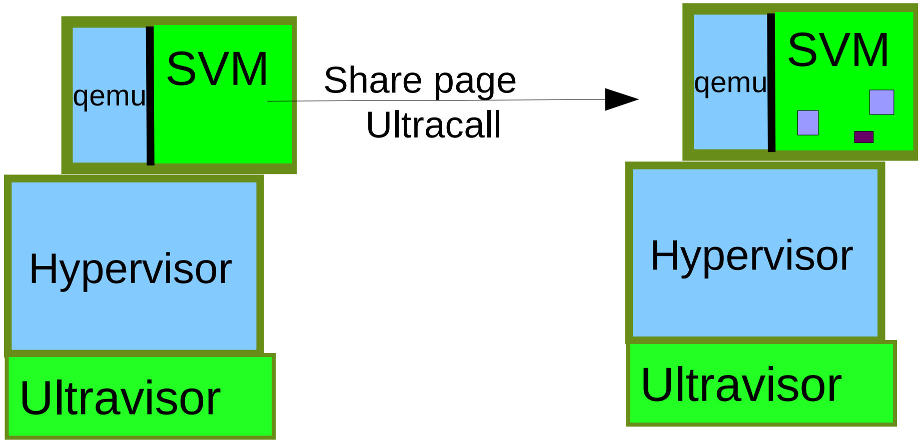
🟩 Secure memory
🟪 Normal memory



Introduction to SVM (cont...)

- **SVM Runtime (cont..)**
  - Explicitly request UV to share address ranges with the Hypervisor. (Shared pages).
  - Needed for
    - VPA(Virtual Processor Area)
      - *https://lists.ozlabs.org/pipermail/linuxppc-dev/2018-August/177334.html*
    - Virtual I/O
      - *https://lkml.org/lkml/2018/7/20/30*

🟩 Secure memory
🟪 Normal memory

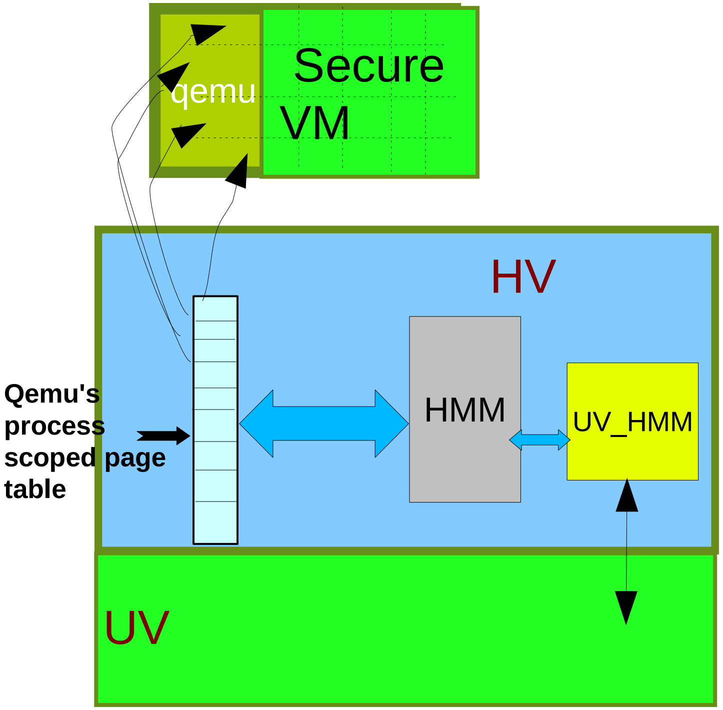


Ultravisor

- **Firmware Code, Opensource GPL**
- **Loads and executes in secure memory.**
- **Responsibility**
  - Authorize/validate VM before transitioning it to Secure mode.
  - Manage secure memory.
  - Back SVM with secure pages.
  - Handle Ucalls from Hypervisor and from SVM/VM
  - Provide services to SVM
    - Marshall and reflect select Hcalls and Exceptions to Hypervisor.
    - Handle other hcalls and exceptions.
  - Offload non-security related services to Hypervisor.
    - Scheduling
    - I/O

Hypervisor

- **Aware of secure pages**
  - Secure pages are mapped into qemu's address space.
- **Treats secure memory as heterogeneous memory.**
- **UV_HMM module orchestrates secure-data movement**
  - From normal memory to secure memory and vice-versa
  - HV or UV can initiate the movement.
  - But UV always moves the content.
    - Encrypted when moved to HV.
    - Decrypted when moved from HV.
  - *https://www.mail-archive.com/linuxppc-dev@lists.ozlabs.org/msg140597.html*



Disclaimer

This work represents the view of the authors and does not necessarily represent the view of IBM.

All design points disclosed herein are subject to finalization and upstream acceptance

The features described may not ultimately exist or take the described form in a product

IBM is a registered trademark of International Business Machines Corporation in the United States and/or other countries.

Linux is a registered trademark of Linus Torvalds.

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

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

Backup

Ultracalls (an incomplete list. Under development)

Ultracalls made by Virtual Machines:

- UV_ESM : *Execute in Secure Mode.*

- *UV_SHARE_PAGE: Share the page at the provide address with the Hypervisor.*

- *UV_UNSHARE_PAGE: Unshare the page at the specified address.*

- *UV_UNSHARE_ALL: Unshare all shared pages.*

Ultracalls (an incomplete list. Under development)

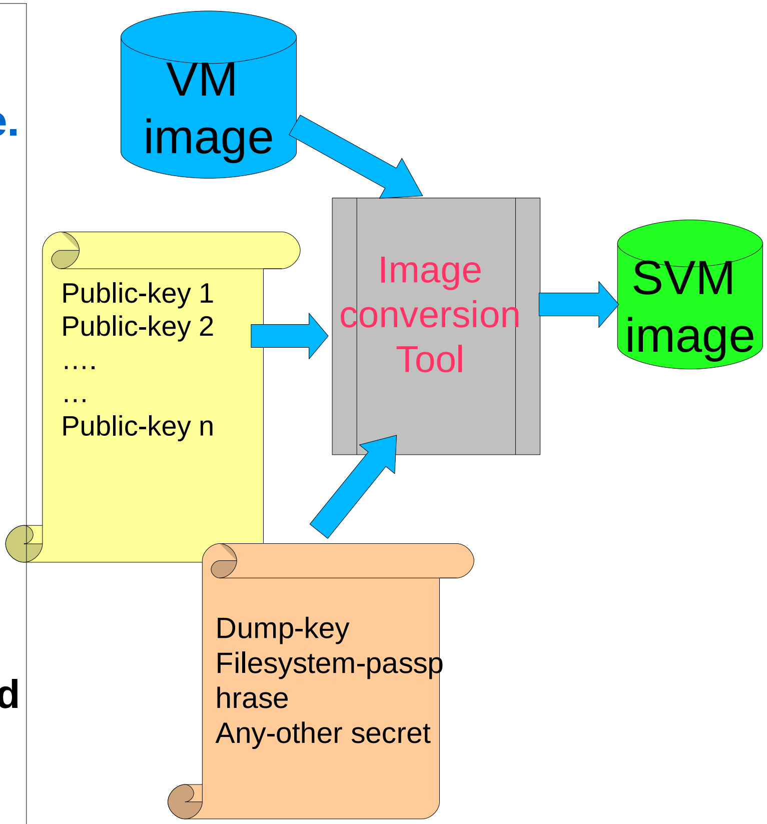
Ultracalls made by Hypervisor:

- UV_PAGE_OUT : Move the contents of a secure page into normal page.
- *UV_PAGE_IN:* Move the contents of a normal page into secure page.
  Or, use the normal page for sharing.
- *UV_PAGE_INVAL: Invalidate a shared page.*
- *UV_REGISTER_MEM_SLOT: register a memory slot for a given SVM.*
-
- *UV_REGISTER_MEM_SLOT: unregister a memory slot of a given SVM.*
- …..

Steps to deploy a secure virtual machine

1. **Get the public keys of all the Power platforms you trust to deploy your VM image.**

2. **Convert your VM image into Secure VM image using a new open source tool.**
   - **This step must be done in your private setup.**
   - **Feed all the public keys to the tool.**
   - **Feed any other secrets that you choose to store in the image.**
     - **Crash dump key**
     - **File-system encryption pass-phrase**
     - **Etc.**
   - **The secrets in the VM image gets encrypted with a dynamically created symmetric key.**
   - **The tool also outputs the symmetric key. Save the symmetric key securely.**

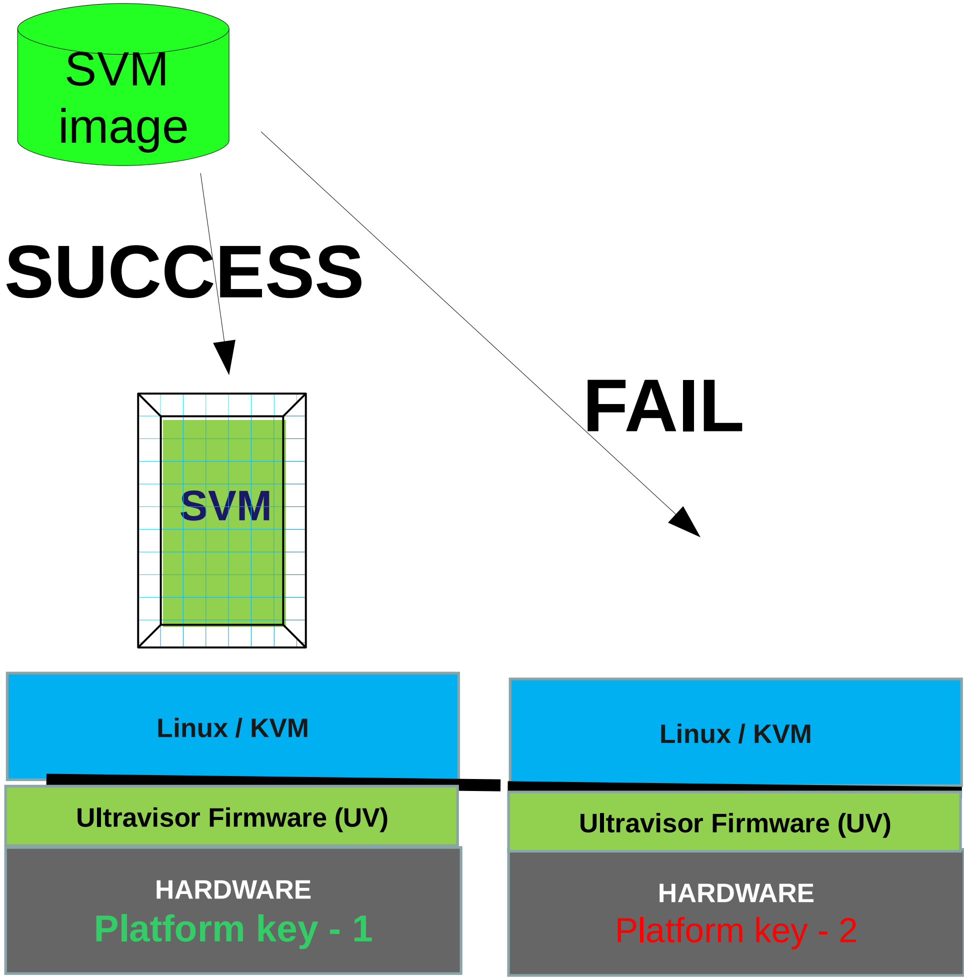


Steps to deploy a secure virtual machine cont..

3. Upload the Secure VM image to your Cloud Service Provider.

4. Deploy the Secure VM image on the POWER platform in the cloud.

- **The ultravisor will only be able to read and deploy the SVM image if the image was created using the machine's public key.**
- **Otherwise it will fail.**



Steps for switching a VM to a SVM (UV_ESM ucall)

Allocate secure pages to the VM

Move the contents of VM's normal page into the secure page.

Locate the lock-box

Procure the symmetric key from the lock box with the help of TPM.

Using the symmetric key, unlock the contents of the secrets-box.

Match the kernel-hash, initrd-ram hash, kernel command line parameters hash against the hashes located in the secrets-box.

If match fail, return failure.

Commit all the secure pages to the VM's page table.

Enable the secure-page access capability for the VM.

Return Success.