SDC

STORAGE DEVELOPER CONFERENCE

SNIA SANTA CLARA, 2017

Persistent Memory Security
SDC 2017

Mark Carlson, Toshiba

Contents

- Persistent Memory Technology
- Persistent Memory Overview
- NVM Programming Model
- NVDIMM
- PM Security
- Multi-Tenant PM Security
- Threat Model

SDC

STORAGE DEVELOPER CONFERENCE

SNIA SANTA CLARA, 2017

Persistent Memory

Persistent Memory (PM) Technology

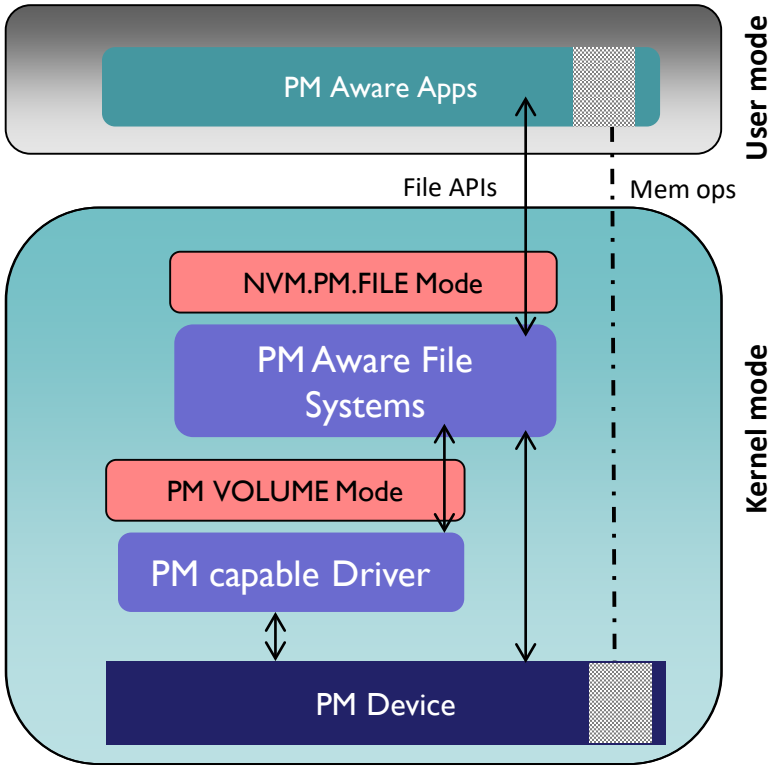
is a type of Non-Volatile Memory (NVM)

- Disk-like non-volatile memory
  - Persistent RAM disk
  - Appears as disk drives to applications
  - Accessed as traditional array of blocks
- Memory-like non-volatile memory (PM)
  - Appears as memory to applications
  - Applications store data directly in byte-addressable memory
  - No IO or even DMA is required
- This talk will focus on PM with Memory Access

Programming Model

Persistent Memory (PM) Modes

- NVM.PM.VOLUME Mode
  - Software abstraction for persistent memory hardware
  - Address ranges
  - Thin provisioning management
- NVM.PM.FILE Mode
  - Application behavior for accessing PM
  - Mapping PM files to application address space
  - Syncing PM files



SDC

STORAGE DEVELOPER CONFERENCE

SNIA SANTA CLARA, 2017

Persistent Memory Security

Purpose of SNIA PM Security work

- This work documents approaches for encryption of data on persistent memory (PM); particularly considering unique characteristics of PM.
  - Discover gaps in existing technologies related to PM security
  - Create a treat model and suggest requirements that could resolve these gaps
- The NVM Programming TWG has established an alliance with the Trusted Computing Group (TCG) outlining a collaboration between the SNIA NVMP TWG, TCG. The collaboration is structured as follows.
  - SNIA provides application/user level roles, behaviors and threat models
  - TCG provides security protocol definitions
- TCG, SNIA also approaching JEDEC
  - JEDEC provides NVDIMM specific specifications

PM Security

- Many aspects of security are unchanged by PM
  - Administrative security
  - Key management
  - Memory protection
- First order requirement: encryption of data at rest
  - Authentication/Re-authentication Triggers
  - Real time encryption mechanics
  - Continuity of principal identity

PM Security

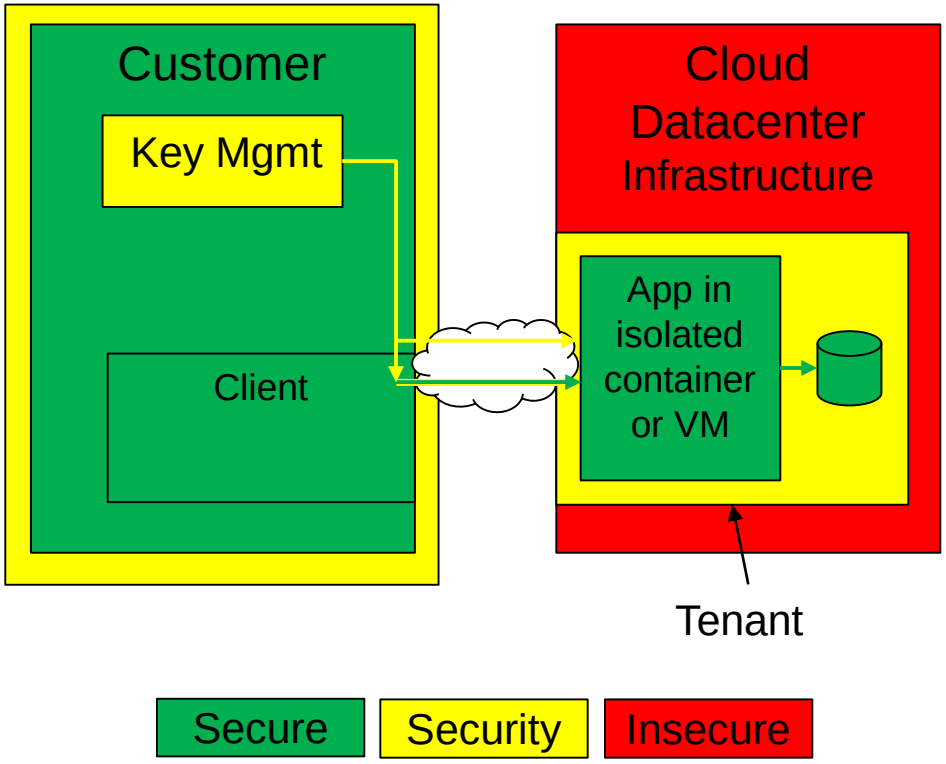
- Protection granularity at the file and volume layers
  - Device, partition or volume protection of data at rest
  - Memory mapped file access authorization enforcement
- Achieving isolation analogous to external storage
  - Limiting access enablement windows
  - Rapid privilege transition

Public and private cloud requirements

- Public speaks to how trust is established and isolation is assured in shared public cloud infrastructure
- Private speaks to multi-tenancy HW support
- Both – encryption at rest, issues from prior 2 slides

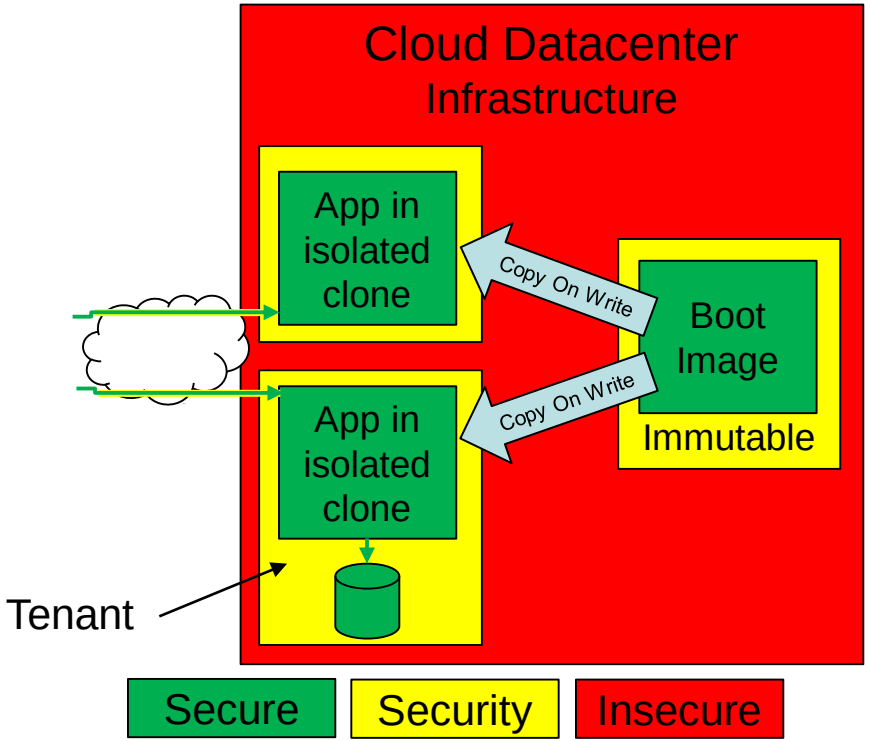
Roles in a Multi-Tenant Cloud Datacenter

- Cloud datacenters are not necessarily trusted by customers where their applications and data are tenants.
- Customer establishes an account with the cloud datacenter
- Customer becomes a Tenant by running an application in an isolated VM or container.
- Application securely mounts storage (HDD, SSD or PM) that is isolated from other tenants
- Customer manages and uses keys to insure trusted execution and storage access.



PM Clone Use Case

- PM boot image is trusted gold standard
- PM boot image is immutable
- Tenants run in clones of boot image
- Writes exist only in isolated clones
- Additional security such as digital signature and virus protection may be required
- Immutability is ensured by cloud provider and enforced by features of the OS and memory controller
- Storage (HDD, SSD, PM) access is authorized based on customer provided keys
  - Mounted after image creation
  - Becomes part of the tenant environment

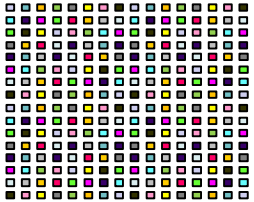


Multi-Tenant Infrastructure Features

- Isolated execution environment for customer applications
  - Customer provides key to enable execution
  - Many per customer, many customers: 10^2 – 10^6 instances
- Secure access to cloud storage
  - Customer provides key to access Files, Objects
  - Similar or larger number of instances
- “Per-Tenant” storage volume/partition
  - Enables secure erase of deleted data
  - 10^1 keys per drive
- Both Persistent and Non-Persistent (temporary for guest) storage usage
- Storage partitions do not attain cloud scale

Multi-Tenant Storage Allocation

Tenants who rely on provider data security

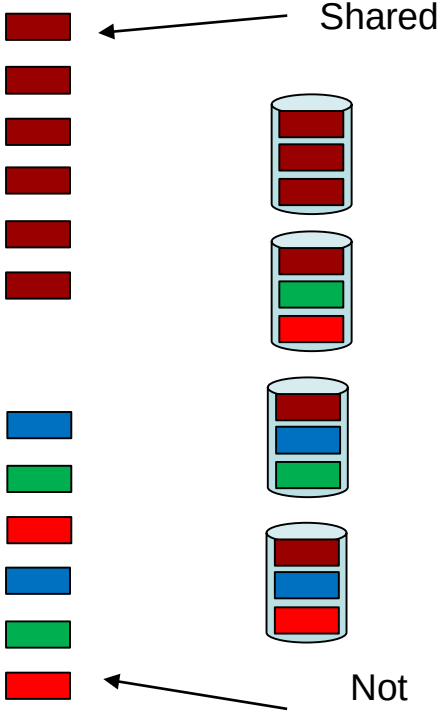


Dynamic Tenant Mapping

Tenants who achieve data security using provider supported (HW) secure erase features

Partitioned Tenant Mapping

Per-tenant data



Storage Partitions

Key Management

- Secure key management techniques must be applied including the use of Key Encryption Keys.
- Any retention of unencrypted data that is in the process of being encrypted or scheduled for same must guaranteed to be unrecoverable after any event that could compromise security such as power loss, reset or component removal.
- Customers should use standards such as KMIP to manage their key store
- Security audits should be performed regularly including the key management

Other Considerations

- Code Origin and Delivery Protection such as digital signatures
  - Signing the executable to prevent malware
  - Integrity mechanisms to ensure non-repudiation of images
- Memory Protection
  - Memory protection is primarily OS process centric. One can view VM's as processes run on hypervisors. Containers run programs in OS processes. All applications run in one or more processes. (Doug to provide update from F2F discussion)
  - Memory Management Units (MMU's) enforce memory protection using both virtual address space mapping and physical memory access protection. Details of both of these levels are MMU Implementation specific.

SDC
STORAGE DEVELOPER CONFERENCE
SNIA SANTA CLARA, 2017

Persistent Memory Threat Model

Roles for Threat Model

- Customer – Security Principal/Data Owner Organization
- Developer – Storage/Application Developer, DevOps
- Security Officer – Security Rights Assigner
- Administrator – System configuration manager
- Deliver-er/Repair-er – Factory/Channel Support, Supply Chain

Secure Insecure

Threat Model

Attack		Attacker	Applicable existing approach	New issues with PM
Cross-Tenant	Privacy/ Confidentiality	Tenant, Administrator, Repair-er	Traditional authorization, authentication. Encryption at rest. Separation of roles. Memory protection.	None
	Integrity	Developer, tenant, administrator	Traditional authorization, authentication. Separation of roles. Memory protection.	Increased scope of damage due to mismanaged pointers, memory resources
	Availability – denial of service	Tenant, Developer	Per-tenant QoS	Potential for rapid disruption with limited detection window

Threat Model

Attack		Attacker	Applicable existing approach	New issues with PM
Cross-Tenant	Tenant, Administrator	Tenant, Administrator, Repair-er	Secure erasure (physical or cryptographic) during deletion	More rapid free space recycling in memory than disk.
Insider	Local HW attacks (e.g. DMA)	Tenant, Administrator, Developer	Memory Protection, Per-tenant QoS applied to IO	
	Remote access threats (e.g. RDMA)	Tenant, Administrator, Developer	RDMA security, s-tag, range access enforcement	

Threat Model

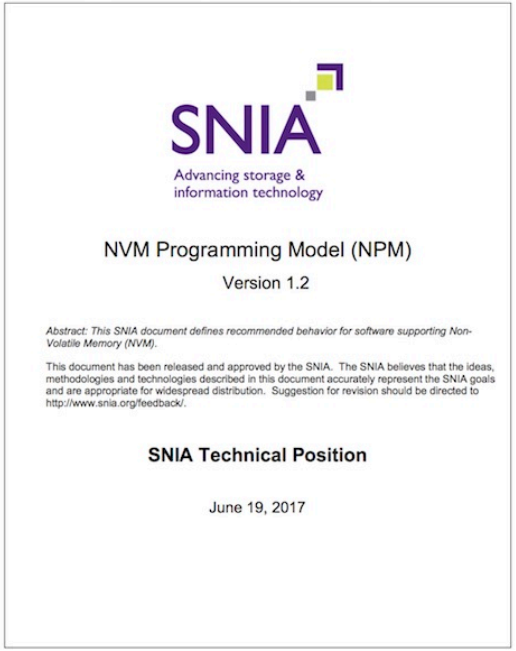
Attack		Attacker	Applicable existing approach	New issues with PM
Insider	Malware	Developer, deliver-er, repair-er, Administrator	Digital signing, virus protection	
	Access by admin/support	Administrator	Role separation, authentication/ Authorization	

What needs done?

- How to control the security features of PM/NVDIMM?
  - Reserved memory area for control structures?
- IOCTL support for establishing a root of trust?
  - Reestablish root on power, reset, hot plug and heartbeat loss
- Shadowing of volatile area (clear text) with PM backing store (cypher text)

Role of the NVM Programming Model

- Rally the industry around a view of Persistent Memory that is:
  - Application centric
  - Vendor neutral
  - Achievable today
  - Beyond storage
    - Applications
    - Memory
    - Networking
    - Processors
- PM Security white paper at https://www.snia.org/tech_activities/publicreview


SNIA
Advancing storage & information technology

NVM Programming Model (NPM)

Version 1.2

Abstract: This SNIA document defines recommended behavior for software supporting Non-Volatile Memory (NVM).

This document has been released and approved by the SNIA. The SNIA believes that the ideas, methodologies and technologies described in this document accurately represent the SNIA goals and are appropriate for widespread distribution. Suggestion for revision should be directed to http://www.snia.org/feedback/.

SNIA Technical Position

June 19, 2017

SDC

STORAGE DEVELOPER CONFERENCE

SNIA SANTA CLARA, 2017

Thank You!