Persistent Memory Security Threat Model

Mark Carlson
Doug Voigt

Presented at Flash Memory Summit 2018
Session PMEM-101-1

Contents

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

Persistent Memory

Persistent Memory (PM) Technology

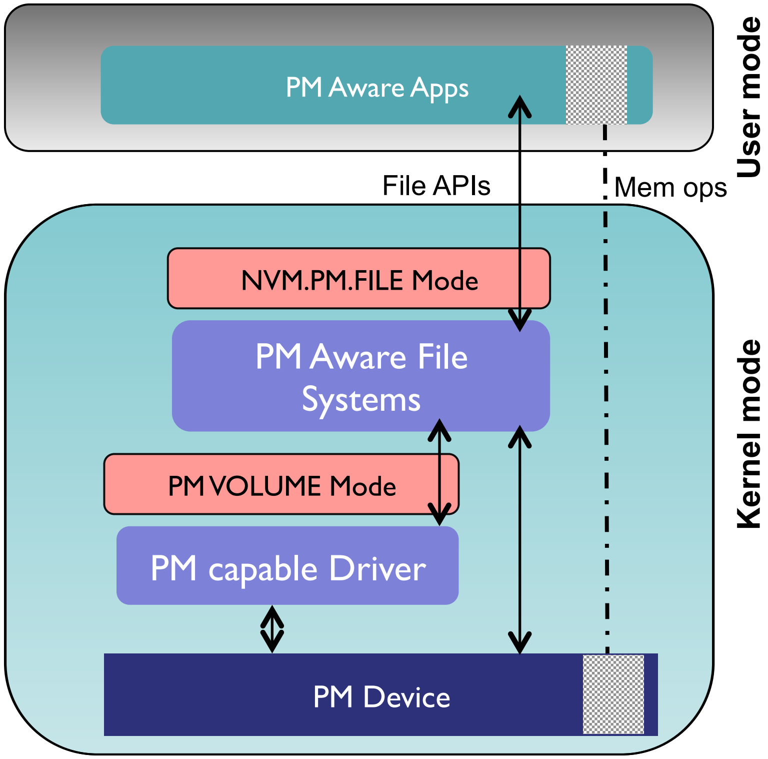
is a type of Non-Volatile Memory (NVM)

- 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

- For more on persistent memory see
  - Bright talks about persistent memory

Persistent Memory (PM) Modes in the NVM Programming Model

- 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



Persistent Memory Security

Purpose of SNIA PM Security work

- This work documents security threats that could create exposure due to unique characteristics of PM. Encryption of data on persistent memory (PM) and multi-tenancy are highlighted.
  - Create a threat model
  - Discover gaps in existing technologies related to PM security
- 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

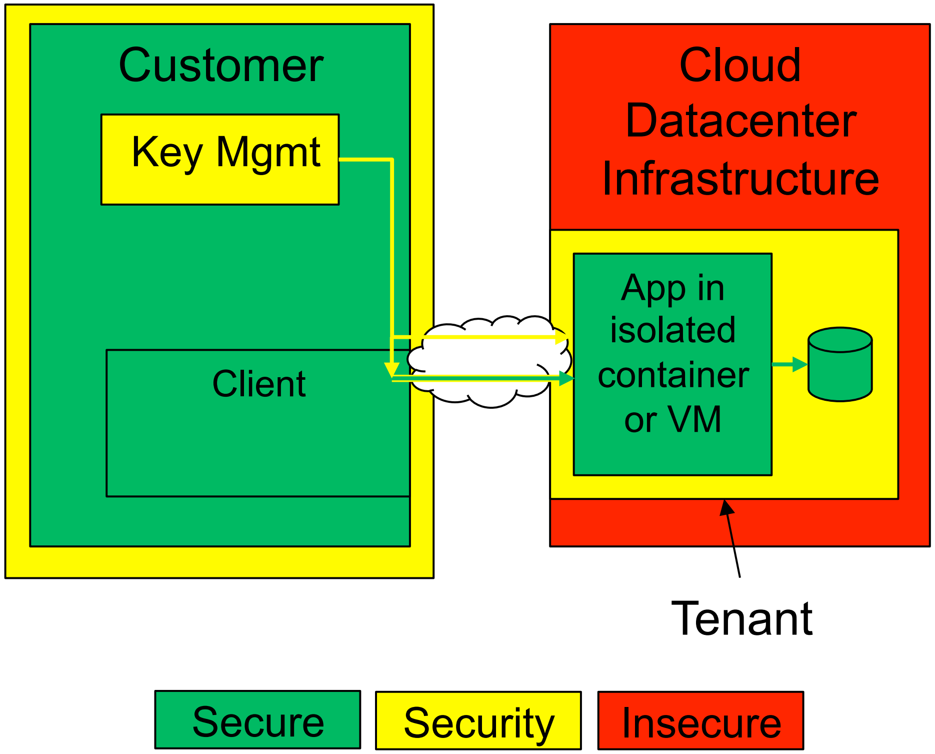
- Protection granularity at the file and volume layers
  - Device, partition or volume protection of data at rest
  - Memory mapped file access authorization enforcement
- Achieving isolation analogous to external storage
  - Limiting access enablement windows
  - Rapid privilege transition
- Malware
  - PM increases the relevance of in-memory virus scan techniques
  - PM may complicate detection of ingest quarantine and scan triggers

Public and private cloud requirements

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

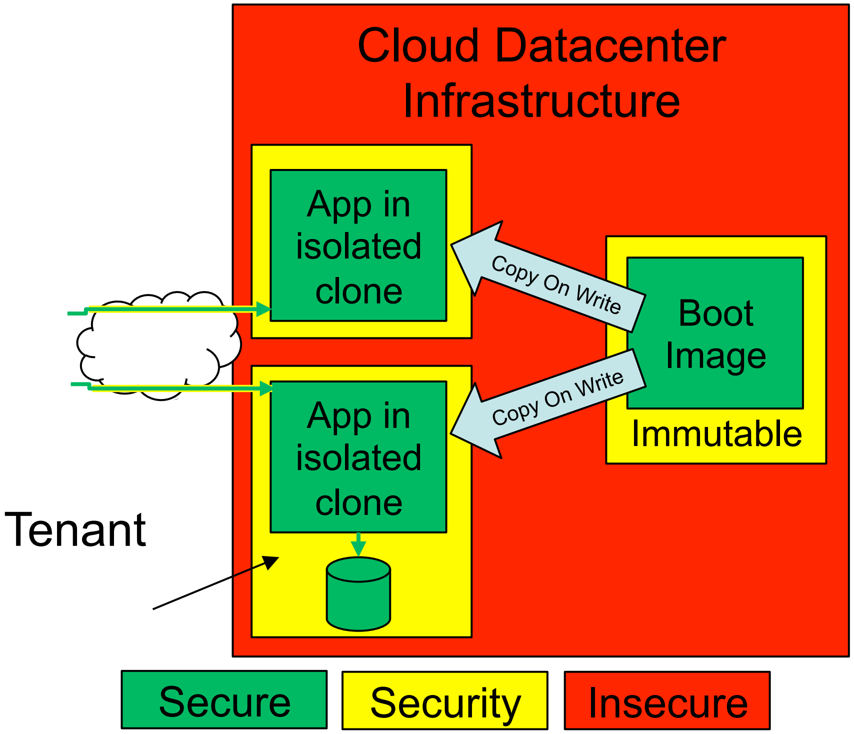
Roles in a Multi-Tenant Cloud Datacenter

- 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.


Customer
Key Mgmt
Client
Cloud Datacenter Infrastructure
App in isolated container or VM
Tenant
Secure
Security
Insecure

PM Clone Use Case

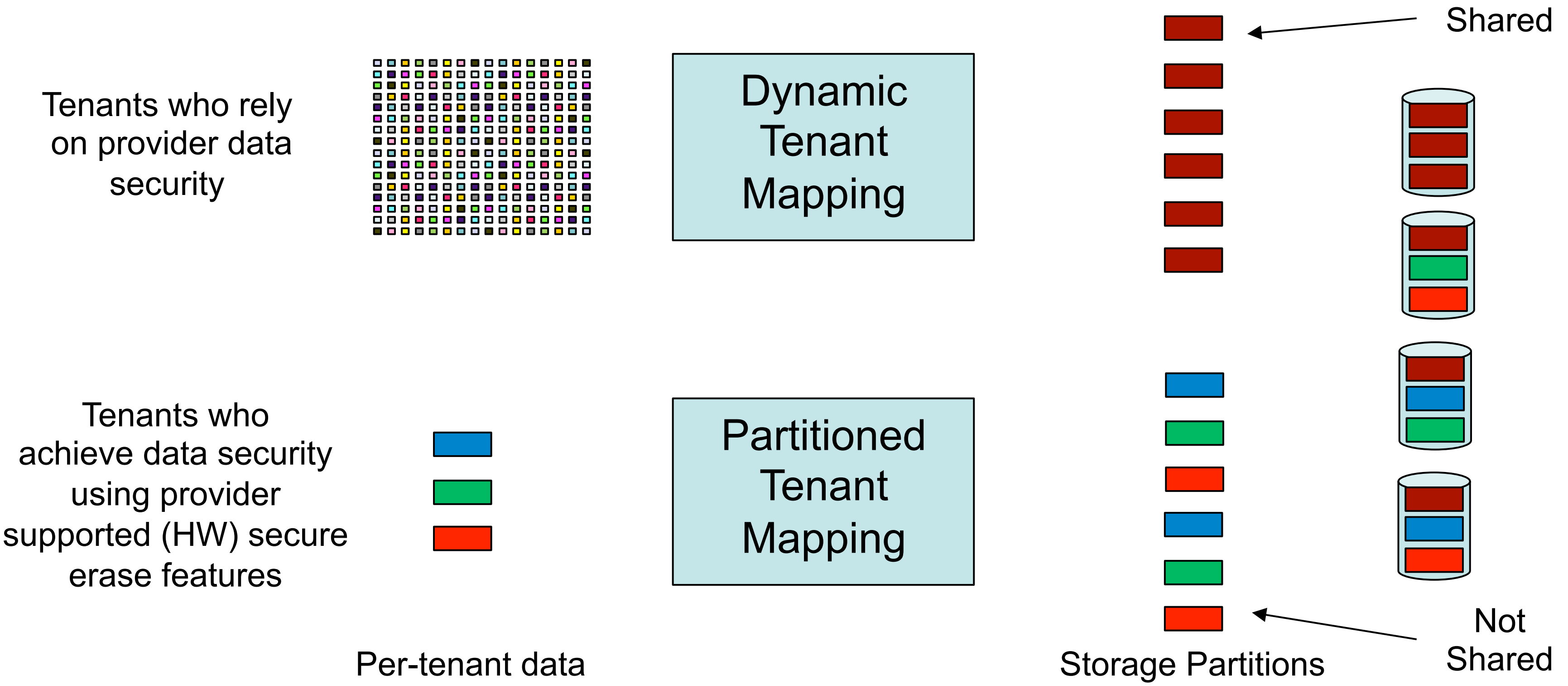
- PM boot image is trusted gold standard
- PM boot image is immutable
- Tenants run in clones of boot image
- Writes exist only in isolated clones
- Additional security such as digital signature and virus protection may be required
- Immutability is ensured by cloud provider and enforced by features 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

Shared

Not Shared

Reasoning about Tenants per Device

- Two potential limiters on number of tenants
  - Device Tenants (Number of tenants supported by one device)
  - Capacity of tenants relative to drives
- Capacity should be the limit, not Device Tenant count
- Tenant capacity is often striped across devices
- Rule of thumb:

$$DeviceTenants > DataDevicesPerGroup * [DeviceCapacity / AverageTenantCapacity\ ]$$

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

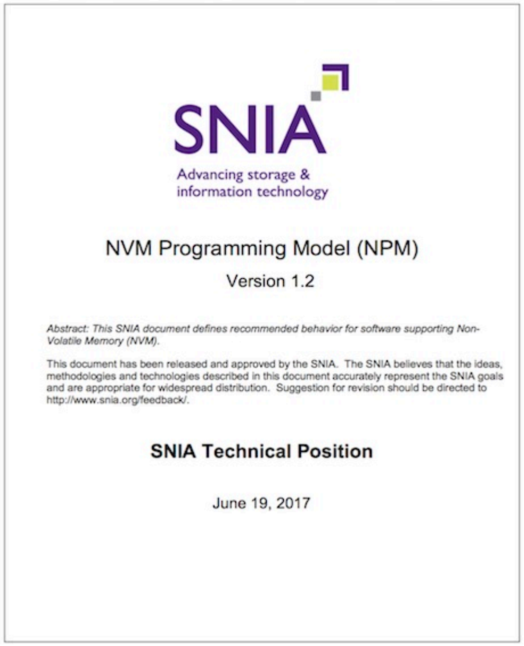
- Code Origin and Delivery Protection such as digital signatures
  - Signing the executable to prevent malware
  - Integrity mechanisms to ensure non-repudiation of images
- Memory Protection
  - Current memory protection practices apply to persistent memory.
  - Memory Management Units (MMU's) enforce memory protection using both virtual address space mapping and physical memory access protection.
  - Details are MMU Implementation specific, and are applied in OS specific ways.

What needs done?

- How to control the security features of PM/NVDIMM?
  - Reserved memory area for control structures?
  - NVMe Persistent Memory Region (PMR)
- 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)
- Additional safeguards on control over memory protection
  - Avoid memory mapping of inactive regions
  - Limit duration and scope of memory write permissions

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

Thank You!