DATAART

PRACTICAL RESULTS: ATTESTATION, REMOTE ATTESTATION, CONFINEMENT AND NETWORK ACCELERATION TECHNOLOGIES

ETSI NFVSEC(16)000104

Michael Lazar
mlazar@dataart.com

DISCLAIMER

DATAART

The information in this presentation is provided "as is" and no guarantee or warranty is given that this information is suitable for any particular purpose. The user thereof uses the information at their own risk.

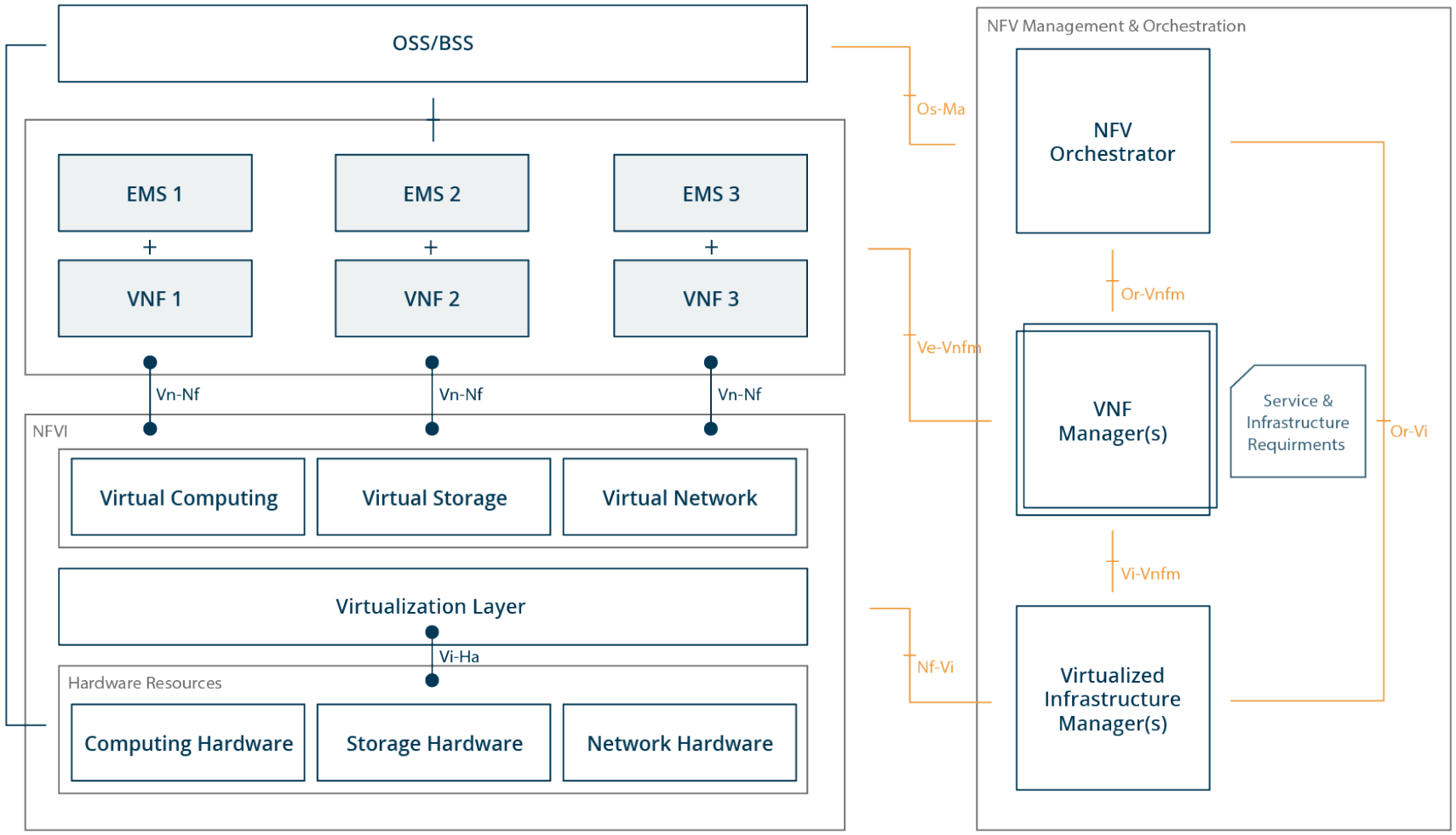
ABOUT THIS PRESENTATION

DATAART

- Used COTS equipment and Open Source Software to put together a ‘real world’ NFV implementation
- System consisted of OpenStack / OPNFV / OpenDaylight
- Intel architecture was utilized
- The implementation utilized several security technologies
- No specific use NFV-SEC case was used.
- Presentation is for educational and informative purposes only

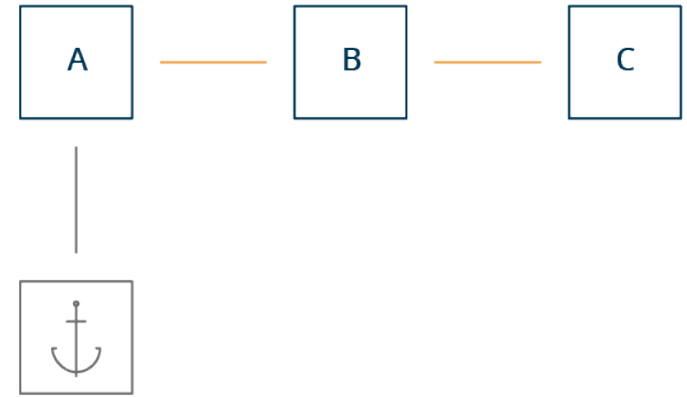
ETSI NFV REFERENCE ARCHITECTURE

DATAART



CHAIN OF TRUST – ATTESTATION IS DESIGNED TO PRODUCE A SECURE ROOT OF TRUST

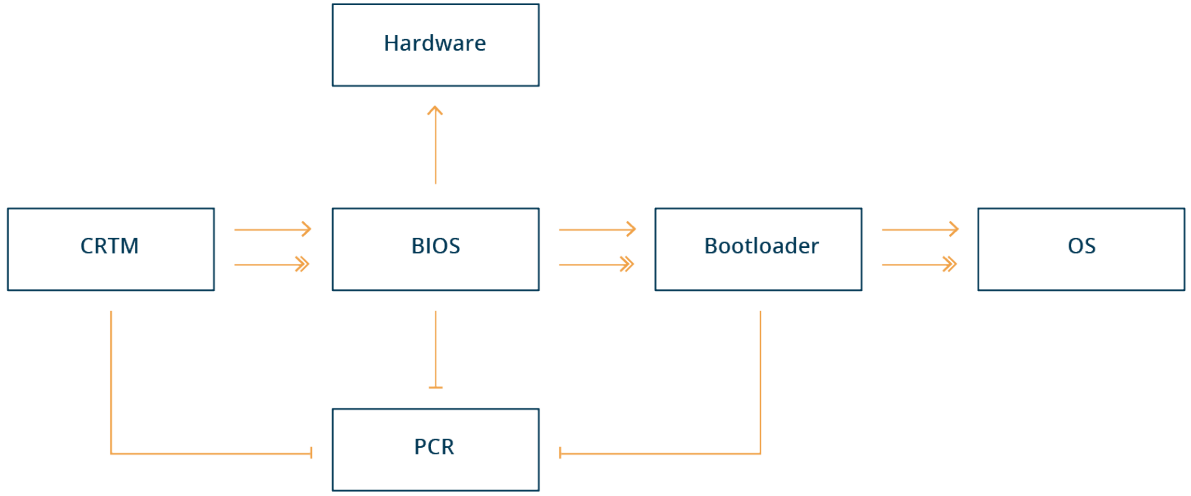
DATAART



Consider that entity A launches entity B, then B launches C.

A measures B then passes control to B
B measures C and passes control to C

The question now becomes "who measures A?"



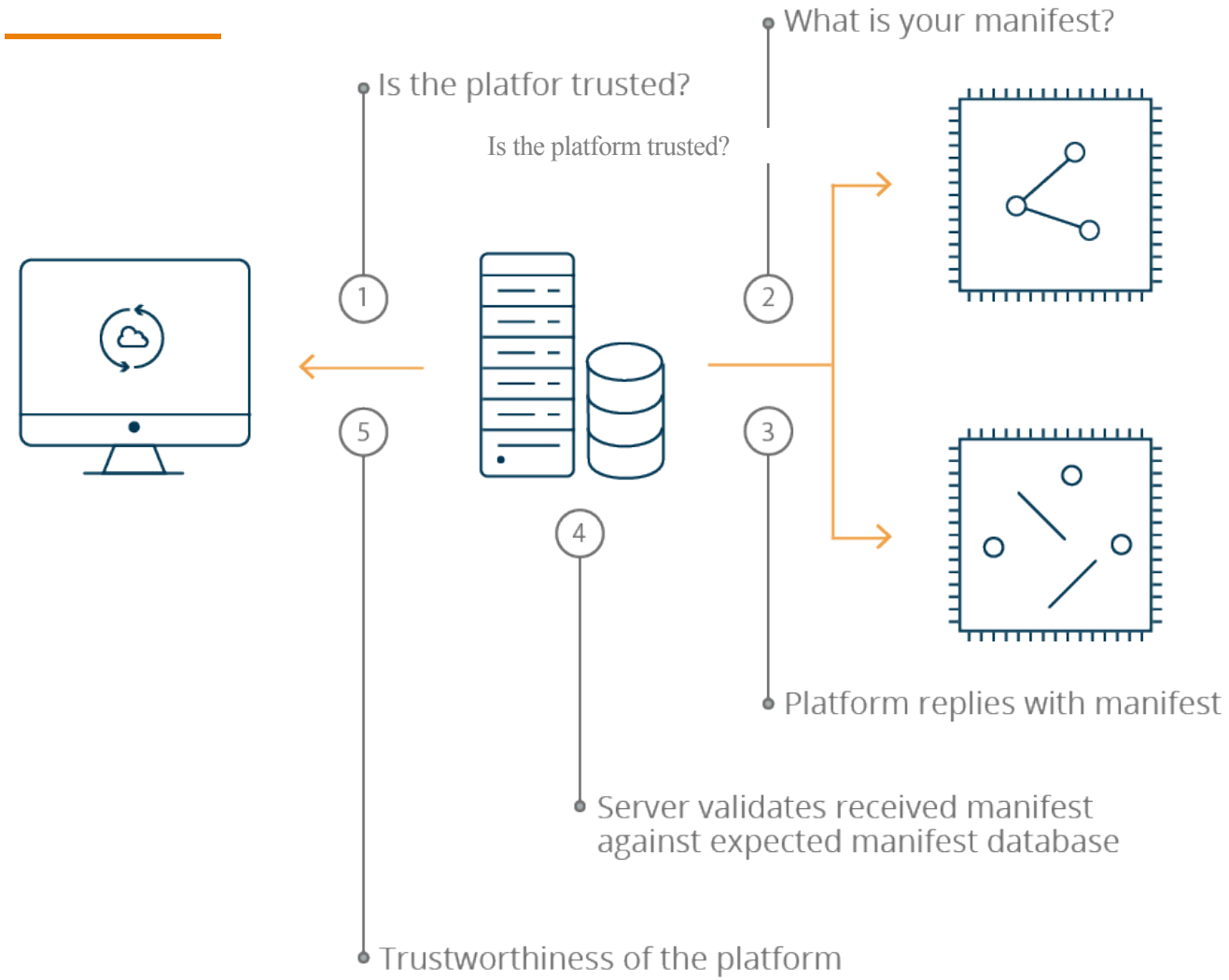
The Core Root of Trust for Measurement (CRTM) is the *BIOS* boot block code. This piece of code is considered trustworthy. It does not change during the life of the system.

When a TPM is present and enabled (TPM_INIT) the system will evaluate components and extend values in the PCRs. These can then be verified to be valid.

*BIOS is being used as a generic term.

REMOTE ATTESTATION ARCHITECTURE – OVERVIEW

DATAART



Remote Attestation is a means by which a trusted computer assures a remote computer of its trustworthy status.

VIRTUALIZATION – THE ‘ROOT’ OF THE ISSUE

DATAART

The (vast) majority of todays commercial physical compute resources and operating systems fundamentally work off of a implicit trust model. To be more explicit, there is trust between the hardware subsystems and kernel operations. Even when zero trust models are implemented in user space, todays kernels (and kernel variants) rely on implicit trust to function.

Virtualization attack vectors have become more sophisticated focusing on virtual machine attacks (break out), hypervisor attacks (blue pill) and compromised hardware (malicious hardware).

Over the last years several hardware and software technologies have been made available, including VT-d, Authenticated boot, Trusted Platform Modules (TPM), Trusted boot (tboot), SELinux, sVirt, AppArmor, OAT SDK (remote attestation toolkit) and Trusted Execution Technology (TXT) to make platforms more secure.

Additional technologies are available or emerging including TrustZone (ARM/AMD) and Software Guard Extensions (Intel SGX).

SOME ATTACK VECTORS – POTENTIAL REMEDIATION

DATAART

-- Intel terminology used (not meant to be exclusive)

- **VECTOR**

Modified Hardware (malicious)	Authenticated boot / Static measurement w/TPM
Foreign Hardware (unauthorized/ malicious)	Intel VT-d (AMD IOMMU) prevents raw PCI access
Side channel attacks (breakout of confinement)	SELinux, sVirt, AppArmor, Seccomp2
Hypervisor modification (malicious code)	Trusted Execution Technology (TXT) / tboot / Remote attestation

Several hypervisors are available (OpenStack supports many variants) not all are equally secure

- **Potential Remediation**

  SRTM - Static Root of Trust for Measurements (authenticated boot) with TPM is a mature technology. Hardware may be attested at boot time, however, SRTM requires keeping measurements of the entire platform boot sequence including BIOS config, 3rd party boot ROMs (e.g. network cards). Any change to the environment requires new measurements (which are disruptive and complex to maintain).

  DRTM - Dynamic Root of Trust for Measurements (tboot) utilizes TXT and TPM. It can verify the hardware and software (hypervisor) have not been tampered with on boot and can take direct action (halt). While it can replace SRTM they may also be used together allowing SRTM to provide a key measurements (PCR0/1) while DRTM provides the remainder of the measurements without ‘loading’ down the attestation process.

  Utilizing Intel TXT and TPM the OpenAttestationToolkit (OAT) provides Remote Attestation and “Trusted Compute Pools” (implementation as a Security Console)

SO WE HAVE “TRUSTED HOSTS” EVERYTHING IS GOOD – RIGHT?

DATA ART

- Implementation in OpenStack/OPFNV is set by **trusted_filter flag** in the abstract scheduler for a VM.
  - Change the flag in memory and the VM is no longer required to run in a trusted environment.
  - Alter the scheduler to ignore the flag, while reporting everything is ‘OK’.
  - OAT (SDK) uses certificates.
- Binary trust model – hosts / hypervisors are either trusted or not. Currently there is no available implementation of a hierarchal trust model.
- VMs with the trusted flag set can only run on trusted hosts (attested), however untrusted VMs (no flag set) **can also run on trusted hosts**.
- Only Compute nodes (NOVA) support remote attestation. Management and storage nodes do not support natively.
- Neutron (networking service) runs on unattested hosts. Compromise of management and storage nodes adds to attack surface.
- Similar to hardware trust model issues, the virtualization administrators have nearly unlimited power.

SO WE HAVE TRUSTED COMPUTE POOLS – EVERYTHING IS GOOD – RIGHT?

DATAART

- Revoking trust on a running host leaves trusted VMs operational (they continue to run)
- Compute pools are managed by a "security station", which now becomes the target. OAT for OpenStack requires SELINUX to run in permissive mode by default.
- In general SELinux / sVirt / AppArmor rules are deficient. Either missing or set so loose as to reduce the value of confinement
- Shared memory represents risks for leakage or exposure of sensitive data (KSM)
- High performance networking solutions (Intel DPDK) have security considerations (shared VM+GuestOS memory or disabling of confinement).
- Hypervisor choice matters:
  - **Baremetal** (Ironic)- still early in development lifecycle, under/over cloud implementation does not support SELinux/sVirt out of the box. PXE+TFTP loading not secure. Ironman plugin available
  - **KVM** - under/over cloud implementation does not support SELinux/sVirt out of the box.
  - **QEMU** – hardening guide specifically mentions as a risk, however high performance solutions continue to use/require it
  - **LXD/LXC** - containers will always (by design) share the same kernel as the host. Therefore, any vulnerabilities in the kernel interface, unless the container is forbidden the use of that interface (i.e. using seccomp2) can be exploited by the container to harm the host.

SOME THOUGHTS

DATAART

Virtual environments create new challenges for security, it is not the same old world just running on a hypervisor.

Several technologies now exist to help create more secure virtual environments that start with the basic concept of ‘hardware root of trust’, through booting trusted hypervisors and virtual machines (containers, etc). However due to implementation issues these are not as secure as they seem at first review.

Vendors need to be encouraged to use available technologies as “default”.
(random sampling shows TXT is not enabled and in several cases VT-d was disabled)

The release of the OAT as an SDK was done to enable commercial development (BSD license). It does not appear that this has been embraced, instead the absolute minimum has been done to ‘check the box’.

The binary trust zone issue should be addressed.

Need to work with Open Source communities to encourage better security models using available technology.

DATAART

Thank You !